Graver basis for an undirected graph and its application to testing the beta model of random graphs

Mitsunori Ogawa[*], Hisayuki Hara[†] and Akimichi Takemura[*‡]

February 2011

Abstract

In this paper we give an explicit and algorithmic description of Graver basis for the toric ideal associated with a simple undirected graph and apply the basis for testing the beta model of random graphs by Markov chain Monte Carlo method.

Keywords and phrases: Markov basis, Markov chain Monte Carlo, Rasch model, toric ideal.

1 Introduction

Random graphs and their applications to the statistical modeling of complex networks have been attracting much interest in many fields, including statistical mechanics, ecology, biology and sociology (e.g. Newman [11], Goldenberg et al. [7]). Statistical models for random graphs have been studied since Solomonoff and Rapoport [19] and Erdős and Rényi [6] introduced the Bernoulli random graph model. The beta model generalizes the Bernoulli model to a discrete exponential family with vertex degrees as sufficient statistics. The beta model was discussed by Holland and Leinhardt [9] in the directed case and by Park and Newman [15], Blitzstein and Diaconis [2] and Chatterjee et al. [3] in the undirected case. The Rasch model [17], which is a standard model in the item response theory, is also interpreted as a beta model for undirected complete bipartite graphs. In this article we discuss the random sampling of graphs from the conditional distribution in the beta model when the vertex degrees are fixed.

In the context of social network the vertices of the graph represent individuals and their edges represent relationships between individuals. In the undirected case the graphs are sometimes restricted to be simple, i.e., no loops or multiple edges exist. The sample

[*]Department of Mathematical Informatics, Graduate School of Information Science and Technology, University of Tokyo.

[†]Department of Technology Management for Innovation, University of Tokyo

[‡]JST CREST

size for such cases is at most the number of edges of the graph and is often small. The goodness of fit of the model is usually assessed by large sample approximation of the distribution of a test statistic. When the sample size is not large enough, however, it is desirable to use a conditional test based on the exact distribution of a test statistic.

Random sampling of graphs with a given vertex degree sequence enables us to numerically evaluate the exact distribution of a test statistics for the beta model. Blitzstein and Diaconis [2] developed a sequential importance sampling algorithm for simple graphs. In this article we construct a Markov chain Monte Carlo algorithm for sampling graphs by using the Graver basis for the toric ideal arising from the beta model.

A Markov basis [4] is often used for sampling from discrete exponential families. Algebraically a Markov basis for the beta model is defined as a set of generators of the toric ideal arising from the beta model. A set of graphs with a given degree sequence is called a fiber of the beta model. A Markov basis for the beta model is also considered as a set of Markov transition operators connecting every fiber. Petrović et al. [16] discussed some properties of the toric ideal arising from the model of [9] and provided Markov bases of the model for small directed graphs. Properties of toric ideals arising from a graph have been studied in a series of papers by Ohsugi and Hibi ([13], [12], [14]).

The Graver basis is the set of primitive binomials of the toric ideal. Since the Graver basis is a superset of any minimal Markov basis, the Graver basis is also a Markov basis and therefore connects every fiber. When the graph is restricted to be simple, however, a Markov basis does not necessarily connect every fiber. A recent result by Hara and Takemura [8] implies that the set of square-free elements of the Graver basis connects every fiber of simple graphs with a given vertex degree sequence. Thus if we have the Graver basis for the beta model, we can sample graphs from any fiber, with or without the restriction that graphs are simple, in such a way that every graph in the fiber is generated with positive probability.

In the sequential importance sampling algorithm of [2] the underlying graph for the model was assumed to be complete, i.e., all the edges have positive probability. In our approach we can allow that some edges are absent from the beginning (structural zero edges in the terminology of contingency table analysis), such as the bipartite graph for the case of the Rasch model. In fact the Graver basis of an arbitrary graph is obtained by restriction of the Graver basis of the complete graph to the existing edges of G (cf. Proposition 4.13 of Sturmfels [20]). This is the advantage of obtaining the Graver basis.

The Graver basis for small graphs can be computed by a computer algebra system such as 4ti2 [1]. For even moderate-sized graphs, however, it is difficult to compute the Graver basis via 4ti2 in a practical amount of time. In this article we first provide a complete description of the Graver basis for the beta model. In general the number of elements of the Graver basis is too large. So we construct an adaptive algorithm for sampling elements from the Graver basis, which is enough for constructing a connected Markov chain over any fiber. Our theoretical results on the Graver basis have many overlaps with the results of resent paper of Reyes et al. ([18]). However our results are more suitable for sampling elements from the Graver basis.

The organization of this paper is as follows. In Section 2 we give a brief review on some statistical models for random graphs and clarify the connection between the models and toric ideals arising from graphs. In Section 3 we provide an explicit description of the Graver basis for the toric ideal associated with an undirected graph. Section 4 gives an algorithm for random sampling of square-free elements of the Graver basis. In Section 5 we apply the proposed algorithm to some data sets and confirm that it works well in practice. We conclude the paper with some remarks in Section 6.

2 The beta model of random graphs

In this section we give a brief review of the beta model for undirected graphs according to Chatterjee et al. [3].

Let G be an undirected graph with n vertices $V(G) = \{1, 2, \ldots, n\}$. Here we assume that G has no loop. Let $E = E(G)$ be the set of edges and let $d_1, \ldots, d_n$ be a degree sequence. Denote $\boldsymbol{d} := (d_1, \ldots, d_n)$. For each edge $\{i, j\} \in E$, let a non-negative integer x_{ij} be the weight for $\{i, j\}$ and denote $\boldsymbol{x} = \{x_{ij} \mid \{i, j\} \in E\}$. $\boldsymbol{x}$ is considered as an $|E|$ dimensional integer vector. We assume that an observed graph H is generated by independent binomial distribution $B(n_{ij}, p_{ij})$ for each edge $\{i, j\} \in E$, i.e., $x_{ij} \sim B(n_{ij}, p_{ij})$ with

$$p_{ij} := \frac{e^{\beta_i+\beta_j}}{1+e^{\beta_i+\beta_j}} = \frac{\alpha_i\alpha_j}{1+\alpha_i\alpha_j}, \qquad \alpha_i = e^{\beta_i}.$$

Then the probability of H is described as

$$\begin{aligned} P(H) &\propto \prod_{\{i,j\}\in E} p_{ij}^{x_{ij}}(1-p_{ij})^{n_{ij}-x_{ij}} \\ &= \frac{1}{\prod_{\{i,j\}\in E}(1+\alpha_i\alpha_j)^{n_{ij}}} \prod_{\{i,j\}\in E} (\alpha_i\alpha_j)^{x_{ij}} \\ &= \frac{\prod_{i\in V} \alpha_i^{d_i}}{\prod_{\{i,j\}\in E}(1+\alpha_i\alpha_j)^{n_{ij}}}, \end{aligned} \tag{1}$$

where we note that $d_i = \sum_{j:\{i,j\}\in E} x_{ij}$. The model (1) is called the beta model ([3]). Note that if $x_{i,j} = 0$ then the observed graph H does not have an edge $\{i, j\}$ even if $\{i, j\} \in E(G)$ for the underlying graph G.

This model was considered by many authors (e.g. Park and Newman [15], Blitzstein and Diaconis [2] and Chatterjee et al. [3]). The p_1 model for random directed graphs by Holland and Leinhardt [9] can be interpreted as a generalization of the beta model. When G is a complete bipartite graph, the beta model coincides with the Rasch model ([17]). The many-facet Rasch model by Linacre [10], which is a multivariate version of the Rasch model, can be interpreted as a generalization of the beta model such that G is a complete k-partite graph.

