Towards random uniform sampling of bipartite graphs with given degree sequence *†

István Miklós ‡ Péter L. Erdős § Lajos Soukup ¶

Alfréd Rényi Institute of Mathematics,
Hungarian Academy of Sciences,
Budapest, P.O. Box 127, H-1364 Hungary

<miklosi,elp,soukup>@renyi.hu

Submitted: Jan 24, 2011; Accepted: Jan 8, 2013; Published: Jan 21, 2013
Mathematics Subject Classifications: 05C85 60J10 94C15 05D40

Abstract

In this paper we consider a simple Markov chain for bipartite graphs with given degree sequence on n vertices. We show that the mixing time of this Markov chain is bounded above by a polynomial in n in case of *half-regular* degree sequence. The novelty of our approach lies in the construction of the multicommodity flow in Sinclair's method.

1 Introduction

The *degree sequence*, $d(G)$, of a graph G is the non-increasing sequence of its vertex degrees. A sequence $\mathbf{d} = (d_1, \ldots, d_n)$ is *graphical* iff $d(G) = \mathbf{d}$ for some simple graph G, and G is a *graphical realization* of $\mathbf{d}$.

Already at the beginning of the systematic graph theoretical research (late fifties and early sixties) there were serious efforts to decide whether a non-increasing sequence is graphical. Erdős and Gallai (1960, [3]) gave a necessary and sufficient condition, while Havel (1955, [6]) and Hakimi (1962, [5]) independently developed a greedy algorithm to built a graphical realization if there exists any. (For more details see for example [8].)

*This research was supported in part by the Hungarian Bioinformatics MTKD-CT-2006-042794, Marie Curie Host Fellowships for Transfer of Knowledge.

†IM and PLE acknowledge financial support from grant #FA9550-12-1-0405 from the U.S. Air Force Office of Scientific Research (AFOSR) and the Defense Advanced Research Projects Agency (DARPA).

‡Partly supported by Hungarian NSF, under contract Nos. NK 78439 and PD84297

§Partly supported by Hungarian NSF, under contract Nos. NK 78439 and K68262

¶Partly supported by Hungarian NSF, under contract Nos. NK 83726 and K68262

Generating some (or all possible) graphs realizing a given degree sequence or finding a typical one among the different realizations are ubiquitous problems in network modeling, ranging from social sciences to chemical compounds and biochemical reaction networks in the cell. (See for example the book [10] for a detailed analysis, or the paper [8] for a short explanation.)

When the number of different realizations is small, then the uniform sampling of the different realizations can be carried out by generating all possible ones and choosing among them uniformly.

However in cases where there are many different realizations this approach can not work. In these cases some stochastic processes can provide solutions. Here we mention only one of the preceding results: Molloy and Reed (1995, [9]) applied the *configuration model* (Bollobás (1980, [1]) for the problem. (In fact, Wormald had used it already in 1984 to generate random regular graphs of *moderate* degrees [14].) They successfully used the model to generate random graphs with given degree sequences where the degrees are (universally) bounded. It is well known that this method is computationally infeasible in case of general, unbounded degree sequences.

A different method was proposed by Kannan, Tetali and Vempala (1995, [7]), which is based on the powerful Metropolis-Hastings algorithm: some local transformation generates a random walk on the family of all realizations. They conjectured that this process is *rapidly mixing* i.e. starting from an arbitrary realization of the degree sequence the process reaches a completely random realization in reasonable (i.e. polynomial) time. However, they could prove it only for bipartite regular graphs. Their conjecture was proved for arbitrary regular graphs by Cooper, Dyer and Greenhill (2007, [2]).

The original goal of this paper was to attack Kannan, Tetali and Vempala's conjecture for arbitrary bipartite degree sequences, performing a more subtle choice of *multicommodity flow*. We obtained the following result:

Theorem 1.1. *The Markov process - defined by Kannan, Tetali and Vempala - is rapidly mixing on each bipartite half-regular degree sequence.* (*In these bipartite graphs the degrees in one vertex class are constant.*)

Actually, we achieved somewhat more: our construction method can be used as a plug-in to a more advanced method for general degree sequences: if two particular graphical realizations at hand differ in edges which can be partitioned into alternating cycles, such that no cycle contains a chord which is an edge of another cycle in the partition, then our *friendly path* method provides a good multicommodity flow.

2 Basic definitions and preliminaries

Let $G = (U, V; E)$ be a simple bipartite graph (no parallel edges) with vertex classes $U = \{u_1, \dots, u_k\}$, $V = \{v_1, \dots, v_l\}$. The *(bipartite) degree sequence* of G, $\mathrm{bd}(G)$ is defined as follows:

$$\mathrm{bd}(G) = \Big(\big(d(u_1), \dots, d(u_k)\big), \big(d(v_1), \dots, d(v_l)\big)\Big),$$

where the vertices are ordered such that both sequences are non-increasing. From now on when we say "degree sequence" of a bipartite graph, we will always mean the bipartite degree sequence. We will use n to denote the number of vertices, that is $n = k + l$.

A pair $(\mathbf{a}, \mathbf{b})$ of sequences is a (*bipartite*) *graphical sequence* (BGS for short) if $(\mathbf{a}, \mathbf{b}) = \mathrm{bd}\,(G)$ for some simple bipartite graph G, while the graph G is a *(graphical) realization* of $(\mathbf{a}, \mathbf{b})$.

Next we define the swaps, our basic operation on bipartite graphs.

Definition 2.1. Let $G = (U, V; E)$ be a bipartite graph, $u_1, u_2 \in U$, $v_1, v_2, \in V$, such that induced subgraph $G[u_1, u_2; v_1, v_2]$ is a 1-factor, (i.e. $(u_1, v_j), (u_2, v_{3-j}) \in E$, but $(u_1, v_{3-j}), (u_2, v_j) \notin E$ for some j.) Then we say that the **swap on** $(u_1, u_2; v_1, v_2)$ is **allowed**, and it transforms the graph G into a graph $G' = (U, V; E')$ by replacing the edges $(u_1, v_j), (u_2, v_{3-j})$ by edges (u_1, v_{3-j}) and (u_2, v_j), i.e.

$$E' = E \setminus \{(u_1, v_j), (u_2, v_{3-j})\} \cup \{(u_1, v_{3-j}), (u_2, v_j)\}. \tag{2.1}$$

So a swap transforms one realization of the BGS to another (bipartite graph) realization of the same BGS. The following proposition is a classical result of Ryser (1957, [11]).

Theorem 2.2 (Ryser)**.** *Let $G_1 = (U, V; E_1)$ and $G_2 = (U, V; E_2)$ be two realizations of the same BGS. Then there exists a sequence of swaps which transforms G_1 into G_2 through different realizations of the same BGS.*

Ryser's result used the language of 0 - 1 matrices. Here, to make the paper self contained, we give a short proof, using the notion of swaps. The proof is based on a well known observation of Havel and Hakimi ([6, 5]):

Lemma 2.3 (Havel and Hakimi)**.** *Let $G = (U, V; E)$ be a simple bipartite graph, and assume that $d(u') \leqslant d(u)$, furthermore $(u', v) \in E$ and $(u, v) \notin E$. Then there exists a vertex v' such that the swap on $(u, u'; v, v')$ is allowed, and so it produces a bipartite graph G' from G such that $\Gamma_{G'}(v) = (\Gamma_G(v) \setminus \{u'\}) \cup \{u\}$, where, as usual, $\Gamma_G(v)$ is the set of neighbors of v in G.*

Proof: By the pigeonhole principle there exists a vertex $v' \neq v$ such that $(u, v') \in E$ and $(u', v') \notin E$. So the swap defined on vertices $(u, u'; v, v')$ is allowed. □

We say that the previous operation is *pushing up* the neighbors of vertex v. Applying the pushing up operation d times we obtain the following push up lemma.

Lemma 2.4 (Havel and Hakimi)**.** *If $G = (U, V; E)$ is a simple bipartite graph, $d(u_1) \geqslant d(u_2) \geqslant \ldots \geqslant d(u_k)$ and $v \in V$, $d = d(v)$. Then there is a sequence S of d many swaps which transforms G into a graph G' such that $\Gamma_{G'}(v) = \{u_1, \ldots, u_d\}$.*

This pushing-up lemma also suggests (and proves the correctness of) a greedy algorithm to construct a concrete realization of a BGS $\big(\mathbf{a}, \mathbf{b}\big)$.

Proof of Theorem 2.2: We prove the following stronger statement:

(✠) *there exists a sequence of* $2e$ *swaps which transforms* G_1 *into* G_2*, where* e *is the number of edges of* G_i*.*

We will show that any particular realization can be transformed into the same *canonical realization* with at most e swaps. We will do it recursively: taking one by one the vertices $v_1, v_2, \ldots, v_l$ from V we will define their neighbors in U. After every step of the process we update the remaining degree sequence of U, and reorder its actual content.

To do so we introduce the following lexicographic order on the actual remaining $d(u)$ degree sequence. We always take them non-increasing order, and whenever two vertices have the same actual degree, then we take first the vertex with bigger subscript.

So take v_1 and by multiple applications of the Push-up Lemma 2.4 there is a sequence T_1 of at most $d = d(v_1)$ many swaps which transforms G_1 into a G_1' such that $\Gamma_{G_1'}(v_1) = \{u_1, \ldots, u_d\}$ (The actually required push up operations can be smaller if some of the first d vertices were originally adjacent to v_1.)

We consider the bipartite graphs $G_1'' = G_1' \setminus \{v_1\}$ i.e. we remove the vertex v_1 and all the edges connected to v_1. Now we reorder the vertices in the actual U according to our lexicographic order, and repeat the recursive operation.

In this way after at most $\sum_{i=1}^{l} d(v_i) = e$ swaps we transformed G_1 into a well defined canonical realization R, furthermore this R is independent from the original realization.

Now we can easily finish the proof of Theorem 2.2 observing that if a swap transforms H into H', then the "inverse swap" (choosing the same four vertices, and changing back the edges) transforms H' into H. So if the swap sequence T_1 transforms G_1 into R then it has an inverse swap-sequence T_1' which transforms R into G_1. □

We use this upper bound for convenience: for us a linear upper bound on this value is enough to show the polynomial upper bound of the sampling process. If somebody wanted to get tight (or at least better) upper bounds on the sampling process, then a better estimation is necessary for the swap-distance. Recently it was shown that the swap-distance $\mathbf{dist}(G_1, G_2)$ for any two realizations is smaller than

$$\Delta := \frac{1}{2}\big|E(G_1)\Delta E(G_2)\big|.$$

In the forthcoming paper [4] a formula for $\mathbf{dist}(G_1, G_2)$ is determined: this is in the form of $\Delta - \alpha$ where the parameter $\alpha \geqslant 1$. Unfortunately the parameter α is hard to determine.

3 The Markov chain $(\mathbb{G}, P)$

For a bipartite graphical sequence $\big(\mathbf{a}, \mathbf{b}\big)$ (on the fixed vertex bipartition (U, V)) - following Kannan, Tetali and Vempala's lead - we define a Markov chain $(\mathbb{G}, P)$ in the following way. $\mathbb{G}$ is a graph, the vertex set $V(\mathbb{G})$ of the graph $\mathbb{G}$ consists of all possible realizations of our BGS, while the edges represent the possible swap operations: two realizations are connected if there is a swap operation which transforms one realization into the other one (and, recall, the inverse swap transforms the second one to the first one as well).

Let P denote the *transition matrix*, which is defined as follows: if the current realization (state of the process) is G then with probability $\frac{1}{2}$ we stay in the current state (namely, we define a lazy Markov chain) and with probability $\frac{1}{2}$ we choose uniformly two-two vertices $u_1, u_2; v_1, v_2$ from classes U and V respectively and perform the swap if it is possible and move to G'. Otherwise we do not perform a move. The swap moving from G to G' is unique, therefore the probability of this transformation (the *jumping probability* from G to $G' \neq G$) is:

$$\mathrm{Prob}(G \to G') := P(G'|G) = \frac{1}{2\binom{k}{2}\binom{l}{2}}. \tag{3.1}$$

The probability of transforming G' to G is time-independent. The transition probabilities are *time* and edge *independent* and they are also *symmetric*. Therefore P is a symmetric matrix, where all off-diagonal, non-zero elements are the same, while the entries in the main-diagonal are non-zero, but (probably) different values.

We use the convention that upper case letters X, Y and Z stands for vertices of $V(\mathbb{G})$.