The sufficient statistic for (1) is $\boldsymbol{d}$. Let $\boldsymbol{A} : |V| \times |E|$ denote the incidence matrix between vertices and edges of G. Then it is easily seen that $\boldsymbol{x}$ and $\boldsymbol{d}$ is related as

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{d}.$$

A set of graphs (without restriction to be simple) with a given degree sequence $\boldsymbol{d}$ is called a fiber of the beta model. An integer array $\boldsymbol{z}$ of the same dimension as $\boldsymbol{x}$ is called a move of the beta model if $\boldsymbol{A}\boldsymbol{z} = 0$. A move $\boldsymbol{z}$ is written as the difference of its positive part and negative part as $\boldsymbol{z} = \boldsymbol{z}^+ - \boldsymbol{z}^-$. Since $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{A}\boldsymbol{z}^+ - \boldsymbol{A}\boldsymbol{z}^-$, every move is written as the difference of two graphs in the same fiber. A finite set of moves is called a Markov basis if for every fiber any two graphs are mutually accessible by the moves in the set ([4]). By adding or subtracting moves in a Markov basis, we can sample graphs from any fiber in such a way that every graph in the fiber is generated with positive probability.

As mentioned in Section 1, graphs are restricted to be simple in some practical problems. For a simple graph, x_{ij}, $\{i, j\} \in E$, is either zero or one. A Markov basis guarantees the connectivity of every fiber if the restriction that graphs are simple is not imposed. Under the restriction, however, a Markov basis does not necessarily connect every fiber.

For a given $\boldsymbol{x}$, $\mathrm{supp}(\boldsymbol{x}) = \{e \mid x_e > 0\}$ denotes the set of observed edges of $\boldsymbol{x}$. For two moves $\boldsymbol{z}_1, \boldsymbol{z}_2$, the sum $\boldsymbol{z}_1 + \boldsymbol{z}_2$ is called conformal if there is no cancellation of signs in $\boldsymbol{z}_1 + \boldsymbol{z}_2$, i.e., $\emptyset = \mathrm{supp}(\boldsymbol{z}_1^+) \cap \mathrm{supp}(\boldsymbol{z}_2^-) = \mathrm{supp}(\boldsymbol{z}_1^-) \cap \mathrm{supp}(\boldsymbol{z}_2^+)$. The set of moves which can not be written as a conformal sum of two nonzero moves is called the Graver basis. The Graver basis is known to be a Markov basis (e.g. [5]). A move is square-free if the absolute values of its elements are 0 or 1. As a corollary of Proposition 2.1 of Hara and Takemura [8], we can easily obtain the following proposition.

Proposition 1. *The set of square-free moves of the Graver basis for the beta model connects every fiber of simple graphs.*

Therefore it suffices to have the Graver basis to sample graphs from any fiber with or without the restriction that graphs are simple. In the next section we derive the Graver basis for the beta model.

3 Graver basis of a random graph

In this section we will give a simple characterization of the Graver basis for an undirected graph. Theorem 1 in Section 3.2 is the main result of this paper. As mentioned in Section 1 it has many overlaps with [18]. However our Theorem 1 is convenient for constructing an algorithm for sampling elements from the Graver basis, as shown in Algorithm 1 in Section 4.

3.1 Preliminaries

Let $G = (V(G), E(G))$ be a simple connected graph with $V(G) = \{1, 2, \ldots, n\}$ and $E(G) = \{e_1, e_2, \ldots, e_m\}$. A *walk* connecting $i \in V(G)$ and $j \in V(G)$ is a finite sequence of edges of the form

$$w = (\{i_1, i_2\}, \{i_2, i_3\}, \ldots, \{i_q, i_{q+1}\})$$

with $i_1 = i, i_{q+1} = j$. The *length* of the walk w is the number of edges q of the walk. An *even* (respectively *odd*) walk is a walk of even (respectively *odd*) length. A walk w is *closed* if $i = j$. A *cycle* is a closed walk $w = (\{i_1, i_2\}, \{i_2, i_3\}, \ldots, \{i_q, i_1\})$ with $i_l \neq i_{l'}$ for every $1 \leq l < l' \leq q$.

For a walk w, let $V(w) = \{i_1, \ldots, i_{q+1}\}$ denote the set of vertices appearing in w and let $E(w) = \{\{i_1, i_2\}, \{i_2, i_3\}, \ldots, \{i_q, i_{q+1}\}\}$ denote the set of edges appearing in w. Furthermore let $G_w = (V(w), E(w))$ be the subgraph of G, whose vertices and edges appear in the walk w.

In order to describe known results on the toric ideal I_G arising from an undirected graph G, we give an algebraic definition of I_G. Let $K[\mathbf{t}] = K[t_1, \ldots, t_n]$ be a polynomial ring in n variables over K. We will associate each edge $e_r = \{i, j\} \in E(G)$ with the monomial $\mathbf{t}_r = t_i t_j \in K[\mathbf{t}]$. Let $K[\mathbf{s}] = K[s_1, \ldots, s_m]$ be a polynomial ring in $m = |E(G)|$ variables over K and let π be a homomorphism from $K[\mathbf{s}]$ to $K[\mathbf{t}]$ defined by $\pi : s_r \mapsto \mathbf{t}_r$. Then the *toric ideal* I_G of the graph G is defined as

$$I_G = \ker(\pi) = \{f \in K[\mathbf{s}] \mid \pi(f) = 0\}.$$

A binomial $f = u - v \in I_G$ is called *primitive* if there is no binomial $g = u' - v' \in I_G$, $g \neq 0, f$, such that $u'|u$ and $v'|v$. The *Graver basis* of I_G is the set of all primitive binomials belonging to I_G and we denote it by $\mathcal{G}(I_G)$. If we write the monomials u, v as $u = s^{\boldsymbol{x}}, v = s^{\boldsymbol{y}}$, then $u - v \in I_G$ if and only if $\boldsymbol{x} - \boldsymbol{y}$ is a move. Furthermore $u - v \in I_G$ is primitive if and only if $\text{supp}(\boldsymbol{x}) \cap \text{supp}(\boldsymbol{y}) = \emptyset$ and $\boldsymbol{x} - \boldsymbol{y}$ can not be written as a conformal sum of two nonzero moves.

For a given even closed walk $w = (e_{j_1}, e_{j_2}, \ldots, e_{j_{2p}})$ we define a binomial $f_w \in I_G$ as

$$f_w = f_w^+ - f_w^-, \qquad \text{where } \; f_w^+ = \prod_{k=1}^{p} s_{j_{2k-1}}, \;\; f_w^- = \prod_{k=1}^{p} s_{j_{2k}}.$$

An even closed walk w' is a *proper subwalk* of w, if $g_{w'}^+ \mid f_w^+$ and $g_{w'}^- \mid f_w^-$ hold for the binomial $g = g_{w'}^+ - g_{w'}^- (\neq f_w)$. An even closed walk w is called *primitive*, if its binomial f_w is primitive. Then the primitiveness of w is equal to non-existence of a proper subwalk of w.

A characterization of the primitive walks of graph G, which gives a necessary condition for a binomial to be primitive, was given by Ohsugi and Hibi [13].

Proposition 2 (Ohsugi and Hibi [13]). *Let G be a finite connected graph. If $f \in I_G$ is primitive, then we have $f = f_w$ where w is one of the following even closed walks:*

(i) w is an even cycle of G.

(ii) $w = (c_1, c_2)$, where c_1 and c_2 are odd cycles of G having exactly one common vertex.