The graph $\mathbb{G}$ clearly may have exponentially many vertices (that many different realizations of the degree sequence). However, by the statement (✠) (in the proof of Theorem 2.2), its diameter is always relatively small:

Corollary 3.1. *The* swap distance *of any two realizations is at most* $2e$*, where* e *is the number of edges.*

As we observed, the graph $\mathbb{G}$ is connected, therefore the Markov process is *irreducible*. Since our Markov chain is lazy, it is clearly aperiodic. Finally since, as we saw, the jumping probabilities are symmetric, that is $P(G|G') = P(G'|G)$, therefore our lazy Markov process is reversible with the uniform distribution as the globally stable stationary distribution.

4 Sinclair's Method

To start with we recall some definitions and notations from the literature. Since our Markov chain converges to the uniform distribution, we write all theorems for the special uniform distribution case even if the theorem holds for more general distribution, to simplify the notations. Let P^t denote the tth power of the transition probability matrix and define

$$\Delta_X(t) := \frac{1}{2} \sum_{Y \in V(\mathbb{G})} \left| P^t(Y|X) - 1/N \right|,$$

where X is an element of the state space of the Markov chain and N is the size of the state space. We define the *mixing time* as

$$\tau_X(\varepsilon) := \min_t \left\{ \Delta_X(t') \leqslant \varepsilon \text{ for all } t' \geqslant t \right\}.$$

Our Markov chain is said to be *rapidly mixing* iff

$$\tau_X(\varepsilon) \leqslant O\Big(\mathrm{poly}\big(log(N/\varepsilon) \big) \Big)$$

for any X in the state space. Consider the different eigenvalues of P in non-increasing order:

$$1 = \lambda_1 > \lambda_2 \geqslant \ldots \geqslant \lambda_N \geqslant -1.$$

The *relaxation time* τ_{rel} is defined as

$$\tau_{rel} = \frac{1}{1-\lambda^*}$$

where λ^* is the *second largest eigenvalue modulus*,

$$\lambda^* := \max\{\lambda_2, |\lambda_N|\}.$$

However, the eigenvalues of any lazy Markov chain are non-negative, so we do know that $\lambda^* = \lambda_2$ for our Markov chain. The following result was proved implicitly by Diaconis and Strook in 1991, and explicitly stated by Sinclair: [12, Theorem 5']

Theorem 4.1 (Sinclair)**.** $\tau_x(\varepsilon) \leqslant \tau_{rel} \cdot \mathrm{poly}\big(log(N/\varepsilon)\big)$. □

So one way to prove that our Markov chain is rapidly mixing is to find a polynomial upper bound on τ_{rel}. We need rapid convergence of the process to the stationary distribution otherwise the method cannot be used in practice.

Kannan, Tetali and Vempala in [7] could prove that the relaxation time of the Markov chain $(\mathbb{G}, P)$ is a polynomial function of the size $n := 2k$ of $\big(\mathbf{a}, \mathbf{b}\big)$ if it is a regular bipartite degree sequence. Here we extend their proof to show that the process is rapidly mixing for the half-regular bipartite case.

There are several different methods to prove fast convergence, here we use - similarly to [7] - Sinclair's *multicommodity flow method* ([12]).

Theorem 4.2. *Let $\mathbb{H}$ be a graph whose vertices represent the possible states of a time reversible finite state Markov chain $\mathcal{M}$, and where $(U, V) \in E(\mathbb{H})$ iff the transition probabilities of $\mathcal{M}$ satisfy $P(U|V)P(V|U) \neq 0$. For all $X \neq Y \in V(\mathbb{H})$ let $\Gamma_{X,Y}$ be a set of paths in $\mathbb{H}$ connecting X and Y and let $\pi_{X,Y}$ be a probability distribution on $\Gamma_{X,Y}$. Furthermore let*

$$\Gamma := \bigcup_{X \neq Y \in V(\mathbb{H})} \Gamma_{X,Y}$$

where the elements of Γ are called paths*. We also assume that there is a stationary distribution π on the vertices $V(\mathbb{H})$. We define the capacity of an edge $e = (W, Z)$ as*

$$Q(e) := \pi(W)P(Z|W)$$

and we denote the length of a path γ by $|\gamma|$. Finally let

$$\kappa_\Gamma := \max_{e \in E(\mathbb{H})} \frac{1}{Q(e)} \sum_{\substack{X,Y \in V(\mathbb{H}) \\ \gamma \in \Gamma_{X,Y} \,:\, e \in \gamma}} \pi(X)\pi(Y)\pi_{X,Y}(\gamma)|\gamma|. \tag{4.1}$$

Then

$$\tau_{\mathrm{rel}}(\mathcal{M}) \leqslant \kappa_\Gamma \tag{4.2}$$

holds. □

We are going to apply Theorem 4.2 for our Markov chain $(\mathbb{G}, P)$. Using the notation $|V(\mathbb{G})| := N$, the (uniform) stationary distribution has the value $\pi(X) = 1/N$ for each vertex $X \in V(\mathbb{G})$. Furthermore each transition probability has the property $P(X|Y) \geqslant 1/n^4$ (recall that $n = k + l$, that is n denotes the number of the vertices of any realization). So if we can design a multicommodity flow such that each path is shorter then an appropriate poly(n) function, then simplifying inequality (4.1) we can turn inequality (4.2) to the form:

$$\tau_{\mathrm{rel}} \leqslant \frac{\mathrm{poly}(n)}{N} \left(\max_{e \in E(\mathbb{H})} \sum_{\substack{X,Y \in V(\mathbb{H}) \\ \gamma \in \Gamma_{X,Y} \,:\, e \in \gamma}} \pi_{\mathrm{X,Y}}(\gamma) \right). \tag{4.3}$$

If $Z \in e$, then

$$\sum_{\substack{X,Y \in V(\mathbb{H}) \\ \gamma \in \Gamma_{X,Y} \,:\, e \in \gamma}} \pi_{\mathrm{X,Y}}(\gamma) \leqslant \sum_{\substack{X,Y \in V(\mathbb{H}) \\ \gamma \in \Gamma_{X,Y} \,:\, Z \in \gamma}} \pi_{\mathrm{X,Y}}(\gamma), \tag{4.4}$$

so we have

$$\tau_{\mathrm{rel}} \leqslant \frac{\mathrm{poly}(n)}{N} \left(\max_{Z \in V(\mathbb{H})} \sum_{\substack{X,Y \in V(\mathbb{H}) \\ \gamma \in \Gamma_{X,Y} \,:\, Z \in \gamma}} \pi_{\mathrm{X,Y}}(\gamma) \right). \tag{4.5}$$

We make one more assumption. Namely, that for each $X, Y \in V(\mathbb{G})$ there is a non-empty finite set $S_{X,Y}$ (which draws its elements from a pool of symbols) and for each $s \in S_{X,Y}$ there is a path $\Upsilon(X, Y, s)$ from X to Y such that

$$\Gamma_{X,Y} = \{\Upsilon(X, Y, s) : s \in S_{X,Y}\}. \tag{4.6}$$

It can happen that $\Upsilon(X, Y, s) = \Upsilon(X, Y, s')$ for $s \neq s'$, so we consider $\Gamma_{X,Y}$ as a "multiset" and so we should take

$$\pi_{X,Y}(\gamma) = \frac{\left|\left\{s \in S_{X,Y} : \gamma = \Upsilon(X, Y, s)\right\}\right|}{|S_{X,Y}|}$$

for $\gamma \in \Gamma_{X,Y}$.

Putting together the observations and simplifications above we obtain the
Simplified Sinclair's method:
For each $X \neq Y \in V(\mathbb{G})$ find a non-empty finite set $S_{X,Y}$ and for each $s \in S_{X,Y}$ find a path $\Upsilon(X, Y, s)$ from X to Y such that

- each path is shorter than an appropriate poly(n) function,
- for each $Z \in V(\mathbb{G})$

$$\sum_{X,Y \in V(\mathbb{G})} \frac{\left|\left\{s \in S_{X,Y} : Z \in \Upsilon(X, Y, s)\right\}\right|}{|S_{X,Y}|} \leqslant poly(n) \cdot N. \tag{4.7}$$

Then our Markov chain $(\mathbb{G}, P)$ is rapidly mixing.

5 Multicommodity flow - general considerations

Our construction method for multicommodity flow commences on the trail of Kannan, Tetali and Vempala ([7]), and Cooper, Dyer and Greenhill ([2]). However the main difference among these papers lies in the method of the construction of the multicommodity flow.

We fix a bipartite graphical sequence $\big(\mathbf{a},\mathbf{b}\big)$, and consider the graph $\mathbb{G}$ where the vertices of $\mathbb{G}$ are the realizations of $\big(\mathbf{a},\mathbf{b}\big)$, while the edges correspond to the possible swap operations. Therefore if $X \in \mathbb{G}$, then X is a simple bipartite graph $(U, V; E(X))$, where U and V are fixed finite sets.

We can outline the construction of the path system from $X \in \mathbb{G}$ to $Y \in \mathbb{G}$ as follows:

(Step 1) We decompose the symmetric difference Δ of $E(X)$ and $E(Y)$ into alternating circuits:

$$W_1, W_2 \ldots, W_{k_s}.$$

The construction uses the method of [2] to parameterize all the possible decompositions (see Subsection 5.1). Roughly speaking, the parameter set $S_{X,Y}$ will be the collection of all pairings of edges $E(X) \setminus E(Y)$ and $E(Y) \setminus E(X)$ adjacent to w, for all $w \in U \cup V$.

(Step 2) We decompose every alternating circuit W_i into alternating cycles

$$C_1^i, C_2^i \ldots, C_{k_i}^i,$$

and we will construct the canonical path from X to Y in such a way that first we switch the edges $E(X) \setminus E(Y)$ and $E(Y) \setminus E(X)$ in C_1^i, then in C_2^i, etc.

Let Z denote an arbitrary vertex along the canonical path. To apply Sinclair's method we will need that the elements of $S_{X,Y}$ can be reconstructed from elements of $S_{\Delta\cap E(Z),\Delta\setminus E(Z)}$ (using another small parameter set). In [2] the authors could prove that the elements of $S_{X,Y}$ are "almost" in $S_{\Delta\cap E(Z),\Delta\setminus E(Z)}$. Unfortunately, it is not true for our construction. This is the reason that we should introduce a much more complicated "reconstruction" method in Subsection 5.2 below.

5.1 Alternating circuit decompositions

Before we start this subsection we should recall some definitions:

Definition 5.1. In a simple graph, a sequence of pairwise disjoint edges $e_1, \ldots, e_t$ forms a **circuit** iff there are vertices $v_1, \ldots, v_t$ such that $e_i = (v_i, v_{i+1})$ (the summation is performed modulo t). This circuit is a **cycle** iff the vertices $v_1, \ldots, v_t$ are pairwise distinct.

Now let $K = (W, F \cup F')$ be a simple graph where $F \cap F' = \emptyset$ and assume that for each vertex $w \in W$ the F-degree and F'-degree of w are the same: $d(w) = d'(w)$ for all $w \in W$.

An *alternating circuit decomposition* of (F, F') is a circuit decomposition such that no two consecutive edges of any circuit are in F or in F'. Next we are going to parameterize the alternating circuit decompositions.

The set of all edges in F (in F') which are incident to a vertex w is denoted by $F(w)$ (by $F'(w)$, respectively).

If A and B are sets, denote by $[A, B]$ the complete bipartite graph with classes A and B. Let

$$\mathbb{S}(F, F') = \big\{ s : s \text{ is a function, } dom(s) = W, \text{ and for all } w \in W \\ s(w) \text{ is a 1-factor of the complete bipartite graph } [F(w), F'(w)] \big\}. \quad (5.1)$$

Lemma 5.2. *There is a natural one-to-one correspondence between the family of all alternating circuit decompositions of (F, F') and the elements of $\mathbb{S}(F, F')$.*

Proof. If $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ is an alternating circuit decomposition of (F, F'), then define $s_{\mathcal{C}} \in \mathbb{S}(F, F')$ as follows:

$$s_{\mathcal{C}}(w) := \big\{ \big((w, u), (w, u')\big) \in [F(w), F'(w)] : \\ (w, u) \text{ and } (w, u') \text{ are consecutive edges in some } C_i \in \mathcal{C} \big\}. \quad (5.2)$$

On the other hand, to each $s \in \mathbb{S}(F, F')$ assign an alternating circuit decomposition

$$\mathcal{C}_s = \{W_1^s, W_2^s \ldots, W_{k_s}^s\}$$

of (F, F') as follows: Consider the bipartite graph $\mathcal{F} = \big(F, F', R(s)\big)$, where

$$R(s) = \big\{ \big((u, w), (u', w)\big) : w \in W \text{ and } \big((u, w), (u', w)\big) \in s(w) \big\}.$$

$\mathcal{F}$ is a 2-regular graph because for each edge $(u, v) \in F \cup F'$ there is exactly one $(u, w) \in F \cup F'$ with $\big((u, w), (u, w)\big) \in s(u)$, there is exactly one $(t, v) \in F \cup F'$ with $\big((u, v), (t, v)\big) \in s(v)$, therefore the $\mathcal{F}$-neighbors of (u, v) are (u, w) and (t, v).