(iii) $w = (c_1, w_1, c_2, w_2)$*, where* c_1 *and* c_2 *are odd cycles of* G *having no common vertex and where* w_1 *and* w_2 *are walks of* G *both of which contain a vertex* v_1 *of* c_1 *and a vertex* v_2 *of* c_2*.*

Every binomial in the first two cases is primitive but a binomial in the third case is not necessarily primitive.

3.2 Characterization of primitive walks

In this subsection we give a simple characterization of the primitive walks of a graph G as sequences of vertices. Express an even closed walk w as a sequence of vertices: $(i_1, i_2, \ldots, i_{2p}, i_1)$, where $i_1 \equiv i_{2p+1}$. Let $\#_w(i) = \#\{1 \leq l \leq 2p \mid i_l = i\}$ denote the number of times i is visited in the walk w before it returns to the vertex i_1. Consider the following condition for the even closed walk w.

Condition 1. *(i)* $\#_w(i) \in \{1, 2\}$ *for every vertex* $i \in V(w)$*. (ii) For every vertex* $j \in V(w)$ *with* $\#_w(j) = 2$ *and* $j = i_l = i_{l'}$*,* $1 \leq l < l' \leq 2p$*, the closed walks* $w_1^j = (i_l, \ldots, i_{l'})$ *and* $w_2^j = (i_{l'}, \ldots, i_{2p}, i_1, \ldots, i_{l-1}, i_l)$ *are odd walks with* $V(w_1^j) \cap V(w_2^j) = \{j\}$*. (cf. Figure 1).*

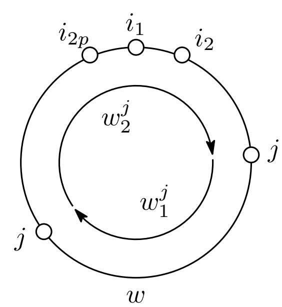

Figure 1: Even closed walk w.

Remark 1. *The equation* $V(w_1^j) \cap V(w_2^j) = \{j\}$ *in Condition 1 means that there are no crossing chords in Figure 1 when adding a chord* $\{j, j\}$ *to the figure for every vertex* $j \in V(w)$ *with* $\#_w(j) = 2$*.*

Using Condition 1, we can characterize primitive walks of a graph G as follows.

Theorem 1. *A binomial* $f \in I_G$ *is primitive if and only if there exists an even closed walk* w *with* $f_w = f$ *satisfying Condition 1.*

Remark 2. *It follows from the definition of primitive walks and Theorem 1 that if an even closed walk* w *is primitive, every even closed walk* w' *with* $f_{w'} = f_w$ *is primitive and satisfies Condition 1.*

To prove Theorem 1, another characterization of primitive walks will be given in Proposition 3. In order to that, we need some more definitions on graphs. For a walk $w = (e_{j_1}, e_{j_2}, \dots, e_{j_q})$, let $W = W(w)$ denote the weighted subgraph $(V(w), E(w), \rho)$ of G where $\rho : E(w) \to \mathbb{Z}$ is the weight function defined by $\rho(e) := \#\{l \mid e_{j_{2l+1}} = e\} - \#\{l \mid e_{j_{2l}} = e\}$ for each edge $e \in E(w)$. For simplicity, we denote a weight $+1$ (respectively -1) by $+$ (respectively $-$) in our figures. For a vertex $i \in V(w)$, we define two kinds of degrees of vertex i:

$$\begin{aligned} \deg_{G_w}(i) &= \#\{e \in E(w) \mid i \in e\}, \\ \deg_W(i) &= \sum_{e \in E(w): i \in e} |\rho(e)|. \end{aligned}$$

$\deg_{G_w}(i)$ is the usual degree of i in G_w. Note that the same weighted graph W might correspond to two different even closed walks w, w', i.e. $W(w) = W(w')$. Given a weighted graph W, we say that w *spans* W if $W = W(w)$ and $\{e_{j_l} \mid l\text{:odd}\} \cap \{e_{j_l} \mid l\text{:even}\} = \emptyset$.

Now we define two operations, *contraction* and *separation*, on a weighted graph W.

- Let $e = \{i, j\} \in E(w)$ be an edge with $|\rho(e)| = 2$, whose removal from G_w increases the number of connected components of the remaining subgraph. *Contraction* of e is an operation as shown in Figure 2. That is, it first replaces W by $W' = (V(w) \setminus \{j\}, E', \rho')$ where E' consists of all edges of W contained in $V(w) \setminus \{j\}$, together with all edges $\{\alpha, i\}$, where $\{\alpha, j\}$ is an edge of W different from e. Then, it defines ρ' by inversion of the signs of weights of edges belonging to the i-side of W.

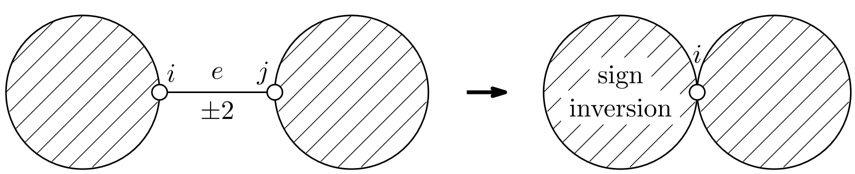

Figure 2: Contraction.

- Let $i \in V(w)$ be a vertex with $\deg_{G_w}(i) = \deg_W(i) = 4$, such that the removal of i increases the number of connected components of the remaining subgraph and the positive side as well as the negative side of i fit to one of three cases (a)–(c) (respectively to the sign reverse cases) in Figure 3. *Separation* of i is an operation as shown in Figure 3. That is, it first deletes the vertex i and all edges connected to i on W. Then, in the case of (a), it adds a new edge $\{k_1, k_2\}$ with weight $+1$. In the case of (b), it redefines $\rho(\{k_1, k_2\}) := +2$ and then contracts $\{k_1, k_2\}$, where we assume that the contraction of $\{k_1, k_2\}$ is possible. In the case of (c), it redefines $\rho(\{k_1, k_2\}) := 0$. We call this $\{k_1, k_2\}$ an edge with weight 0. The sign reverse cases are defined in the same way.

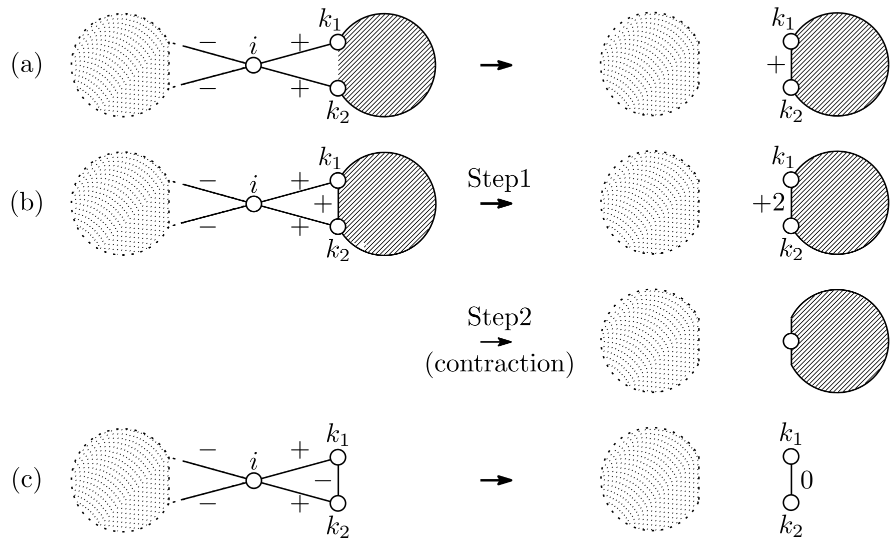

Figure 3: Separation.