$\mathcal{F}$ is a 2-regular, so it is the union of vertex disjoint cycles $\{W_i^s : i \in I\}$. Now W_i^s can also be viewed as a sequence of edges in $F \cup F'$, which is an alternating circuit in $\langle W, F \cup F' \rangle$, so $\{W_i^s : i \in I\}$ is an alternating circuit decomposition of (F, F'). Since

$$s_{\mathcal{C}_s} = s,$$

we proved the Lemma. □

If the F-degree sequence (and therefore the F'-degree sequence) is $d_1, \ldots d_k$, then write

$$t_{F,F'} = \prod_{i=1}^{k} (d_i!). \quad (5.3)$$

Clearly

$$\big|\mathbb{S}\big(F, F'\big)\big| = t_{F,F'}.$$

5.2 Cycle decompositions and circuit reconstructions

In this subsection we make preparations for constructing our multicommodity flow: we describe how we decompose an alternating circuit into alternating cycles.

The problem of this venture is the following: we know along the process the symmetric difference of the edge sets of realizations X and Y but we do not know the distribution of the edges among $E(X)$ and $E(Y)$. If the alternating circuit under investigation is large then its cycle decomposition can contain a linear number of alternating cycles. Each cycle consists of an even number of edges, equally distributed between X and Y. Along the process each cycle needs a parameter representing whether that particular cycle was already processed or not (which, in turn, tells which edges belong to X and Y). Therefore along all decompositions the set of all possible parameter values can be exponentially big, which is not suitable to prove fast mixing property. Therefore we need to find another way to deal with the reconstruction problem. We can proceed as follows:

Let $\mathbf{x} = (x_1, x_2, \dots, x_m)$ be a sequence, then we write $\overleftarrow{\mathbf{x}} = (x_m, \dots, x_2, x_1)$ for the oppositely ordered sequence. (Here we consider $\leftarrow$ to be an operator.)

Assume that $K = (W, F \cup F')$ is a simple bipartite graph where $F \cap F' = \emptyset$ (in our applications we have $|F| = |F'|$), and the sequence

$$\mathbf{e} = (e_1, e_2, \dots, e_m)$$

of edges is an (F, F')-*alternating trail* in K, (i.e. no two consecutive edges from $\mathbf{e}$ are in F or in F', moreover the edges in $\mathbf{e}$ are pairwise different). In this subsection we also use extensively the notation $\mathbf{e} = e_1 e_2 \cdots e_m$ for the same sequence. When $\mathbf{e}'$ and $\mathbf{e}''$ are two sequences, then $\mathbf{e}'\mathbf{e}''$ stands for their concatenation. We will write $e_i = v_i v_{i+1}$. We will also use $\mathrm{b}(e_i) = v_i$ (for the *bottom* of the edge) and $\mathrm{t}(e_i) = v_{i+1}$ (for the *top* of the edge, considering the actual orientations along the trail). So $\mathrm{b}(e_{i+1}) = \mathrm{t}(e_i)$, and $\mathrm{t}(e_m) = \mathrm{b}(e_1)$ iff $\mathbf{e}$ is a circuit.

We will use the notations $\mathbf{e}(i) = e_i$ and $v_{\mathbf{e}}(j) = v_j$ for $1 \leqslant i \leqslant m$ and $1 \leqslant j \leqslant m+1$ (the ith edge and the jth vertex of the trail). If $\mathbf{c} = e_i \cdots e_j$ is a consecutive subsequence of $\mathbf{e}$, we will also write $\mathrm{b}(\mathbf{c}) = \mathrm{b}(e_i)$ and $\mathrm{t}(\mathbf{c}) = \mathrm{t}(e_j)$. Finally $\mathrm{first}(\mathbf{e}')$ denotes the first edge, while $\mathrm{last}(\mathbf{e}')$ denotes the last edge of trail $\mathbf{e}'$.

Now let f be a coloration of the edges along the trail $f : \mathbf{e} \to \{\mathbf{green}, \mathbf{red}\}$. One can imagine it as an indicator whether the edges were processed already along the transformation of the realization X into realization Y. (Green edges are ready for processing while red edges are processed already.)

For $1 \leqslant i < j \leqslant m$ denote $\mathbf{green}_f[e_i, e_j]$ the (not necessarily consecutive) subsequence of $\mathbf{green}_f$ edges from the sequence $e_i \cdots e_j$. (The notation $\mathbf{green}_f$ is a shorthand for $\mathbf{green}_f[e_1, e_m]$, and the notations $\mathbf{red}_f[e_i, e_j]$ are defined analogously.) We will maintain the following property along our algorithm:

($\pounds$) any maximal consecutive **red** subsequence in $\mathbf{e}$ forms a closed alternating trail.

Let f be a coloration on the current alternating trail $\mathbf{e}$ satisfying property ($\pounds$). Furthermore let

$$j = \min\{j : \exists i < j \;\; \mathbf{green}_f[e_i, e_j] \text{ is a cycle}\}, \tag{5.4}$$

and let

$$i = \max\{i : \ \mathbf{green}_f[e_i, e_j] \text{ is a cycle}\}. \tag{5.5}$$

Since $\mathbf{e}$ is not necessarily a closed trail therefore such j does not always exist. However if $\mathbf{e}$ is a closed trail and $\mathbf{green}_f$ is not empty, then such j exists. Indeed, if a closed trail is deleted from a bigger closed trail then the remnant is a closed trail. Furthermore it is clear that integer j determines uniquely the green cycle ending at e_j. However before this cycle (along the original circuit) there may be several red edges. Therefore there may be several different integer i defining the same green cycle. One way to handle this fact is equation (5.5).

Now we are ready to introduce our main tool to control the decomposition of an alternating circuit into alternating cycles.

For that end we define the operator $\mathbb{T}$ on the edges of trail $\mathbf{e}$ and the current coloration f (satisfying condition $(\pounds)$) as follows:

Definition 5.3. $\mathbb{T}(\mathbf{e}, f)$ will be a triple $(\mathbf{e}', f', \mathcal{C}')$, where

(i) $\mathcal{C}'$ is the alternating cycle in $\mathbf{e}$ defined by equalities (5.4) and (5.5), so

$$\mathcal{C}' = \mathbf{green}_f[e_i, e_j];$$

(ii) $\mathbf{e}'$ is an alternating trail obtained by rearranging the edges from $\mathbf{e}$ as explained below;

(iii) $f' : \mathbf{e}' \to \{\mathbf{green}, \mathbf{red}\}$ is defined with

$$\mathbf{red}_{f'} = \mathbf{red}_f \cup \mathcal{C}'.$$

If j is undefined, then $\mathbb{T}(\mathbf{e}, f)$ is undefined. Let us remark that the length of $\mathcal{C}'$ is even, because $(K, F \cup F')$ was a bipartite graph, so $\mathcal{C}'$ is an alternating cycle.

We introduce the following notation:

$$\mathbf{T}(\mathbf{e}, f) = \mathbf{e}', \ \mathbf{f}(\mathbf{e}, f) = f', \text{ and } \mathcal{C}(\mathbf{e}, f) = \mathcal{C}'.$$

What is missing is the description of the new alternating trail $\mathbf{e}'$. Next we do just that. (Let's recall that two sequences written next to each other denotes their concatenation.) Write

$$[e_i, e_j] = \mathbf{g}_1\mathbf{r}_1 \cdots \mathbf{r}_{k-1}\mathbf{g}_k,$$

where

$$\mathbf{green}_f[e_i, e_j] = \mathbf{g}_1\mathbf{g}_2 \cdots \mathbf{g}_{k-1}\mathbf{g}_k$$

and

$$\mathbf{red}_f[e_i, e_j] = \mathbf{r}_1\mathbf{r}_2 \cdots \mathbf{r}_{k-1}.$$

In words: the $\mathbf{g}_i$ and $\mathbf{r}_i$ represent the maximal consecutive $\mathbf{green}_f$ and $\mathbf{red}_f$ subsequences. Let

$$i' = \begin{cases} \min\limits_{\ell}\{\ell < i : [e_\ell, e_{i-1}] \text{ is } \mathbf{red}_f\}, & \text{if } f(e_{i-1}) = \mathbf{red}_f, \\ i, & \text{otherwise.} \end{cases}$$

Furthermore let

$$j' = \begin{cases} \max\limits_{\ell}\{\ell > j : \ [e_{j+1}, e_\ell] \text{ is } \mathbf{red}_f\}, & \text{if } f(e_{j+1}) = \mathbf{red}_f, \\ j, & \text{otherwise.} \end{cases}$$

We define

$$\mathbf{r}^- = \begin{cases} [e_{i'}, e_{i-1}], & \text{if } i' < i, \\ \emptyset, & \text{if } i' = i; \end{cases}$$

and

$$\mathbf{r}^+ = \begin{cases} [e_{j+1}, e_{j'}], & \text{if } j' > j, \\ \emptyset, & \text{if } j' = j. \end{cases}$$

Let

$$\mathbf{e}' = e_1 \cdots e_{i'-1}\mathbf{r}^+\overleftarrow{\mathbf{g}_k}\mathbf{r}_{k-1}\overleftarrow{\mathbf{g}_{k-1}}\cdots\mathbf{r}_1\overleftarrow{\mathbf{g}_1}\mathbf{r}^- e_{j'+1}\cdots e_m. \tag{5.6}$$

This last formula requires some explanation: the cycle $\mathcal{C}'$ consists of the $\mathbf{green}_f$ segments of $[e_i, e_j]$. All the $\mathbf{red}_f$ segments form alternating closed trails that were processed earlier. We may assume without loss of generality, that the very first edge e_i belongs to F, consequently the last edge e_j belongs to F'.

When we finish the required swap operations exchanging the edges from F into edges from F' along cycle $\mathcal{C}'$ (and transferring the actual degree realization closer to realization Y), then listing the edges of $\mathcal{C}'$ in the same way as before would not produce an alternating closed trail anymore. To form an alternating trail again we must consider the edges of $\mathcal{C}'$ in the opposite order. This is done by the subsegments $\overleftarrow{\mathbf{g}_i}$s. Listing $\mathcal{C}'$ in opposite order must list the closed trails $\mathbf{r}_i$s also in opposite order (see (5.6)), which in turns takes care automatically for keeping the alternating order of edges from F and F'.

One can ask the reason to exchange $\mathbf{r}^-$ and $\mathbf{r}^+$ since this is not necessary to keep the trail alternating. This reason lies in equation (5.10).

By induction on $i \in \mathbb{N}$, we can define $\mathbf{T}^i(\mathbf{e}, f)$ and $\mathbf{f}^i(\mathbf{e}, f)$ as follows: $\mathbf{T}^0(\mathbf{e}, f) := \mathbf{e}$, $\mathbf{f}^0(\mathbf{e}, f) := f$, and

$$\mathbf{T}^i(\mathbf{e}, f) := \mathbf{T}\big(\mathbf{T}^{i-1}(\mathbf{e}, f), \mathbf{f}^{i-1}(\mathbf{e}, f)\big) \text{ and } \mathbf{f}^i(\mathbf{e}, f) := \mathbf{f}\big(\mathbf{T}^{i-1}(\mathbf{e}, f), \mathbf{f}^{i-1}(\mathbf{e}, f)\big)$$

for $i > 0$. Let us remark that $\mathbf{T}^i(\mathbf{e}, f)$ and $\mathbf{f}^i(\mathbf{e}, f)$ are not necessarily defined.