Note that the separation is not defined for any vertex i with $\deg_{G_w}(i) = \deg_W(i) = 4$, if i fits to none of three cases (a)–(c) in Figure 3. The vertex i in Figure 4 is such an example, because its positive side fits to none of three cases (a)–(c) in Figure 3.

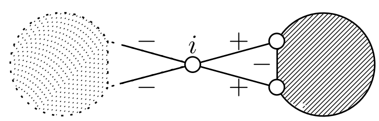

Figure 4: A vertex i whose separation is not defined.

Let *insertion* and *binding* be the reverse operations of contraction and separation, respectively. With these operations, consider the following condition for an even closed walk $w = (e_{j_1}, e_{j_2}, \ldots, e_{j_{2p}})$.

Condition 2. *(i)* $\{e_{j_l} \mid l{:}odd\} \cap \{e_{j_l} \mid l{:}even\} = \emptyset$. *Every vertex* $i \in V(w)$ *satisfies* $deg_W(i) \in \{2, 4\}$. *For every vertex* i *with* $deg_W(i) = 4$, *its removal from* G_w *increases the number of connected components of the remaining subgraph. (ii) Let* $\tilde{W}$ *be a graph obtained by recursively applying contraction and separation of all possible edges and vertices in* W. *Then each connected component of* $\tilde{W}$ *is an even cycle or an edge with weight 0.*

Proposition 3. *For an even closed walk* w, *the binomial* f_w *is primitive if and only if* w *satisfies Condition 2.*

We establish some lemmas to prove Proposition 3. Our proof also shows that $\tilde{W}$ in Condition 2 does not depend on the order of application of contractions and separations excepting the sign inversion of weights of edges of each connected component in $\tilde{W}$.

Lemma 1. *If an even closed walk w is primitive, w satisfies (i) in Condition 2.*

Proof. Consider a vertex $i \in V(w)$. Since w is closed, $\deg_W(i)$ is even. Furthermore, since w is primitive, $\{e_{j_l} \mid l\text{:odd}\} \cap \{e_{j_l} \mid l\text{:even}\} = \emptyset$ holds which implies that there is no cancellation in the calculation of weight on any edge. Then, a half of the weight $\deg_W(i)/2$ is assigned as positive weights and other half $\deg_W(i)/2$ is assigned as negative weights to the edges connected to i on W. Therefore $\deg_W(i) \in \{2, 4, 6, \dots\}$. Now suppose $\deg_W(i) \geq 6$. Consider that we start from a vertex i along an edge with positive weight and go along the walk w or its reverse until returning back to i again for the first time. We can always take this closed walk since w is closed. If we come back to i for the first time along an edge with negative weight, we can take it as a proper subwalk of w, which contradicts the primitiveness of w. Therefore we have to come back to i along an edge with positive weight. Let us continue along w or its reverse until returning back to i. By the same reasoning, the last edge of this closed walk has a negative weight. This implies that we can take this even closed walk as a proper subwalk of w, a contradiction to the primitiveness of w. Therefore $\deg_W(i)$ is 2 or 4.

To prove the remaining part, let $i \in V(w)$ be a vertex with $\deg_W(i) = 4$ and consider all closed walks on W, where the edge starting from i and the edge coming back to i have positive weights. We denote the set of vertices with the exception of i which appear in one of these walks by V^+. We define V^- in the same way. Then we have $V^+ \cup V^- \cup \{i\} = V(w)$. First, we show $V^+ \cap V^- = \emptyset$. Suppose that there exists a vertex $j \in V^+ \cap V^-$. Then, as shown in Figure 5, there are two closed walks $(\{i, i_1^+\}, \Gamma_1^+, \Gamma_2^+, \{i_2^+, i\})$ and $(\{i, i_1^-\}, \Gamma_1^-, \Gamma_2^-, \{i_2^-, i\})$. This implies that we can construct a proper subwalk of w by the combination of $\{i, i_k^+\}, \Gamma_k^+ (k = 1, 2)$, and $\Gamma_l^-, \{i_l^-, i\} (l = 1, 2)$, a contradiction to the primitiveness of w. Therefore $V^+ \cap V^- = \emptyset$. Second, suppose that the removal of the

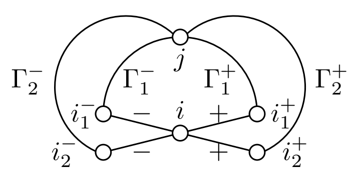

Figure 5: Case that there exists a vertex $j \in V^+ \cap V^-$.

vertex i from G_w does not increase the number of connected components of the remaining subgraph. Then, there are vertices $v^+ \in V^+, v^- \in V^-$ such that $\{v^+, v^-\} \in E(w)$, because $V^+ \cap V^- = \emptyset$ holds as shown above. Hence, as shown in Figure 6, an even closed walk $(\{i, i_k^+\}, \Gamma_k^+, \{v^+, v^-\}, \Gamma_l^-, \{i_l^-, i\})$ is a proper subwalk of w for appropriate $k, l \in \{1, 2\}$, $k \neq l$, which contradicts to the primitiveness of w. Therefore the removal of i from G_w increases the number of connected components of the remaining subgraph. ☐

Lemma 2. *Let an even closed walk w be primitive and $\tilde{W}$ be the weighted graph which is obtained by a contraction for an edge with its weight ± 2 on W. Then any even closed walk $\tilde{w}$ spanning $\tilde{W}$ is primitive.*

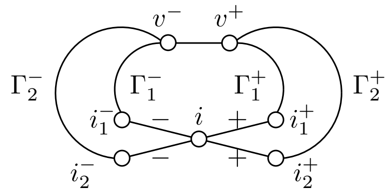

Figure 6: Case that there exists an edge $\{v^+, v^-\}$.

Proof. The contraction of the edge with its weight ± 2 on W is possible from Lemma 1. We denote this edge by $e = \{i, j\}$ as shown in Figure 7. Suppose $\tilde{w}$ is not primitive. Then

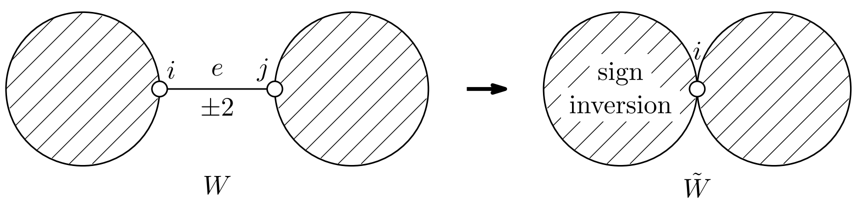

Figure 7: Contraction of an edge e.

there exists an proper subwalk $\tilde{w}'$ of $\tilde{w}$. If $i \notin V(\tilde{w}')$, $\tilde{w}'$ is also a proper subwalk of w, a contradiction to the primitiveness of w. Then $i \in V(\tilde{w}')$. However, a proper subwalk of w is constructed by embedding e into $\tilde{W}'$. Therefore, $\tilde{w}$ is primitive. $\square$

Lemma 3. *Let an even closed walk w be primitive and W_1, W_2 be the weighted graphs obtained by the separation of a vertex i. Then any even closed walks $w_l (l = 1, 2)$ spanning $W_l (l = 1, 2)$ are primitive or of length two with $f_{w_l} = 0$.*

Proof. We consider the case that both positive and negative sides of i correspond to (a) in Figure 3 and relevant edges are labeled as shown in Figure 8. Suppose w_1 is neither primitive nor of length two. Then there exists a proper subwalk w_1' of w_1 on W_1. If

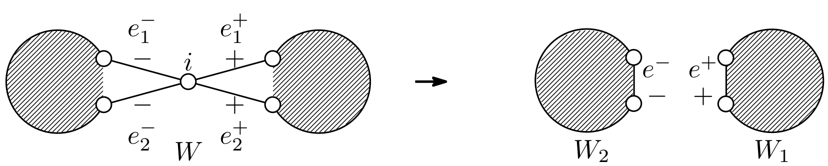

Figure 8: Separation of a vertex i.

$e^+ \notin E(w_1')$, w_1' is also a proper subwalk of w, a contradiction to the primitiveness of w. Then $e^+ \in E(w_1')$. Now w_1' is expressed as follows:

$$w_1' = (e_{i_1}, e_{i_2}, \ldots, e_{i_k}, e^+, e_{i_{k+1}}, \ldots, e_{i_s}).$$

Then an even closed walk on W

$$(e_{i_1}, e_{i_2}, \ldots, e_{i_k}, e_1^+, e_1^-, \ldots, e_2^-, e_2^+, e_{i_{k+1}}, \ldots, e_{i_s})$$

is a proper subwalk of w. This contradicts the primitiveness of w. Therefore w_1 is primitive or of length two. The cases of (b) and (c) in Figure 3 are shown in the same way. Note that it is easy to confirm the possibility of contraction after the step 1 in the case (b) from Lemma 1 and then the primitiveness is guaranteed by Lemma 2. By the same argument, the case of w_2 is confirmed. □

We have so far discussed contraction and separation. From now we show that the inverse operations, insertion and binding, preserve primitiveness.

Lemma 4. *Let w be a primitive walk and let $\tilde{W}$ be the weighted graph obtained by the insertion to i with $deg_W(i) = 4$ on W. Then any even closed walk $\tilde{w}$ spanning $\tilde{W}$ is primitive.*

Proof. Let e be the new edge appearing through the insertion to i as shown in Figure 9. Suppose $\tilde{w}$ is not primitive. Then there exists a proper subwalk $\tilde{w}'$ of $\tilde{w}$. If $e \notin E(\tilde{w}')$,

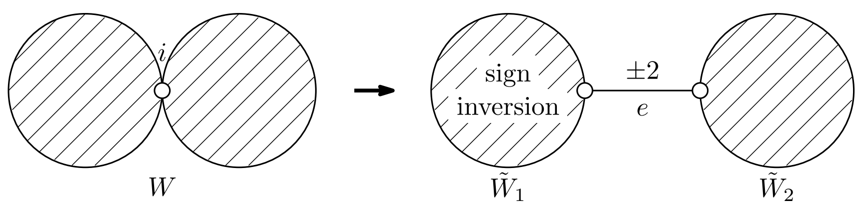

Figure 9: Insertion to a vertex i.

$\tilde{w}'$ is contained in $\tilde{W}_1$ or $\tilde{W}_2$. Then $\tilde{w}'$ or its reverse becomes a proper subwalk of w. This contradicts the primitiveness of w. Hence $e \in E(\tilde{w}')$. Then we can construct a proper subwalk of w by removing e from $\tilde{w}'$ and reversing the weights of edges belonging to $E(w_1)$, a contradiction to the primitiveness of w. Therefore, $\tilde{w}$ is primitive. □

Lemma 5. *Let each w_l $(l = 1, 2)$ be a primitive walk or a closed walk with length two, and W be the weighted subgraph obtained by binding of W_1 and W_2. Then any even closed walk w spanning W is primitive.*

Proof. Let i be the new vertex appearing through the binding. We consider the case that both positive and negative sides of i correspond to (a) in Figure 3 and relevant edges are labeled as shown in Figure 10. Other cases are shown in the same way. Suppose w is

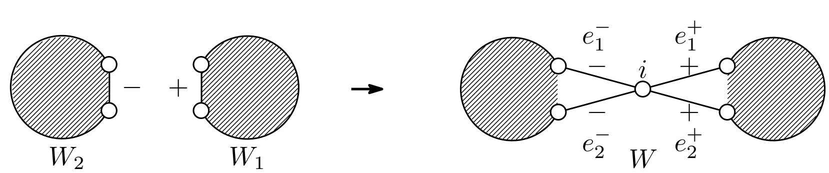

Figure 10: Binding of W_1 and W_2.

not primitive. Then there exists a proper subwalk w' of w. Here we choose a primitive walk as w'. If $i \notin V(w')$, w' is also a proper subwalk of w_1 or w_2. Then $i \in V(w')$. This implies that all four edges connected to i appear in w'. Let us consider the separation of i to W'. Then the resulting two weighted graphs W_1', W_2' are primitive from Lemma 3. Furthermore at least one of w_i' $(i = 1, 2)$ is a proper subwalk of w_i, a contradiction to the primitiveness of w_i. Therefore w is primitive. □

Proof of Proposition 3. Let w be a primitive walk. From Lemma 1 w satisfies (i) in Condition 2 and every edge e with $|\rho(e)| = 2$ can be contracted. Furthermore, it is easy to see that every vertex i with $\deg_W(i) = 4$ can be separated after recursively applying contractions of all possible edges. Therefore $\deg_W(i) = 2$ holds for every vertex i on $\tilde{W}$. From Lemmas 2 and 3, each even closed walk corresponding to the connected component of $\tilde{W}$ is primitive or of length two. Then, every connected component of $\tilde{W}$ is an even cycle or an edge with weight 0, because from Proposition 2 every primitive walk includes a vertex i with $\deg_W(i) = 4$ if it is not an even cycle. Therefore, a primitive walk w satisfies Condition 2. Conversely, suppose an even closed walk w satisfies Condition 2. From Proposition 2 and Lemmas 4 and 5, w is primitive. □

Proof of Theorem 1. Let w be a primitive walk. From Lemma 1, $\#_w(i) \in \{1, 2\}$ holds for each vertex $i \in V(w)$ and $V(w_1^j) \cap V(w_2^j) = \{j\}$ holds for each vertex $j \in V(w)$ with $\#_w(j) = 2$. By the primitiveness of w, the closed walks $w_1^j = (j, \dots, j)$ and $w_2^j = (j, \dots, i_1, \dots, j)$ along w are odd closed walks. Therefore w satisfies Condition 1.

Conversely, let w be an even closed walk with Condition 1. From Proposition 3, it suffices to show that w satisfies Condition 2. The condition (i) in Condition 2 follows from Condition 1. Then, it is enough to confirm that w satisfies the condition (ii) in Condition 2.

First, we claim that every edge $e \in E(w)$ with $|\rho(e)| = 2$ can be contracted and every vertex j with $\#_w(j) = 2$ and $\deg_{G_w}(j) = 4$, i.e. $\deg_{G_w}(j) = \deg_W(j) = 4$, can be separated. The case of contraction is obvious from Condition 1. We confirm the case of separation. Consider the vertex j in Figure 11. If an edge $\{k_1, k_2\}$ dose not exist or exists

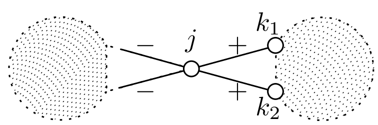

Figure 11: A vertex j with $\deg_{G_w}(j) = \deg_W(j) = 4$.

with weight $+1$, it belongs to the case (a) or (b) in Figure 3, respectively. Let us consider the case that there exists an edge $\{k_1, k_2\}$ with weight -1 and suppose that the vertex k_1 connects to more than three edges as shown in Figure 12. Then, j, k_1 and k_2 appear in w like $(j, k_1, \dots, k_1, k_2, j)$ or $(j, k_1, \dots, k_1, k_2, \dots, k_2, j)$, because $V(w_1^j) \cap V(w_2^j) = \{j\}$ holds. This implies that $(k_1, \dots, k_1)$ is even as shown in Figure 13, which contradicts Condition 1. Hence the case with $\{k_1, k_2\}$ with weight -1 belongs to (c) in Figure 3. Therefore the claim is confirmed.