Now we are ready to describe the control mechanism to govern the swap sequence to change the edges of the current realization belonging to F into the edges belonging to F' along the alternating closed trail $\mathbf{e}$. For that end denote $\mathfrak{g}$ the constant **green** function on $\mathbf{e}$, i.e. $\mathbf{green}_{\mathfrak{g}} = [e_1, e_m]$, furthermore let $\mathbf{e}_0 := \mathbf{e}$ and $f_0 := \mathfrak{g}$. Now we define the sequence

$$(\mathbf{e}_1, f_1, \mathcal{C}_1), (\mathbf{e}_2, f_2, \mathcal{C}_2), \ldots, (\mathbf{e}_n, f_n, \mathcal{C}_n)$$

by the formula

$$(\mathbf{e}_{\ell+1}, f_{\ell+1}, \mathcal{C}_{\ell+1}) := \mathbb{T}(\mathbf{e}_\ell, f_\ell)$$

for $\ell = 0, 1, \ldots$. We stop when $\mathbb{T}(\mathbf{e}_n, f_n)$ is undefined. We define $n(\mathbf{e}) := n$ and observe that

$$\mathbf{T}^i(\mathbf{e}, \mathfrak{g}) = \mathbf{e}_i \text{ and } \mathbf{f}^i(\mathbf{e}, \mathfrak{g}) = f_i \quad \text{for } 0 \leqslant i \leqslant n(\mathbf{e}). \tag{5.7}$$

We also define the sequence

$$(F_0, F_0'), (F_1, F_1'), \ldots, (F_n, F_n')$$

of partitions of $F \cup F'$ as follows:

(1) let $F_0 := F$ and $F_0' := F'$,

(2) let $F_{i+1} := F_i \cup (\mathcal{C}_{i+1} \setminus F_i) \setminus (\mathcal{C}_{i+1} \cap F_i)$ and $F_{i+1}' := (F \cup F') \setminus F_{i+1}$.

It is easy to see, and we will show formally in Lemma 5.5, that if $\mathbf{e}$ is a circuit then $\mathcal{C}_1, \ldots, \mathcal{C}_{n(\mathbf{e})}$ will be a circuit decomposition of $\mathbf{e}$. Later we will use this decomposition to obtain our canonical path system.

We will prove a series of observations. We start with some easy direct consequences of definitions (5.4), (5.5) and (5.6). In Lemmas 5.4, 5.5 and 5.6 below we will use the notation $\mathbf{e}_i = \mathbf{T}^i(\mathbf{e}, \mathfrak{g})$ and $f_i = \mathbf{f}^i(\mathbf{e}, \mathfrak{g})$ for $0 \leqslant i \leqslant n(\mathbf{e})$.

Lemma 5.4. *During the algorithm, at any given iteration κ we have:*

(i) *in the current alternating trail $\mathbf{e}_{\kappa-1}$ the edge e_{j_κ} is after all $\mathbf{red}_{f_{\kappa-1}}$ edges, where j_κ denotes the value j used in the κth iteration of the construction;*

(ii) *for any red edge the size of the maximal red subsequence containing it cannot decrease;*

(iii) *the number of maximal red subsequences can be increased by at most one, but can drop to 1.* □

Lemma 5.5. *For each $0 \leqslant \nu \leqslant n(\mathbf{e})$ we have:*

(i) *maximal $\mathbf{red}_{f_\nu}$ intervals $[\mathbf{e}_\nu(k), \mathbf{e}_\nu(\ell)]$ in $\mathbf{e}_\nu$ are circuits (recall, $\mathbf{e}_\nu(d)$ is the dth edge along $\mathbf{e}_\nu$);*

(ii) *the edge sequence $\mathbf{e}_\nu$ is a trail which alternates between F_ν and F_ν';*

(iii) *$v_{\mathbf{e}_\nu}(1) = v_{\mathbf{e}}(1)$ and $v_{\mathbf{e}_\nu}(m+1) = v_{\mathbf{e}}(m+1)$ (these are the very first and very last vertices in $\mathbf{e}$);*

(iv) *$\mathbf{green}_{f_\nu}[\mathbf{e}_\nu]$ is a trail from $v_{\mathbf{e}}(1)$ to $v_{\mathbf{e}}(m+1)$ (while only just a part of the edges of $\mathbf{e}_\nu$ are **green** they still provide an alternating trail between those vertices).*

(v) *if $\mathbf{e}$ is circuit, then $f_n(\mathbf{e})$ is the constant red function, i.e. we processed all edges, while $\mathcal{C}_1, \ldots, \mathcal{C}_n$ is an alternating cycle decomposition of $\mathbf{e}$.*

Proof. We prove the statements by induction on ν. For $\nu = 0$ the statements are trivial because $\mathbf{green}_{f_0} = [e_1, e_m]$. Consider now the inductive step $\nu - 1 \to \nu$. Assume that

$$\mathbf{e}_{\nu-1} = e_1 \cdots e_{i'-1}\mathbf{r}^-\mathbf{g}_1\mathbf{r}_1 \cdots \mathbf{g}_{k-1}\mathbf{r}_{k-1}\mathbf{g}_k\mathbf{r}^+e_{j'} \cdots e_m \tag{5.8}$$

and

$$\mathbf{e}_\nu = e_1 \cdots e_{i'-1}\mathbf{r}^+\overleftarrow{\mathbf{g}_k}\mathbf{r}_{k-1}\overleftarrow{\mathbf{g}_{k-1}} \cdots \mathbf{r}_1\overleftarrow{\mathbf{g}_1}\mathbf{r}^-e_{j'} \cdots e_m.$$

(Here it is important to recall, that when $\mathbf{r}^-$ and/or $\mathbf{r}^+$ is empty, then $i' = i - 1$ and/or $j' = j + 1$. If some of these cases apply, then the corresponding remarks on $\mathbf{r}^-$ and $\mathbf{r}^+$ are void.)

(i) The intervals $\mathbf{r}_\ell$ are maximal red intervals, so by the inductive assumption they are circuits, i.e. $\mathrm{b}(\mathbf{r}_\ell) = \mathrm{t}(\mathbf{r}_\ell)$. Moreover, by the construction, the first vertex of $\mathbf{g}_1$ and the last vertex of $\mathbf{g}_k$ are the same: $\mathrm{b}(\mathbf{g}_1) = \mathrm{t}(\mathbf{g}_k)$, and $\mathbf{g}_\ell$ is a path from $\mathrm{b}(\mathbf{g}_\ell)$ to $\mathrm{t}(\mathbf{g}_\ell)$. Since $\mathrm{t}(\mathbf{g}_\ell) = \mathrm{b}(\mathbf{r}_\ell) = \mathrm{t}(\mathbf{r}_\ell) = \mathrm{b}(\mathbf{g}_{\ell+1})$, we have that

$$\mathbf{c} \;=\; \overleftarrow{\mathbf{g}_k}\mathbf{r}_{k-1}\overleftarrow{\mathbf{g}_{k-1}} \cdots \mathbf{r}_1\overleftarrow{\mathbf{g}_1}$$

is a $\mathbf{red}_{f_\nu}$ circuit.

To finish the proof of (i) there is only one remaining case: if the maximal $\mathbf{red}_{f_\nu}$ interval $[e_l, e_\ell]$ in $\mathbf{e}_\nu$ properly contains the interval $[e_i, e_j]$. (This is the case when at least one of $\mathbf{r}^-$ and $\mathbf{r}^+$ are not empty.) Then both $[e_l, e_{i-1}]$ and $[e_{j+1}, e_\ell]$ are maximal $\mathbf{red}_{f_{\nu-1}}$ intervals in $\mathbf{e}_{\nu-1}$ so $[e_l, e_{i-1}] = \mathbf{r}^-$ and $[e_{j+1}, e_\ell] = \mathbf{r}^+$, therefore $[e_l, e_{i-1}] = \mathbf{r}^-[e_i, e_j]\mathbf{r}^+$ is the concatenation of at most three circuits (since $\mathbf{r}^-$ or $\mathbf{r}^+$, but not both, can be empty), so it is also a circuit.

(ii) The vertices $v_{\mathbf{r}^-}(1) = v_{\mathbf{r}^+}(1) = v_{\mathbf{g}_1}(1)$ are identical in $\mathbf{e}_{\nu-1}$. Therefore $e_{i'} \in F_{\nu-1}$ if and only if $e_{j+1} \in F_{\nu-1}$, and the same applies for the edges $\mathrm{last}_{\nu-1}(\mathbf{r}^-) = e_{i-1}$ and $\mathrm{last}_{\nu-1}(\mathbf{r}^+) = e_{j'}$. (Here the index in $\mathrm{last}_{\nu-1}()$ refers to the order of the trail $\mathbf{e}_{\nu-1}$.) So, since $e_1 \cdots e_{i'-1}\mathbf{r}^-$ an alternating trail in $\mathbf{e}_{\nu-1}$ therefore the same applies for $e_1 \cdots e_{i'-1}\mathbf{r}^+$ (and analogously for $\mathbf{r}^-e_{j'+1} \cdots e_m$) in $\mathbf{e}_\nu$. In other words it makes no difference in the behavior (relating to the sub-trail $[e_i, e_j]$) of the trails $[e_1, e_{i-1}]$ and $[e_{j+1}, e_m]$ whether $\mathbf{r}^-$ and/or $\mathbf{r}^+$ is/are empty.

Furthermore we have

$$\mathrm{t}(e_1 \cdots e_{i'-1}\mathbf{r}^+) = \mathrm{b}(\mathbf{g}_1) = \mathrm{b}(\mathbf{c}) = \mathrm{t}(\mathbf{g}_k) = \mathrm{t}(\mathbf{c}) = \mathrm{b}(\mathbf{r}^-e_{j'+1} \cdots e_m),$$

so $\mathbf{e}_\nu$ is a trail.

Next we check whether $\mathbf{e}_\nu$ alternates between F_ν and F'_ν.

Since we have $F_\nu \cap \{e_0 \cdots e_{i'-1} \cup \mathbf{r}^+\} = F_{\nu-1} \cap \{e_0 \cdots e_{i'-1} \cup \mathbf{r}^+\}$, the interval $e_0 \cdots e_{i'-1}\mathbf{r}^+$ alternates between F_ν and F'_ν and the analogous statements holds for $\mathbf{r}^-e_{j'+1} \cdots e_m$.

We know that

$$e_{i-1} \in F_{\nu-1} \Leftrightarrow e_i \in F'_{\nu-1} \Leftrightarrow e_j \in F_{\nu-1},$$

since $[e_i, e_j]$ is a circuit in $\mathbf{e}_{\nu-1}$. We also have that

$$e_{j'} \in F_\nu \Leftrightarrow e_i \in F_\nu \Leftrightarrow e_j \in F'_\nu.$$

Since e_j is the first edge of $\overleftarrow{\mathbf{g}_k}$, the path $e_0 \cdots e_{i'-1}\mathbf{r}^+\overleftarrow{\mathbf{g}_k}$ alternates between F_ν and F'_ν.

Assume that $\mathbf{r}_{k-1} = e_p \cdots e_r$. Then

$$e_p \in F_{\nu-1} \Leftrightarrow e_r \in F'_{\nu-1} \Leftrightarrow e_{r+1} \in F_{\nu-1}.$$

Thus

$$e_p \in F_\nu \Leftrightarrow e_r \in F'_\nu \Leftrightarrow e_{r+1} \in F'_\nu.$$

Therefore the path $e_1 \cdots e_{i'-1}\mathbf{r}^+\overleftarrow{\mathbf{g}_k}\mathbf{r}_{k-1}$ alternates between F_ν and F'_ν because $\mathrm{last}_\nu(\overleftarrow{\mathbf{g}_k})$ is e_{r+1} and $\mathrm{first}_\nu(\mathbf{r}_{k-1})$ is e_p.

Repeating the arguments above we obtain that the whole path $\mathbf{e}_\nu$ alternates between F_ν and F'_ν which finishes the proof of (ii).

(iii) Here everything is trivial - except if $i = 1$ and/or $j = m$. By symmetry, it is enough to study one of these, let say $j = m$. Then the last segment of $\mathbf{e}_\nu$ is $\mathbf{r}^+\overleftarrow{\mathbf{g}_k} \cdots \overleftarrow{\mathbf{g}_1}\mathbf{r}^-$ which is a circuit, so the current end point of $\mathbf{e}_\nu$ is the same as the original end point of $\mathbf{e}_{\nu-1}$.

(iv) All maximal $\mathbf{red}_{f_\nu}$ intervals are circuits, therefore removing them one by one from $\mathbf{e}_\nu$ does not destroy the connectivity in $\mathbf{green}_{f_\nu}$ from $b(\mathbf{e}_\nu)$ to $t(\mathbf{e}_\nu)$ (as far as there are green edges).

(v) It follows immediately from (iii) and (iv): a non-empty green remainder is a circuit, so the process will not finish while there still exists some green remainder. Consequently $\nu < n(\mathbf{e})$. □

Lemma 5.6. (a) *For each* $0 \leqslant \nu \leqslant n$ *and* $1 \leqslant r < s \leqslant m$, *if* $\mathbf{e}_\nu(r)$ *is* $\mathbf{green}_{f_\nu}$ *and* $\mathbf{e}_\nu(s)$ *is* $\mathbf{red}_{f_\nu}$, *then* $\mathrm{b}(\mathbf{e}_\nu(r)) \notin \mathbf{e}_\nu(s)$.
(b) *Furthermore if* $\mathbf{e}_\nu(r')$ *is also* $\mathbf{green}_{f_\nu}$ *where* $r < r' < s$, *then* $\mathrm{b}(\mathbf{e}_\nu(r)) \neq \mathrm{t}(\mathbf{e}_\nu(r'))$.

Proof. To prove (a) assume on the contrary that for some $1 \leqslant r < s \leqslant m$ we have $\mathrm{b}(\mathbf{e}_\nu(r)) \in \mathbf{e}_\nu(s)$, $\mathbf{e}_\nu(r)$ is $\mathbf{green}_{f_\nu}$ and $\mathbf{e}_\nu(s)$ is $\mathbf{red}_{f_\nu}$.

Consider a counterexample where ν is minimal. Assume that

$$\mathbf{e}_{\nu-1} = e_1 \cdots e_{i'-1}\mathbf{r}^-\mathbf{g}_1\mathbf{r}_1 \cdots \mathbf{g}_{k-1}\mathbf{r}_{k-1}\mathbf{g}_k\mathbf{r}^+e_{j'+1} \cdots e_m$$

and

$$\mathbf{e}_\nu = e_1 \cdots e_{i'-1}\mathbf{r}^+\overleftarrow{\mathbf{g}_k}\mathbf{r}_{k-1}\overleftarrow{\mathbf{g}_{k-1}} \cdots \mathbf{r}_1\overleftarrow{\mathbf{g}_1}\mathbf{r}^-e_{j'+1} \cdots e_m.$$

The edge sequence $\mathbf{g}_1\mathbf{r}_1 \cdots \mathbf{g}_{k-1}\mathbf{r}_{k-1}\mathbf{g}_k$ (in $\mathbf{e}_{\nu-1}$) is a circuit.

Since $\mathbf{e}_\nu(r)$ is unprocessed in $\mathbf{e}_\nu$ therefore $\mathbf{e}_\nu(r) \in e_0 \dots e_{i'-1} \cup e_{j'+1} \cdots e_m$. Furthermore $\mathbf{e}_\nu(s) \in \overleftarrow{\mathbf{g}_k} \cup \overleftarrow{\mathbf{g}_{k-1}} \cup \cdots \cup \overleftarrow{\mathbf{g}_1}$ otherwise its color would be the same under $f_{\nu-1}$ and f_ν therefore ν would not be a minimal counterexample. But then the property $r < s$ infers that $\mathbf{e}_\nu(r) \in e_0 \dots e_{i'-1}$.

Moreover $b(\mathbf{e}_{\nu-1}(r)) = b(\mathbf{e}_\nu(r)) \neq b(e_i) = t(e_j)$. Indeed, if $\mathbf{r}^-$ is not empty, then $\mathbf{e}_{\nu-1}(r)$ and e_{i-1} would form a counterexample to Lemma 5.6(a) in $\mathbf{e}_{\nu-1}$, which contradicts the minimality of ν (the other case is similar). If both $\mathbf{r}^-$ and $\mathbf{r}^+$ are empty, then $[\mathbf{e}_{\nu-1}(r), \mathbf{e}_{\nu-1}(i-1)]$ would be a circuit and it would contain a $\mathbf{green}_{f_\nu}$ cycle, a contradiction to the definition of $\mathcal{C}_\nu$ (in $\mathbf{e}_{\nu-1}$).

Therefore $b(\mathbf{e}_\nu(r))$ must be an inner vertex of the cycle $e_i \dots e_j$. Now if this vertex is not the last vertex of a $\overleftarrow{\mathbf{g}_\ell}$, that is we have $b(e_\nu(r)) = b(e_\nu(s))$ then $\mathbf{green}_{f_{\nu-1}}[\mathbf{e}_{\nu-1}(r), \mathbf{e}_{\nu-1}(s)]$ would be a circuit, containing a $\mathbf{green}_{f_{\nu-1}}$ cycle with smaller maximal element than e_j, which contradicts to the definition of $\mathcal{C}_\nu$ in $\mathbf{e}_{\nu-1}$. Finally, if this vertex is the last vertex of a $\overleftarrow{\mathbf{g}_\ell}$ then it is also the first vertex of $\mathbf{r}_{\ell-1}$ therefore edges $(e_\nu(r))$ and $\mathrm{first}(\mathbf{r}_{\ell-1})$ would form already in $\mathbf{e}_{\nu-1}$ the forbidden configuration of the statement, contradicting the minimality of ν.

The proof of (b) uses a similar argument. □

Lemma 5.7. *Assume that for some $r \geqslant 0$,*

$$\mathbf{T}^r(\mathbf{e}, \mathfrak{g}) = \mathbf{g}_1\mathbf{r}_1\mathbf{g}_2\mathbf{r}_2 \dots \mathbf{r}_k\mathbf{g}_{k+1}, \tag{5.9}$$

where the first and/or the last green subsequence can be empty. Then

$$\mathbf{e} = \mathbf{g}_1\overleftarrow{\mathbf{r}_1}\mathbf{g}_2\overleftarrow{\mathbf{r}_2} \dots \overleftarrow{\mathbf{r}_k}\mathbf{g}_{k+1}, \tag{5.10}$$

so we obtain back the original edge sequence $\mathbf{e}$.

It is important to understand that here we do not have any realization in the background (and no alternation is considered on the edges), we consider only the order of the edges. The operations above are nothing else, just turning back all maximal $\mathbf{red}_{f^r(\mathbf{e})}$ intervals in $\mathbf{T}^r(\mathbf{e}, \mathfrak{g})$.

Proof. We apply mathematical induction on r. For $r = 0$ the statement is trivial because $\mathbf{T}^0(\mathbf{e}, \mathfrak{g}) = \mathbf{e} = \mathbf{g}_1$.

Now we assume that the statement is true for $(r-1)$ and we are going to prove it for r. For that end assume that

$$\mathbf{T}^{r-1}(\mathbf{e}, \mathfrak{g}) = \mathbf{g}_1\mathbf{r}_1 \cdots \underbrace{\mathbf{r}^-\mathbf{g}_t\mathbf{r}_t \cdots \mathbf{r}_{u-1}\mathbf{g}_u\mathbf{r}^+}\,\mathbf{g}_{u+1} \cdots\cdots \mathbf{r}_k\mathbf{g}_{k+1}. \tag{5.11}$$

where the formulas 5.4 and 5.5 select the intervals $\mathbf{r}^-\mathbf{g}_t\mathbf{r}_t \cdots \mathbf{r}_{u-1}\mathbf{g}_u\mathbf{r}^+$ to process (where $\mathbf{r}^-$ and/or $\mathbf{r}^+$ can be empty).

To compute $\mathbf{T}^r(\mathbf{e}, \mathfrak{g})$ we should check if $\mathbf{r}^-$ and $\mathbf{r}^+$ are empty or not. Altogether there are four cases to investigate, however the properties of one end of the sequence of $\mathcal{C}_r$ does not influence the other end, therefore it is enough to consider one "generic case", say, when $\mathbf{r}^-$ is empty but $\mathbf{r}^+$ is not empty. Then

$$\mathbf{T}^r(\mathbf{e}, \mathfrak{g}) = \mathbf{g}_1\mathbf{r}_1 \cdots \underbrace{\mathbf{r}^+\overleftarrow{\mathbf{g}_u}\mathbf{r}_{u-1}\overleftarrow{\mathbf{g}_{u-1}} \cdots \mathbf{r}_t\overleftarrow{\mathbf{g}_t}}_{\text{red in } \mathbf{f}^r(\mathbf{e}, \mathfrak{g})}\,\mathbf{g}_{u+1} \cdots\cdots \mathbf{r}_k\mathbf{g}_{k+1}. \tag{5.12}$$

Now $\mathbf{r}^+\overleftarrow{\mathbf{g}_u}\mathbf{r}_{u-1}\overleftarrow{\mathbf{g}_{u-1}} \cdots \mathbf{r}_t\overleftarrow{\mathbf{g}_t}$ is a maximal red interval in $\mathbf{f}^r(\mathbf{e}, \mathfrak{g})$. When we "turn back" the $\mathbf{f}^r(\mathbf{e}, \mathfrak{g})$-red maximal intervals in $\mathbf{T}^r(\mathbf{e}, \mathfrak{g})$ we get:

$$\begin{aligned}
\mathbf{g}_1\overleftarrow{\mathbf{r}_1} \cdots \mathbf{g}_{t-1}\Big(\overleftarrow{\mathbf{r}^+\overleftarrow{\mathbf{g}_u}\mathbf{r}_{u-1} \cdots \mathbf{r}_t\overleftarrow{\mathbf{g}_t}}\Big)\mathbf{g}_{u+1}\overleftarrow{\mathbf{r}_{u+1}} \cdots\cdots \overleftarrow{\mathbf{r}_k}\mathbf{g}_{k+1} &= \\
\mathbf{g}_1\overleftarrow{\mathbf{r}_1} \cdots \mathbf{g}_{t-1}\Big(\mathbf{g}_t\overleftarrow{\mathbf{r}_t} \cdots \mathbf{g}_{u-1}\overleftarrow{\mathbf{r}_{u-1}}\mathbf{g}_u\overleftarrow{\mathbf{r}^+}\Big)\mathbf{g}_{u+1} \cdots\cdots \overleftarrow{\mathbf{r}_k}\mathbf{g}_{k+1} &= \\
\mathbf{g}_1\overleftarrow{\mathbf{r}_1} \cdots \mathbf{g}_{t-1}\mathbf{g}_t\overleftarrow{\mathbf{r}_t} \cdots \mathbf{g}_{u-1}\overleftarrow{\mathbf{r}_{u-1}}\mathbf{g}_u\overleftarrow{\mathbf{r}^+}\mathbf{g}_{u+1} \cdots\cdots \overleftarrow{\mathbf{r}_k}\mathbf{g}_{k+1} &= \mathbf{e}
\end{aligned} \tag{5.13}$$

where (5.13) is just the inductive assumption. □

Lemma 5.8. *Let* $\mathbf{e}' = \mathbf{T}^r(\mathbf{e}, \mathfrak{g})$ *for some* $r \geqslant 0$*, and assume that* (5.9) *holds, and define* $n_\ell := n(\mathbf{r}_\ell)$ *for* $1 \leqslant \ell \leqslant k$*. Furthermore let* $t_0 = 0$ *and* $t_\ell := n_1 + \cdots + n_\ell$ *for all* $1 \leqslant \ell \leqslant k$*. Then*

$$\mathbf{T}^{t_\ell}(\mathbf{e}', \mathfrak{g}) = \mathbf{g}_1 \overleftarrow{\mathbf{r}_1} \dots \mathbf{g}_\ell \overleftarrow{\mathbf{r}_\ell} \mathbf{g}_{\ell+1} \mathbf{r}_{\ell+1} \cdots \mathbf{r}_k \mathbf{g}_{k+1}.$$

Remark 5.9. It is important to emphases that there is no reason that the algorithm running on $\mathbf{e}'$ would provide the same cycle decompositions of circuits $\mathbf{r}_i$ as the the same algorithm, running on the original $\mathbf{e}$ would do. As a matter of fact one can construct example where this is not the case.

Proof of the Lemma 5.8. We apply induction on ℓ. For $\ell = 0$ there is no processed edge in $\mathbf{e}'$, moreover $t_0 = 0$, so nothing to prove. So assume that $\ell \geqslant 1$ and we know the statement for $\ell - 1$. For $v = 0, \dots, n_\ell$ let $\tau_v := t_{\ell-1} + v$. We are going to show that

$$\mathbf{T}^{\tau_v}(\mathbf{e}', \mathfrak{g}) = \mathbf{g}_1 \overleftarrow{\mathbf{r}_1} \cdots \overleftarrow{\mathbf{r}_{\ell-1}} \mathbf{g}_\ell \mathbf{T}^v(\mathbf{r}_\ell, \mathfrak{g}) \mathbf{g}_{\ell+1} \mathbf{r}_{\ell+1} \cdots \mathbf{g}_{k+1}. \tag{5.14}$$

In words: iterations $t_{\ell-1}+1, \dots, t_\ell$ of our algorithm work on $\mathbf{r}_\ell$ and completely process it, furthermore at each iteration we have

$$\mathbf{f}^{\tau_v}(\mathbf{e}', \mathfrak{g})\big|_{\mathbf{r}_\ell} = \mathbf{f}^v(\mathbf{r}_\ell, \mathfrak{g}). \tag{5.15}$$

We prove it with induction on v. When $v = 0$ then we have nothing to prove, since case τ_0 coincides with $t_{\ell-1}$. Assume now that (5.14) and (5.15) hold for τ_{v-1} and prove it for $\tau_v = \tau_{v-1} + 1$.