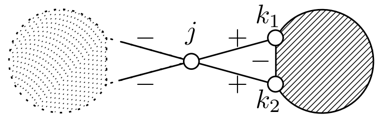

Figure 12: A vertex j which does not exist in w with Condition 1.

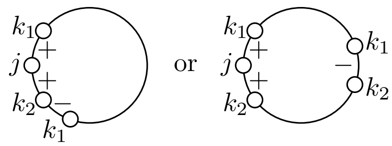

Figure 13: Case that there exists a vertex j in Figure 12.

Second, we verify that contraction and separation on W preserve Condition 1. Consider the case of contraction of an edge $\{i, j\} \in E(W)$. From Condition 1, such i, j appear in w as $w = (i_1, \ldots, i_{l_1}, i, j, i_{l_2}, \ldots, i_{l_3}, j, i, i_{l_4}, \ldots, i_1)$. The contraction of $\{i, j\}$ is equivalent to replacing w by $(i_1, \ldots, i_{l_1}, i, i_{l_2}, \ldots, i_{l_3}, i, i_{l_4}, \ldots, i_1)$. This change causes the decrease of two edges from w, and preserves Condition 1. The case of separation is checked in the same way.

Finally, consider the weighted graph W' obtained by all possible contractions and separations on W. From the claims above, every connected component of W' satisfies Condition 1 and has no vertex j with $\#_w(j) = 2$, i.e. an even cycle or an edge with weight 0. Therefore w satisfies Condition 2. □

4 Algorithms for generating elements of Graver basis

In this section we present two algorithms for generating elements randomly from the Graver basis of a simple undirected graph. Square-free elements of the Graver basis are mainly discussed, because for testing the beta model of random graphs we only need square-free elements of the Graver basis as shown in Proposition 1. Therefore the main objective of this section is to construct an algorithm which generates every square-free element of the Graver basis with positive probability.

First we start with an algorithm based on graph search, which is naturally derived from Theorem 1, and stated as follows.

Algorithm 1 (Algorithm based on graph search)**.**
Input : A simple undirected graph $G = (V(G), E(G))$.
Output : A primitive walk w.

1. *Choose a root vertex $i \in V(G)$ randomly and color it by red.*
2. *Depth first search from i with the following additional rules.*

   - *Each vertex is visited at most twice after starting from i.*
   - *Each edge is used at most once.*

- *Each vertex is colored by red and blue alternately.*
- *When a vertex $j \in V(G)$ is visited twice, distinguish the following two cases:*
  - *If its coloring order is the same as the first visit, then pick up the search path from j to j as an even closed walk w and go to 3.*
  - *Else delete the vertices included in the search path from j to j from candidates to visit after passing j.*
- *If all paths are tested, return to 1.*

3. *Output w.*

It is easy to see that Algorithm 1 can generate every square-free element of the Graver basis with positive probability, because it just searches an even closed walk under Condition 1. However, large backtracks might happen in the search. Then it is desirable to construct a more efficient algorithm.

Next, we discuss another algorithm through a weighted tree with a certain condition introduced below. Let T be a weighted tree $(V(T), E(T), \mu)$ where $\mu : V(T) \to \mathbb{Z}_{\geq 3} = \{3, 4, \dots\}$ is a weight function. For this weighted tree T, let us consider the following condition.

Condition 3. *(i) If $|V(T)| = 1$, the weight of the single vertex is even. (ii) If $|V(T)| > 1$, T satisfies the following two:*

(a) For each leaf $v_T \in V(T)$, $\mu(v_T)$ is odd.

(b) For each vertex $v_T \in V(T)$, $\deg(v_T) \leq \mu(v_T)$ and $\deg(v_T) \equiv \mu(v_T) \mod 2$.

With these tools, let us consider generating a square-free element of the Graver basis of a simple undirected graph $G = (V(G), E(G))$. For simplicity, suppose that G is complete. We will discuss later the case that G is not complete. Let $T = (V(T), E(T), \mu)$ be a weighted tree satisfying Condition 3 and the following equation:

$$\sum_{v_T \in V(T)} \mu(v_T) - |E(T)| \leq |V(G)|. \tag{2}$$

Then, we can construct a primitive walk in G using T as follows. First, we assign the set of vertices $V_{v_T} \subseteq V(G)$ with $|V_{v_T}| = \mu(v_T)$ for each vertex $v_T \in V(T)$ under the equation

$$|V_{v_T} \cap V_{v'_T}| = \begin{cases} 1, & \text{if } \{v_T, v'_T\} \in E(T), \\ 0, & \text{if } \{v_T, v'_T\} \notin E(T), \end{cases} \qquad (v'_T \in V(T))$$

and every vertex $v \in V(G)$ is assigned at most twice. Equation (2) guarantees that this assignment is possible. Second, we make cycles in G by arbitrarily ordering the vertices V_{v_T}. Then we make a subgraph of G by taking the union of these cycles. Finally, we obtain a closed walk by choosing a root vertex from this subgraph and going around it. It is easy to see that this closed walk is primitive by Theorem 1.

Conversely we can construct a weighted tree with Condition 3 and (2) from each primitive walk. Let w be a square-free primitive walk. First, the vertex set $V(T)$ is constructed by creating a vertex v_c of T for each cycle c in G_w. Second, the edge set $E(T)$ is obtained by adding edge $\{v_c, v_{c'}\}$ to $E(T)$ for each pair of cycles c, c' in G_w with $V(c) \cap V(c') \neq \emptyset$. Then, we assign weight $\mu(v_c) := |V(c)|$ to each vertex $v_c \in V(T)$.

Therefore, once we have a weighted tree T with Condition 3 and (2), we can construct a square-free element of the Graver basis of G. Such a tree T is constructed by the following algorithm.

Algorithm 2 (Algorithm for constructing an weighted tree)**.**
Input : A complete graph $G = (V(G), E(G))$.
Output : A weighted tree $T = (V(T), E(T), \mu)$ with Condition 3 and (2).

1. *Let $V(T), E(T)$ be empty sets and $n := |V(G)|$.*
2. *Add a root vertex r to $V(T)$.*
3. *Assign $\mu(r)$ a weight from $\{3, 4, \dots, n\}$ randomly.*
4. *Grow T by the following loop.*
   - (a) *For each vertex $v_T \in V(T)$ which is deepest from r, add edges $\{v_T, v_T^i\}$ to $E(T)$ and the endpoints v_T^i $(i = 0, 1, \dots, I_{v_T})$ to $V(T)$, where the number I_{v_T} is randomly decided under the following two conditions:*
     - $I_{v_T} + 1 \leq \mu(v_T)$.
     - $I_{v_T} + 1 \equiv \mu(v_T) \mod 2$.
   - (b) *For each new vertex v_T^i, assign $\mu(v_T^i)$ a weight from $\{3, 4, \dots, n-\alpha\}$ randomly, where $\alpha := \sum_{v_T \in V(T)} \mu(v_T) - |E(T)|$.*
   - (c) *Recompute α and if $\alpha > n$, delete all new vertices and edges in the above (a) and break the loop.*
   - (d) *If the total number of new edges is equal to 0, break the loop.*
   - (e) *Return to (a).*
5. *If $|V(T)| = 1$ and $\mu(r)$ is odd, change $\mu(r)$ to $\mu(r) - 1$ or $\mu(r) + 1$.*
6. *If $|V(T)| > 1$ and T has a leaf with even weight, subtract or add 1 to the weight.*
7. *Output T.*

Algorithm 2 provides a simple algorithm for generating a square-free element of Graver basis as follows.