Now we compute

$$\mathbf{T}^{\tau_v}(\mathbf{e}', \mathfrak{g}) = \mathbf{T}\big(\mathbf{T}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g}), \mathbf{f}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g})\big).$$

Let j_{τ_v} and i_{τ_v} be the natural numbers j and i given by formulas (5.4) and (5.5) for $\mathbf{T}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g})$ and $\mathbf{f}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g})$).

By Lemma 5.4 (i) the current $e_{j_{\tau_v}}$, the j_{τ_v}th element of $\mathbf{T}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g})$, is after all $\mathbf{red}_{\mathbf{f}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g}))}$ edges. However it is within $\mathbf{r}_\ell$ since the original execution of our algorithm producing $\mathbf{e}'$ fully processed the closed trail $\mathbf{r}_\ell$ while in $\mathbf{T}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g})$ it is not achieved yet: there exists at least one unprocessed cycle. Finally, for the same reason, $e_{i_{\tau_v}}$, the i_{τ_v}th element of $\mathbf{T}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g})$, also should be in $[\mathbf{r}_\ell]$. So we know that $[e_{i_{\tau_v}}, e_{j_{\tau_v}}]$ (computed in $\mathbf{T}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g})$) is inside $\mathbf{r}_\ell$.

Thus, by the inductive hypothesis (5.15), the operation $\mathbb{T}$ described in Definition 5.3 produces the same circles for $\mathbf{T}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g})$ and $\mathbf{f}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g})$, and for $\mathbf{T}^{v-1}(\mathbf{r}_\ell, \mathfrak{g})$ and $\mathbf{f}^{v-1}(\mathbf{r}_\ell, \mathfrak{g})$, i.e.,

$$\mathcal{C}\big(\mathbf{T}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g}), \mathbf{f}^{\tau_{v-1}}(\mathbf{e}', \mathfrak{g})\big) = \mathcal{C}\big(\mathbf{T}^{v-1}(\mathbf{r}_\ell, \mathfrak{g}), \mathbf{f}^{v-1}(\mathbf{r}_\ell, \mathfrak{g})\big).$$

which, in turn, proves (5.15) and (5.14) for τ_v. □

Now we are ready to formalize the center piece of our control mechanism to govern the construction of the required multicommodity flow (or, in other words, the swap sequences

between different realizations). With the previous definitions one can quantify the size of a parameter set to follow the current status of the cycles in the decomposition of the alternating circuit $\mathbf{e}$. It clearly can be exponentially big, so this cannot prove fast mixing time.

However, we do not need to know the status of those cycles. What we really have to know is the original trail $\mathbf{e}$. And, surprisingly enough, we can determine it with high probability. More precisely the following property holds:

Theorem 5.10. *If* $\mathbf{e}$ *is a circuit, and* $0 \leqslant s \leqslant n(\mathbf{e})$*, then*

$$\mathbf{T}^s(\mathbf{T}^r(\mathbf{e}, \mathfrak{g}), \mathfrak{g}) = \mathbf{e} \tag{5.16}$$

for some $0 \leqslant s \leqslant n(\mathbf{T}^r(\mathbf{e}), \mathfrak{g})$.

Proof. Write $\mathbf{e}' = \mathbf{T}^r(\mathbf{e}, \mathfrak{g})$ and assume (5.9) that is

$$\mathbf{e}' = \mathbf{g}_1\mathbf{r}_1\mathbf{g}_2\mathbf{r}_2 \dots \mathbf{r}_k\mathbf{g}_{k+1}.$$

The application of Lemma 5.8 for $\ell = k$ proves the statement, noting Lemma 5.7. □

What this statement says is the following. Assume that we performed a certain amount of swaps along the cycle decomposition of the original alternating circuit (using our decomposition algorithm) and we have the alternating circuit $\mathbf{T}^r(\mathbf{e}, \mathfrak{g})$ in our hands. Then, if we consider this alternating circuit as a totally fresh one and we use our decomposition algorithm, furthermore we perform our swap operations along this decomposition, then this procedure will process the red $\mathbf{r}_\ell$ subsequences one by one. But our problem here is that we do not know - yet - when this procedure processes fully all necessary $\mathbf{r}_k$s. In other words: when we should halt the algorithm.

However knowing the number of processed edges in the fully processed circuits of $\mathbf{e}'$ completely solves this problem, since we can use this parameter to halt our algorithm on $\mathbf{e}'$. And the size of the set of the possible numbers is simply quadratic. This set together with the polynomial running time of the algorithm named in (5.16) provides a polynomial means to determine $\mathbf{e}$ with its alternations.

One can ask the reason why this newly developed method is so effective. In the attempted approach described shortly at the beginning of Subsection 5.2 we tried to deal with all possible cycle decompositions of the circuits (this consist of all cycles and all their order). Analysis of the new method only requires consideration of a quadratic number of possible cycle decompositions.

5.3 Construction

If $X, Y \in V(\mathbb{G})$ let $E(X \triangle Y)$ be the symmetric difference of the edge sets $E(X)$ and $E(Y)$, set $E(X - Y) = E(X) \setminus E(Y)$, and $E(Y - X) = E(Y) \setminus E(X)$.

Before we describe the construction of our multicommodity flow we need some further definitions:

Definition 5.11. For $T \in V(\mathbb{G})$ let M_T be the bipartite $k \times l$ adjacency matrix of T. For $X, Y, Z \in V(\mathbb{G})$ write $\widehat{M}(X + Y - Z) = M_X + M_Y - M_Z$. (As we will see in the proof of the Key Lemma, these $k \times l$ matrices essentially *encode* the paths from X to Y along Z.)

If M and M' are $m \times m'$ matrices then let $\mathfrak{d}(M, M')$ be the number of non-zero elements in $M - M'$ (the well-known *Hamming distance*).

Outline of the construction of the path system. Fix a total order $\preceq$ on $U \times V$. This will induce a total order $\preceq'$ on all subsets of that product (namely we take the induced lexicographic order), in particular also on circuits in $[U, V]$. This will also induce a total order $\preceq^*$ on all sets of circuits in $[U, V]$ (we can take again the induced lexicographic order).

For each $X \neq Y \in V(\mathbb{G})$ do the following.

(A) Let $S_{X,Y} = \mathbb{S}(E(X-Y), E(Y-X))$. (This notation was introduced at (5.2).) To each $s \in \mathbb{S}(E(X-Y), E(Y-X))$ consider the *unordered* alternating circuit decomposition $\mathcal{C}_s$ of $(E(X - Y), E(Y - X))$. (This is described in Lemma 5.2.)

(B) Order $\mathcal{C}_s$ using $\preceq'$ to obtain the *ordered* alternating circuit decomposition

$$W_1^s, W_2^s \ldots, W_{k_s}^s$$

of $(E(X - Y), E(Y - X))$.

(C) Every W_i^s is an alternating circuit in the bipartite graph $(U \cup V, E(X-Y) \cup E(Y-X))$. Consider the enumeration $e_1 \ldots e_m$ of W_i^s, where e_1 is the $\prec'$-minimal edge in W_i^s, and e_2 is the smaller edge for $\prec'$ among its two neighboring edges, while e_m is the bigger. (This fixes uniquely the trail which traverses this circuit.) Now we can apply the method of Subsection 5.2 to determine the cycle decomposition of W_i^s for $1 \leqslant i \leqslant k_s$:

$$C_1^{s,i}, C_2^{s,i}, \ldots, C_{\ell_{s,i}}^{s,i}.$$

Actually, we obtain cycle $C_j^{s,i}$ as a sequence of edges. We keep this order to process $C_j^{s,i}$ further in (F).

(D) Let

$$C_1, C_2, \ldots, C_{m_s}.$$

be the short hand notation for the (alternating) cycle decomposition

$$C_1^{s,1}, C_2^{s,1}, \ldots, C_{\ell_{s,1}}^{s,1}, C_1^{s,2}, C_2^{s,2}, \ldots, C_{\ell_{s,2}}^{s,2}, \ldots, C_1^{s,k_s}, C_2^{s,k_s}, \ldots, C_{\ell_{s,k_s}}^{s,k_s}$$

of $E(X \triangle Y)$. We will call it a *canonical* cycle decomposition.

(E) For each cycle C in this decomposition we inherit an enumeration of that cycle (see (C)), which also determines a direction on the cycle. So for $a, b \in C$ we can define $[a, b]_C$ as the trail from a to b in C according to this fixed direction.

The following observation plays a crucial role in our method:

Observation 5.12. *The function s itself determines this canonical decomposition, and also determines the direction of the cycles in the decomposition. So we do not need to know $E(X-Y)$ and $E(Y-X)$ to compute the $C_j^{s,i}$, or even $[a,b]_{C_j^{s,i}}$ from s.*

(F) Let $\Upsilon(X,Y,s)$ be a path of realizations

$$X = G_0, G_1, \ldots, G_{n_1}, G_{n_1+1}, \ldots, G_{n_2}, \ldots, G_{n_{m_s}} = Y \tag{5.17}$$

in $\mathbb{G}$ from X to Y such that

(a) $n_{m_s} \leqslant c \cdot n^2$,

(b) $E(G_{n_i}) = \big(E(G_{n_{i-1}}) \cup (E(Y) \cap E(C_i)\big) \setminus \big(E(X) \cap E(C_i)\big)$ for $i = 1, 2, \ldots, m_s$,

(c) if for $i < m_s$ we denote the first vertex of the cycle C_{i+1} in the order inherited from the construction by a_{i+1}, then for each $n_i \leqslant j \leqslant n_{i+1}$ there is is a vertex b_j in C_{i+1} such that

$$\big|E(G_j) \triangle F\big| \leqslant \Omega_1,$$

where

$$F = \big(E(G_{n_i}) \cup \big([a_{i+1}, b_j]_{C_{i+1}} \cap E(Y)\big) \setminus \big([a_{i+1}, b_j]_{C_{i+1}} \cap E(X)\big)\big),$$

(d) for each $j = 1, 2, \ldots, n_{m_s}$ there is $T \in V(\mathbb{G})$ such that

$$\mathfrak{d}\left(\widehat{M}(X+Y-G_j), M_T\right) \leqslant \Omega_2,$$

where c, Ω_1 and Ω_2 are fixed "small" natural numbers. (Recall here, that by the definition of the Markov chain, in this path each graph $G_{\ell+1}$ is constructed from the previous one G_ℓ by a valid swap operation.)

Key Lemma 5.13. *Let $X \neq Y \in \mathbb{G}$. If we can assign paths*

$$\Big\langle \Upsilon(X,Y,s) : s \in \mathbb{S}(E(X-Y), E(Y-X)), X, Y \in V(\mathbb{G}) \Big\rangle$$

according to (A)-(F) *then (4.7) holds and so our Markov chain is rapidly mixing.*

Proof of the Key Lemma: Fix $Z \in V(\mathbb{G})$. We need to prove (4.7):

$$\sum_{X,Y \in V(\mathbb{G})} \frac{\big|\big\{s \in S_{X,Y} : Z \in \Upsilon(X,Y,s)\big\}\big|}{|S_{X,Y}|} \leqslant poly(n) \cdot N.$$

Let

$$\mathfrak{M} = \Big\{\widehat{M}(X+Y-Z) : Z \in \Upsilon(X,Y,s) \text{ for some } X, Y \in V(\mathbb{G}) \text{ and } \ s \in \mathbb{S}(X,Y)\Big\}.$$

By (F)(d) for each $\widehat{M} = \widehat{M}(X+Y-Z) \in \mathfrak{M}$ there is $T \in V(\mathbb{G})$ such that $\mathfrak{d}(\widehat{M}, M_T) \leqslant \Omega_2$, i.e. there are at most Ω_2 positions where M_T and $\widehat{M}(X+Y-Z)$ are different, so we have at most $(n^2)^{\Omega_2} = n^{2\Omega_2}$ difference sets. Furthermore every entry of $\widehat{M}(X+Y-Z)$ lies in the set $\{-1, 0, 1, 2\}$, so a fixed difference set we have most 3^{Ω_2} possibilities. So

$$|\mathfrak{M}| \leqslant |V(\mathbb{G})| \cdot n^{2\Omega_2} \cdot 3^{\Omega_2} \leqslant \mathrm{poly}(n) \cdot |V(\mathbb{G})| = \mathrm{poly}(n) \cdot N.$$

For $\widehat{M} \in \mathfrak{M}$ let

$$\mathfrak{X}(Z, \widehat{M}) = \left\{(X, Y, s) : s \in \mathbb{S}(X, Y),\ Z \in \Upsilon(X, Y, s),\ \widehat{M}(X+Y-Z) = \widehat{M}\right\}. \quad (5.18)$$

Since $|\mathfrak{M}| \leqslant \mathrm{poly}(n) \cdot N$, if we can prove that

$$\sum_{(X,Y,s) \in \mathfrak{X}(Z,\widehat{M})} \frac{1}{|S_{X,Y}|} \leqslant poly(n) \quad (5.19)$$

for all $\widehat{M} \in \mathfrak{M}$, then (4.7) holds.