Algorithm 3 (Algorithm through a weighted tree)**.**
Input : A complete graph $G = (V(G), E(G))$.
Output : A primitive walk w.

1. *Construct a weighted graph T with Condition 3 and (2) by Algorithm 2.*

2. *Construct a primitive walk by assigning vertices of G and ordering them randomly.*
3. *Output w.*

Since there is no restarts in Algorithm 3, it has a fixed worst case running time. In each step, the algorithm performs $O(|V(G)|)$ operations. Then it generates one element of the Graver basis of G in $O(|V(G)|)$ time.

A demonstration for the case of a complete graph G with $|V(G)| = 25$ is shown in Figures 14 and 15.

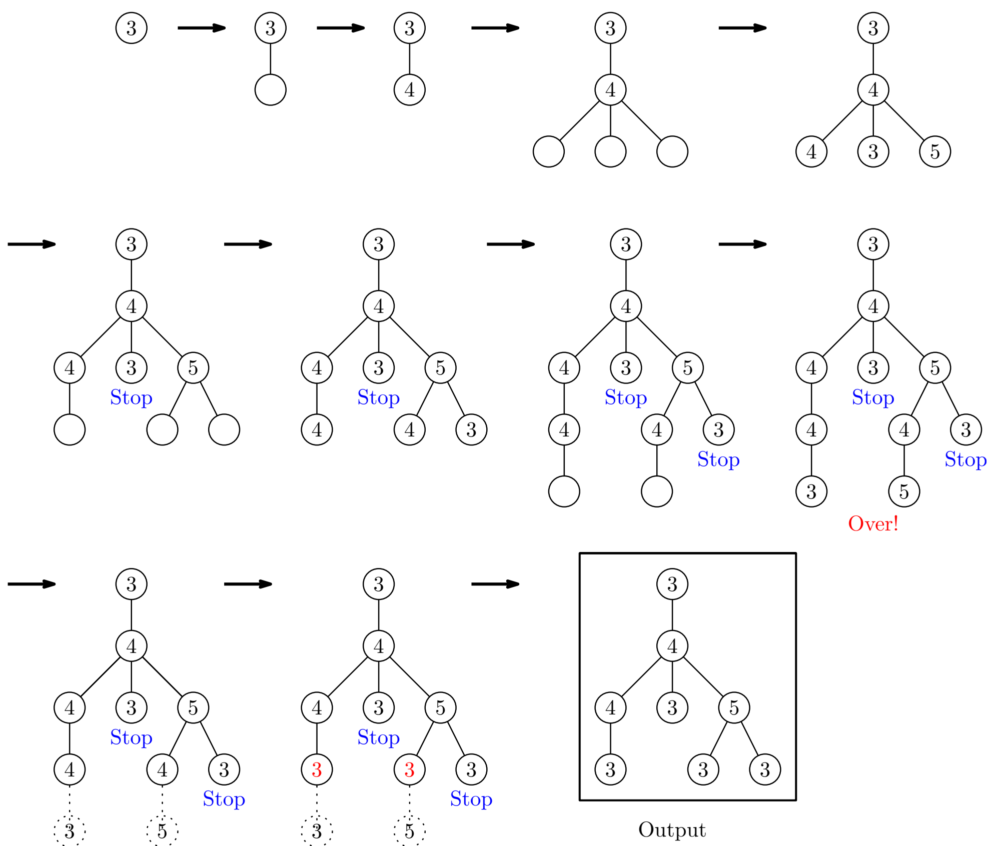

Figure 14: Demonstration of Algorithm 2.

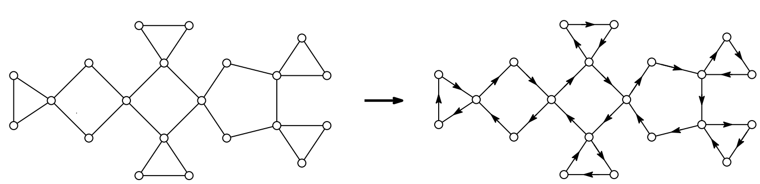

Figure 15: Demonstration of Algorithm 3.

Remark 3. *For the case that an input graph G is not complete, the elements of the Graver basis of G can be generated by throwing away elements with supports not contained in G*

(Proposition 4.13 of Sturmfels [20]). In fact this is the advantage of considering the Graver basis. General non-square-free elements of the Graver basis can be generated by Algorithm 3 with a slight modification. In fact, it suffices to change merely $\{3, 4, \ldots\}$ *to* $\{2, 3, 4, \ldots\}$ *in Step 3 and in (b) of Step 4 in Algorithm 2.*

Remark 4. *The output of Algorithm 3 is not uniformly distributed over all square-free elements of Graver basis. The distribution depends on how to implement the randomness in Step 3 and in (b) of Step 4 in Algorithm 2.*

5 Numerical experiments

In this section we present numerical experiments with elements of the Graver basis computed by Algorithm 3 in Section 4.

5.1 A simulation with a small graph

We shall show that Algorithm 3 allows us to uniformly sample graphs with the common degree sequence. It is done by constructing a connected Markov chain of graphs through Metropolis-Hastings method with the Graver basis as follows. In each iteration, a primitive walk is randomly generated by Algorithm 3. If the primitive walk is applicable, a new subgraph with the same degree sequence is obtained by adding the primitive walk, otherwise reject the primitive walk.

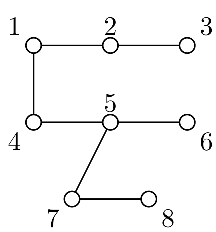

Figure 16: Small graph H_0.

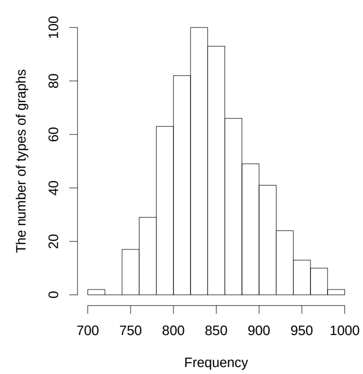

Figure 17: Histogram from sampling.

We run a Markov chain over the fiber containing a small graph H_0 in Figure 16. The underlying graph $G = K_8$ is assumed to be complete with eight vertices. By the Markov chain we sampled 510,000 graphs in the fiber, including 10,000 burn-in steps by 8,760,926 iterations. The number of types of obtained graphs in our chain is 591. By enumeration we checked that 591 is actually the number of the elements of the fiber of H_0. The histogram of this experiment is shown in Figure 17. The horizontal axis expresses the frequency of each type of graph and the vertical axis expresses the number of types. The mean of the number of appearances of each type is 846 and the standard deviation is 50. This experiment shows that the algorithm samples each element of the fiber almost uniformly.

5.2 The beta model for the food web data

We apply Algorithm 3 for testing of the real data, the observed food web of 36 types of organisms in the Chesapeake Bay during the summer. This data is available online at [21]. Blitzstein and Diaconis [2] analyzed essentially the same data set.

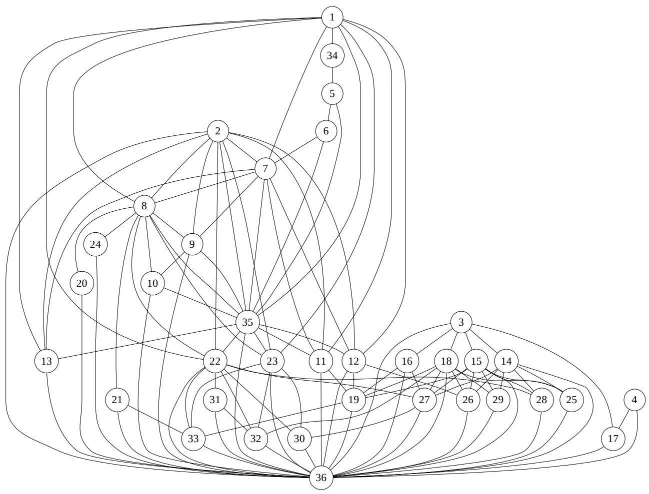

Figure 18: Food web for the Chesapeake Bay during the summer.

The graph H of the data is shown in Figure 18. The vertices of the graph represent the types of organisms like blue crab, bacteria etc., and the edges represent the relationship of one preying upon the other. The degree sequence of H is

$$(9, 10, 6, 2, 3, 3, 9, 11, 6, 4, 6, 7, 5, 7, 8, 4, 3, 8, 7, 2, 3, 11, 8, 2, 4, 5, 7, 4, 4, 4, 3, 5, 5, 2, 14, 29).$$

Although there is a self loop at the vertex 19 in the observation, we ignored it for simplicity.

We set the beta model (1) in Section 2 with $n_{ij} = 1$ for each edge $\{i, j\}$ as the null hypothesis. Then the probability of H is described as

$$P(H) \propto \frac{\prod_{i \in V} \alpha_i^{d_i}}{\prod_{\{i,j\} \in E}(1 + \alpha_i \alpha_j)}. \tag{3}$$

Parameter α_i ($i \in V$) is interpreted as the value of organism represented by the vertex i as a food to other organisms. Then the beta model (3) implies that a vertex i with large α_i is likely to be connected to many edges. Let $P \in (3)$ mean that P can be expressed by (3) for a set of parameters $\{\alpha_i\}_{i \in V}$. Consider now the statistical hypothesis testing problem

$$H_0 : P \in (3) \text{ versus } H_1 : P \notin (3).$$

Starting from the graph in Figure 18, we construct a Markov chain of 510,000 graphs including 10,000 burn-in steps by 167,277,350 iterations and compute the chi-square statistic of each graph as a test statistic. Using the maximum likelihood estimator, the chi-square value of observed graph H is 477 and the histogram of the estimated distribution of the chi-square values is shown in Figure 19. The approximate p-value is 0.278. This value is not so small and there is no evidence against the beta model (3).

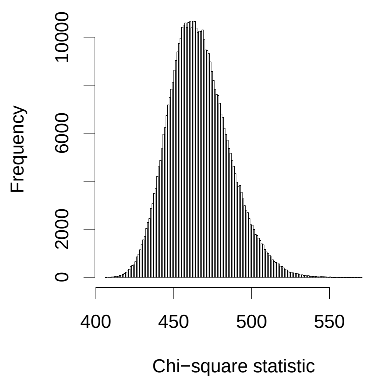

Figure 19: Histogram of the chi-square statistic.

6 Concluding remarks

In this paper we obtained a simple characterization of the Graver basis for toric ideals arising from undirected graphs. This Graver basis allows us to perform the conditional test of the beta model for arbitrary underlying graph. Our characterization allows us to construct an algorithm for sampling elements of the Graver basis, which is sufficient for performing the conditional test.

By numerical experiments we confirmed that our procedure works well in practice. We should mention that the sequential importance sampling method of Blitzstein and Diaconis [2] may work faster for the case of complete underlying graph.

If we allow multiple edges, then we do not need the Graver basis. A minimal Markov basis, which is often much smaller than the Graver basis, is sufficient for connectivity of Markov chains. Properties of Markov basis for the p_1-model have been given in Petrović et al. [16]. It is of interest to study properties of minimal Markov bases for undirected graphs, including the case of allowing self loops.

Acknowledgment

We are very grateful to Hidefumi Ohsugi for valuable discussions.

References

[1] 4ti2 team. 4ti2—a software package for algebraic, geometric and combinatorial problems on linear spaces. Available at www.4ti2.de.

[2] Joseph Blitzstein and Persi Diaconis. A sequential importance sampling algorithm for generating random graphs with prescribed degrees. Available at http://www.people.fas.harvard.edu/~blitz/BlitzsteinDiaconisGraphAlgorithm.pdf, 2006. Preprint.

[3] Sourav Chatterjee, Persi Diaconis, and Allan Sly. Random graphs with a given degree sequence. arXiv:1005.1136v4, 2010.

[4] Persi Diaconis and Bernd Sturmfels. Algebraic algorithms for sampling from conditional distributions. *Ann. Statist.*, 26(1):363–397, 1998. ISSN 0090-5364.

[5] Mathias Drton, Bernd Sturmfel, and Seth Sullivant. *Lectures on Algebraic Statistics*. Oberwolfach Seminars. Birkhäuser Basel, 2008.

[6] P. Erdős and A. Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

[7] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2:129–233, 2009.

[8] Hisayuki Hara and Akimichi Takemura. Connecting tables with zero-one entries by a subset of a markov basis. In *Algebraic Methods in Statistics and Probability II*, volume 516 of *Contemp. Math.*, pages 199–213. Amer. Math. Soc., Providence, RI, 2010.

[9] Paul Holland and Samuel Leinhardt. An exponential family of probability distribution for directed graphs. *J. Amer. Statist. Soc.*, 76(373):33–50, 1981.

[10] J. M. Linacre. *Many-facet Rasch Measurement*. MESA Press, Chicago, 1989.

[11] Mark E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[12] Hidefumi Ohsugi and Takayuki Hibi. Koszul bipartite graphs. *Adv. in Appl. Math.*, 22(1):25–28, 1999. ISSN 0196-8858.

[13] Hidefumi Ohsugi and Takayuki Hibi. Toric ideals generated by quadratic binomials. *J. Algebra*, 218(2):509–527, 1999. ISSN 0021-8693.

[14] Hidefumi Ohsugi and Takayuki Hibi. Indispensable binomials of finite graphs. *J. Algebra Appl.*, 4(4):421–434, 2005. ISSN 0219-4988.

[15] Juyong Park and Mark E. J. Newman. The statistical mechanics of networks. *Phys. Rev. E*, 70:066117, 2004.

[16] Sonya Petrović, Alessandro Rinaldo, and Stephen E. Fienberg. Algebraic statistics for a directed random graph model with reciprocation. In *Algebraic Methods in Statistics and Probability II*, volume 516 of *Contemp. Math.*, pages 261–283. Amer. Math. Soc., Providence, RI, 2010.

[17] G. Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests*. University of Chicago Press, Chicago, 1980.

[18] Enrique Reyes, Christos Tatakis, and Apostolos Thoma. Minimal generators of toric ideals of graphs. arXiv:1002.2045v1, 2010.

[19] R. Solomonoff and A. Rapoport. Connectivity of random nets. *Bulletin of Mathematical Biophysics*, 13:107–117, 1951.

[20] Bernd Sturmfels. *Gröbner Bases and Convex Polytopes*, volume 8 of *University Lecture Series*. American Mathematical Society, Providence, RI, 1996. ISBN 0-8218-0487-1.

[21] Robert E. Ulanowicz. Ecosystem network analysis web page, 2005. URL `http://www.cbl.umces.edu/~{}ulan/ntwk/network.html`.