To verify (5.19) fix $\widehat{M} \in \mathfrak{M}$. Let $(X, Y, s) \in \mathfrak{X}(Z, \widehat{M})$ be arbitrary. Since $M_Z + \widehat{M} = M_X + M_Y$, we can compute

$$\Delta = E(X \triangle Y)$$

from Z and $\widehat{M}$. Denote by $(2d_1, \ldots, 2d_h)$ the degree sequence of $E(X \triangle Y)$. Put

$$t_\Delta = \prod_1^h (d_i!).$$

Clearly

$$t_\Delta = |S_{X,Y}|,$$

and so

$$\sum_{(X,Y,s) \in \mathfrak{X}(Z,\widehat{M})} \frac{1}{|S_{X,Y}|} = \sum_{(X,Y,s) \in \mathfrak{X}(Z,\widehat{M})} \frac{1}{t_\Delta} = \frac{\left|\mathfrak{X}(Z, \widehat{M})\right|}{t_\Delta}.$$

Thus to prove (5.19) we need to show that

$$\left|\mathfrak{X}(Z, \widehat{M})\right| \leqslant poly(n) \cdot t_\Delta. \quad (5.20)$$

Let

$$\mathcal{S} = \left\{s :\ \text{for some } (X, Y) \text{ we have } (X, Y, s) \in \mathfrak{X}(Z, \widehat{M})\right\}. \quad (5.21)$$

To get (5.20) it is enough to show the following statement.

Lemma 5.14. *For each possible Z and the corresponding set $\mathcal{S}$ we have:*

(a) $|\mathcal{S}| \leqslant poly(n) \cdot t_\Delta$,

(b) *for each* $s \in \mathcal{S}$ *we have*

$$\left|\left\{(X,Y) : (X,Y,s) \in \mathfrak{X}(Z,\widehat{M})\right\}\right| \leqslant poly(n). \tag{5.22}$$

To prove this lemma fix $(X,Y,s) \in \mathfrak{X}(Z,\widehat{M})$. We should recall the construction of the path $\Upsilon(X,Y,s)$ which can be demonstrated as:

$$W_1, \overbrace{\underbrace{C_1^k, \quad \cdots \quad, C_\ell^k, \quad \cdots \quad, C_{\ell_k}^k}_{G_1^{k,\ell},\ \cdots,\ G_m^{k,\ell},\ \cdots,\ G_{m_{k,\ell}}^{k,\ell}}}^{\cdots \quad \cdots \quad W_k, \quad \cdots \quad \cdots} W_{k_s} \tag{5.23}$$

where

(1) we consider first the circuit decomposition of $(E(X-Y), E(Y-X))$ determined by s:

$$W_1, W_2 \ldots, W_{k_s};$$

(2) then, using the method of subsection 5.2 for each $1 \leqslant k \leqslant k_s$ we define an alternating circuit decomposition of W_k:

$$C_1^k, C_2^k, \ldots, C_{\ell_k}^k;$$

(3) then in (F) for each $1 \leqslant k \leqslant k_s$ and $1 \leqslant \ell \leqslant \ell_k$ we define a sequence of elements of $\mathbb{G}$:

$$G_1^{k,\ell}, \ldots, G_m^{k,\ell}, \ldots G_{m_{k,\ell}}^{k,\ell},$$

such that

$$E\left(G_1^{k,\ell}\right) = \left[E(Y) \cap \Big(\bigcup_{k'<k} E(W_{k'}) \cup \bigcup_{\ell'<\ell} E\left(C_{\ell'}^k\right) \Big)\right] \bigcup \left[E(X) \cap \Big(\bigcup_{k'>k} E(W_{k'}) \cup \bigcup_{\ell' \geqslant \ell} E\left(C_{\ell'}^k\right) \Big)\right], \tag{5.24}$$

and $G_{m_{k,\ell}}^{k,\ell} = G_1^{k,\ell+1}$ if $\ell < \ell_k$, $G_{m_{k,\ell_k}}^{k,\ell_k} = G_1^{k+1,1}$ if $k < k_s$, and $G_{m_{k_s,\ell_{k_s}}}^{k_s,\ell_{k_s}} = Y$ (the equation 5.24 is just a reformulation of (F)(b));

(4) finally $\Upsilon(X,Y,s)$ is the path

$$X = G_1^{1,1}, G_2^{1,1}, \ldots, G_m^{k,\ell}, \ldots, G_{m_{k_s,\ell_{k_s}}}^{k_s,\ell_{k_s}} = Y \tag{5.25}$$

in $\mathbb{G}$ from X to Y (see (5.17)). As we observed in (3) above, $G_{m_{k,\ell}}^{k,\ell} = G_1^{k,\ell+1}$ if $\ell < \ell_k$, $G_{m_{k,\ell_k}}^{k,\ell_k} = G_1^{k+1,1}$ if $k < k_s$. We include just one copy of each of these graphs in the sequence above.

Fix k, ℓ, m such that $Z = G_m^{k,\ell}$, which means that we are processing the ℓth cycle from the kth circuit.

By (F)(c) there are two vertices a and b in C_ℓ^k such that

$$\left|E(G_m^{k,\ell}) \triangle F\right| \leqslant \Omega_1, \tag{5.26}$$

where

$$F = \left(E\left(G_1^{k,\ell}\right) \cup \left([a,b]_{C_\ell^k} \cap E(Y)\right) \setminus \left([a,b]_{C_\ell^k} \cap E(X)\right)\right). \tag{5.27}$$

To prove Lemma 5.14 (b) we show that

(†) *there is a function Ψ and a parameter set $\mathbb{B}$ such that $\mathbb{B}$ has* poly(n) *elements, and for each $(X,Y,s) \in \mathfrak{X}(Z,\widehat{M})$ there is $B \in \mathbb{B}$ such that*

$$\Psi\left(Z, \widehat{M}(X+Y-Z), s, B\right) = (X,Y). \tag{5.28}$$

Recall that $Z = G_m^{k,\ell}$ so we have $E\left(G_m^{k,\ell}\right)$. If we choose the parameter B as the quadruple $\left(i,a,b,\ E\left(G_m^{k,\ell}\right)\Delta F\right)$, then we can compute $F = E(G_m^{k,\ell})\Delta\big(E(G_m^{k,\ell})\Delta F\big)$ using this parameter, and so

$$\begin{aligned} E(X) \setminus E(Y) = {} & \left([a,b]_{C_\ell^k} \setminus F\right) \bigcup \left([b,a]_{C_\ell^k} \cap F\right) \bigcup \\ & \left[\Big(\bigcup_{k'<k} E(W_{k'}) \cup \bigcup_{\ell'<\ell} E\left(C_{\ell'}^k\right)\Big) \setminus F\right] \bigcup \left[F \cap \Big(\bigcup_{k'>k} E(W_{k'}) \cup \bigcup_{\ell'\geqslant\ell} E\left(C_{\ell'}^k\right)\Big)\right]. \end{aligned} \tag{5.29}$$

Since $i \leqslant n^2$, $a,b \leqslant n$ and $\big(E(X)\setminus E(Y)\big) \triangle F$ is an at most Ω_1 element subset of $[U\cup V]^2$, the size of the parameter set is polynomial:

$$|\mathbb{B}| \leqslant n^2 \cdot n \cdot n \cdot (n^2)^{\Omega_1}.$$

Since Z and $\widehat{M}(X+Y-Z)$ determine $E(X) \cap E(Y)$, we can compute $E(X)$. Similarly we can compute $E(Y)$. So we verified (†), and so Lemma 5.14 (b) holds.

Now we turn to prove Lemma 5.14 (a). We will do it in steps (a1) – (a3).

(a1) *Each function $s \in \mathcal{S}$, which corresponds to the circuit decomposition*

$$W_1, \ldots W_{k-1}, W_k, W_{k+1} \ldots, W_{k_s}$$

(see Lemma 5.2), is computable - using a small parameter set - from the function s' corresponding to the circuit decomposition

$$W_1, \ldots W_{k-1}, \mathbf{T}^{\ell-1}(W_k, \mathfrak{g}), W_{k+1} \ldots, W_{k_s}.$$

Indeed, by Theorem 5.10, $\mathbf{T}^t(\mathbf{T}^{\ell-1}(W_k,\mathfrak{g}),\mathfrak{g}) = W_k$ for some t. So, as we described after Theorem 5.10, the parameter k and the number of the processed edges in circuit W_k together determine fully $\mathbf{e}$, and $k \leqslant n^2$ and the number of processed edges is also $\leqslant n^2$. So **(a1)** holds. $\square_{(a1)}$

We need some preparation before we can formulate and prove (a2): Recall that Lemma 5.5 (ii) infers

$$s' \in \mathbb{S}\left(\Delta \cap E\left(G_1^{k,\ell}\right), \Delta \setminus E\left(G_1^{k,\ell}\right)\right).$$

The sequence

$$\mathbf{e}' = e_1 e_2 \dots e_\mu = \mathbf{T}^{\ell-1}(W_k,\mathfrak{g})$$

is an alternating circuit in $G_1^{k,\ell}$. (All circuits of the decomposition with $k' < k$ are already fully processed. No circuit after W_k is touched yet. So it is enough to consider only this.) We use the notations $f = f^{\ell-1}(W_k)$ and

$$\mathbf{e}' = \mathbf{g}_1\mathbf{r}_1 \dots \mathbf{g}_u$$

where $\mathbf{g}_s$ are maximal $\mathbf{green}_f$, and $\mathbf{r}_o$ are maximal $\mathbf{red}_f$ intervals. We know that the current cycle:

$$C_\ell^k = \mathbf{green}_f[e_i, e_j]$$

(for some $0 \leqslant i < j \leqslant n$) is undergoing a series of swaps operations which will exchange its edges between the realizations X and Y. When this swap sequence is completed then the processing of this cycle in the cycle decomposition of circuit W_k will be done, and the coloration of its edges will become $\mathbf{red}_{f^\ell}$. Now the assumption (F)(c) about our swap sequence generation, applying for $G_m^{k,\ell}$, gives us an interval

$$[a,b]_{C_\ell^k} = \mathbf{green}_f[e_i, e_{j'}],$$

for some $i \leqslant j' \leqslant j$.

Assume that $e_i \in \mathbf{g}_o$ and $e_{j'} \in \mathbf{g}_t$. Write $\mathbf{g}_o = \mathbf{g}_{o,0}\mathbf{g}_{o,1}$, where e_i is the first edge of $\mathbf{g}_{o,1}$, and write $\mathbf{g}_t = \mathbf{g}_{t,0}\mathbf{g}_{t,1}$, where $e_{j'}$ is the last edge of $\mathbf{g}_{t,0}$.

Consider the sequence

$$\mathbf{e}'' = \mathbf{g}_1\mathbf{r}_1 \dots \mathbf{r}_{o-1}\mathbf{g}_o\overleftarrow{\mathbf{r}_o}\mathbf{g}_{o+1} \dots \overleftarrow{\mathbf{r}_{t-1}}\mathbf{g}_t\mathbf{r}_t \dots \mathbf{g}_u.$$

Now $\mathbf{e}''$ is the concatenation of three $(\Delta \cap F, \Delta \setminus F)$-alternating paths: $\mathbf{g}_1\mathbf{r}_1 \dots \mathbf{r}_{o-1}\mathbf{g}_{o,0}$ and $\mathbf{g}_{o,1}\overleftarrow{\mathbf{r}_o}\mathbf{g}_{o+1} \dots \overleftarrow{\mathbf{r}_{r-1}}\mathbf{g}_{t,0}$, and finally $\mathbf{g}_{t,1}\mathbf{r}_{t+1} \dots \mathbf{g}_u$. However in general this trail is not necessarily alternating, because on the border of $\mathbf{g}_{t,0}$ and $\mathbf{g}_{t,1}$ furthermore on the border of $\mathbf{g}_{t,0}$ and $\mathbf{g}_{t,1}$ is not alternating anymore. (However, if one or both of the red circuits $\mathbf{r}^-$ and/or $\mathbf{r}^+$ exist then this problem will not occur there (see equation (5.11)).

(a2) *The trail* $\mathbf{e}'$ *is computable* (*using a small parameter set*) *from* $\mathbf{e}''$.

Indeed, $\mathbf{e}^+ = \mathbf{g}_{o,1}\overleftarrow{\mathbf{r}_o}\ldots\overleftarrow{\mathbf{r}_{t-1}}\mathbf{g}_{t,0}$ is computable from $\mathbf{e}''$ because it is a subsequence. Since

$$[e_i, e_j] = \mathbf{g}_{o,1}\mathbf{r}_o\ldots\mathbf{r}_{t-1}\mathbf{g}_{t,0}\mathbf{g}_{t,1}\ldots\mathbf{r}_{t+1}\ldots e_j$$

is a circuit, we can apply Theorem 5.10 to find some $v \leqslant n^2$ such that $\mathbf{T}^v(\mathbf{e}^+, \mathfrak{g}) = \mathbf{g}_{o,1}\mathbf{r}_{o+1}\ldots\mathbf{r}_{t-1}\mathbf{g}_{t,0}$. Thus $[e_i, e_j]$ (in $\mathbf{e}'$) is computable from $\mathbf{e}''$. Since $\mathbf{e}'$ and $\mathbf{e}''$ agree outside $[e_i, e_j]$, we proved **(a2)**.

We turn our attention now to the third obstacle: till now we showed that knowledge of $[a, b]_{C_\ell^k}$ would determine fully s from s'. However we do not know exactly the sequence s', since the assumption of (F) (c) allowed that $\left|E(G_j) \triangle F\right| \leqslant \Omega_1$, so a small number of edges of the current realization are not on the alternating path determined by s'. Next we will deal with this problem:

(a3) *The sequence s'' corresponding to the circuit decomposition*

$$W_1, \ldots, W_{\ell-1}, \mathbf{e}'', W_{\ell+1}, \ldots, W_k$$

is "almost" in $\mathbb{S}\left(\Delta \cap E\left(Z\right), \Delta \setminus E\left(Z\right)\right)$, (see formula 5.30 below) so it is computable from some element of $\mathbb{S}\left(\Delta \cap E\left(Z\right), \Delta \setminus E\left(Z\right)\right)$ using a small parameter set.

Recall first that $Z = G_m^{k,\ell}$. Let

$$E^* = \left(E\left(Z\right) \triangle F\right) \cup C_\ell^k(\mathrm{b}(e_i)) \cup C_\ell^k(\mathrm{t}(e_{j'})).$$

The last two expressions stand for the two edge pairs which are adjacent to the vertices $\mathrm{b}(e_i)$ and $\mathrm{t}(e_{j'})$ in the actual cycle C_ℓ^k. Since C_ℓ^k is a cycle indeed, we have $|E^*| \leqslant \Omega_1 + 4$ (due to (5.26)).

For $w \in U \cup V$, let

$$t(w) = s''(w) \cap \left[\left(\Delta \cap E\left(Z\right)\right)(w), \left(\Delta \setminus E\left(Z\right)\right)(w)\right],$$

i.e $t(w)$ is those elements of $s''(w)$ which alternate between $\Delta \cap E\left(Z\right)$ and $\Delta \setminus E\left(Z\right)$. Since $t(w)$ is a set of independent edges in

$$H = \left[\left(\Delta \cap E(Z)\right)(w), \left(\Delta \setminus E\left(Z\right)\right)(w)\right],$$

we can find a perfect matching extension $s^*(w)$ of $t(w)$ in the complete bipartite graph H, because both side of the H have the same number of vertices. Then for this perfect matching we have

$$s^* \in \mathbb{S}\left(\Delta \cap E\left(Z\right), \Delta \setminus E\left(Z\right)\right).$$

Next we show that the difference between s'' and s^* is small, namely

$$\sum_{w \in U \cup V} |s''(w) \triangle s^*(w)| \leqslant 4\Omega_1 + 16. \tag{5.30}$$

For that end let $w \in U \cup V$ and $(e, e') \in s''(w)$ such that $e, e' \notin E^*$. Then, by the definition of E^*, we have

$$e \in \Delta \cap E(Z) \Leftrightarrow e \in \Delta \cap F \quad \text{and} \quad e' \in \Delta \cap E(Z) \Leftrightarrow e' \in \Delta \cap F.$$

So since (e, e') alternates between $\Delta \cap F$ and $\Delta \setminus F$, (e, e') also alternates between $\Delta \cap E(Z)$ and $\Delta \setminus E(Z)$. So $(e, e') \in t(w) \subset s^*(w)$, and so $(e, e') \notin s''(w) \triangle s^*(w)$. Thus

$$\sum_{w \in U \cup V} |s''(w) \setminus s^*(w)| \leqslant$$
$$\Big| \bigcup_{w \in U \cup V} \Big\{ (w, e, e') : (e, e') \in s''(w), e \in E^* \text{ or } e' \in E^* \Big\} \Big| \leqslant$$
$$2 \cdot |E^*| \leqslant 2\Omega_1 + 8. \quad (5.31)$$

Since $|s''(w)| = |s^*(w)|$, we have $2 \cdot |s''(w) \setminus s^*(w)| = |s''(w) \triangle s^*(w)|$, so (5.31) gives (5.30).

Therefore s^* together with a small parameter set which describes the symmetric differences $s''(w) \triangle s^*(w)$ for $w \in U \cup V$ determines completely s'', thus **(a3)** is true as well.

Putting together (a1)–(a3) we obtain

$$\begin{aligned} |\mathcal{S}| &\leqslant poly_1(n) \cdot poly_2(n) \cdot poly_3(n) \cdot \big|\mathbb{S}\big(\Delta \cap E(Z), \Delta \setminus E(Z)\big)\big| \\ &= poly(n) \cdot t_\Delta. \end{aligned}$$

So Lemma 5.14 (a) holds, which in turns completes the **proof of the Key Lemma**. □

We try to carry out the plan we just described. So:

- Fix $X \neq Y \in V(\mathbb{G})$.
- Pick $s \in \mathbb{S}(X, Y)$.
- s gives an alternating cycle decomposition

$$C_0, C_1, \ldots, C_\ell \quad (5.32)$$

of $E(X \triangle Y)$.

We want to define a path

$$X = G_0, \ldots, G_i, \ldots, G_m = Y \quad (5.33)$$

from X into Y in $\mathbb{G}$ - denoted by $\Upsilon(X, Y, s)$ - such that

(i) the length of this path is $\leqslant c \cdot n^2$ (where c is a suitable constant),

(ii) for some increasing indices $0 < n_1 < n_2 < \dots n_\ell$ we have $G_{n_i} = H_i$, where

$$E(H_i) = E(X) \triangle \left(\bigcup_{i' < i} E(C_{i'}) \right). \tag{5.34}$$

So we have certain "fixed points" of our path $\Upsilon(X, Y, s)$, and this observation reduces our task to the following:

- for each $i < \ell$ construct the path

$$H_i = G'_0, G'_1, \dots, G'_{m'} = H_{i+1} \tag{5.35}$$

between G_{n_i} and $G_{n_{i+1}}$ such that $m' \leqslant c \cdot |C_i|$ and (F)(d) holds, i.e. for each j there is $K_j \in V(\mathbb{G})$ such that $\mathfrak{d}\left(\widehat{M}(X, Y, G'_j), K_j\right) \leqslant \Omega_2$.

From now on we work on that construction. To simplify the notation we write $G = H_i$ and $G' = H_{i+1}$. We know that the symmetric difference of G and G' is just the cycle C_i. Now we are in the following situation:

Generic situation - construction of a path along a cycle

(i) $X, Y, G, G' \in V(\mathbb{G})$.

(ii) The symmetric difference of $E(G)$ and $E(G')$ is a cycle C.

(iii) the symmetric differences $E(X \triangle G)$, $E(G \triangle G')$ and $E(G' \triangle Y)$ are pairwise disjoint.

Construct a path

$$G = G_0, \dots, G_m = G' \tag{5.36}$$

in the graph $\mathbb{G}$ of all realizations such that

(I) $m \leqslant c \cdot |C|$, and the requirement of (F)(c) also holds,

(II) for each j there is $K_j \in V(\mathbb{G})$ such that $\mathfrak{d}\left(\widehat{M}(X, Y, G_j), M_{K_j}\right) \leqslant \Omega_2$.

We will carry out this construction in the next sections. The burden of such a construction is to meet requirement (II). In [7] and in [2] the regularity of the realizations was used.

The **"friendly path method"**.
In the next sections we describe a new general method based on the notion of *friendly paths* (see Definition 6.3) to construct the paths $\Upsilon(X, Y, s)$.

The novelty of our friendly path method can be summarized as follows:

- if our bipartite degree sequence is half-regular then the paths $\Upsilon(X, Y, s)$ satisfy the previous condition (II)
- if our bipartite degree sequence is arbitrary, then $\Upsilon(X, Y, s)$ satisfies (II) provided the symmetric difference of X and Y is a cycle.

Originally we conjectured, that our friendly path method always produces paths which satisfy (II). However we were unable to prove it, and now we think that essentially new ideas are needed to prove the case of general bipartite degree sequences.

6 Multicommodity flow - along a cycle

Let X, Y and Z be three realizations of a given bi-graphical degree sequence. Assume that $E(X) \cap E(Y) \subset E(Z)$, furthermore $E(Z) \subseteq E(X) \cup E(Y)$. Then the realization Z is an **intermediate** realization between X and Y.

In this section we describe the construction a path along an alternating cycle C. Here we have the intermediate realizations G and G' between X to Y, and these two realizations differ only in this cycle C, where $G \cap C = X \cap C$ and $G' \cap C = Y \cap C$. At the beginning of this phase our canonical path is between X and G. Along the process we extend it to reach realization G'. Within the process all swaps will happen between vertices $V(C)$ of the cycle C and the end of the process each chord will be at the same state as it was at the beginning, except the edges along the cycle, where the X-edges will be exchanged by the Y-edges.

In what follows we will imagine our cycles as convex polygons in the plane, and we will denote by the vertices of any particular cycle of 2ℓ edges with $u_1, v_1, u_2, v_2, \ldots, u_\ell, v_\ell$. The edges of the cycle are $(u_1, v_1), (v_1, u_2), \ldots, (u_\ell, v_\ell)$, (v_ℓ, u_1) and they belong alternately to X and Y. All the other (possible, but not necessarily existing) edges among vertices of a particular cycle are the *chords*. (In other words we will use the notion of chord if we want to emphasis that we do not know whether the two vertices form an edge or not in the current graph.) A chord is a *shortest* one, if in one direction there are only two vertices (that is three edges) of the cycle between its end points. The middle edge of this three is the *root* of the chord.

W.l.o.g. we may assume that (u_1, v_1) is an edge in G while (v_1, u_2) belongs to G'. We are going to construct now a sequence of graphical realizations between G and G' such that any two consecutive elements in this sequence differ from each other in one swap operation. The general element of this sequence will be denoted by Z.

We have to control which graphs belong to this sequence. For that purpose we assigned a matrix $\widehat{M}$ to each graph Z. If G is a vertex in $\mathbb{G}$ then M_G denotes the adjacency matrix of the bipartite realization G where the columns are indexed by the vertices of V, numbered from left to right, and the rows are indexed by the vertices of U, numbered from bottom to top. Hence the entry in row i, column j of the matrix will be written as (j, i) and corresponds to the chord (v_j, u_i). With some abuse of notation we also will use the word "chord" to refer to the matrix position as well. This is nonstandard notation for the entries of a matrix, but matches the Cartesian coordinate system. Then let

$$\widehat{M}(X + Y - Z) = M_X + M_Y - M_Z.$$

By definition each entry of an adjacency matrix is 0 or 1. Therefore only $-1, 0, 1, 2$ can be the entries of $\widehat{M}$. An entry is -1 if the corresponding edge is missing from both X and Y but it exists in Z. The entry is 2 if the corresponding edge is missing from Z but exists in both X and Y. The entry is 1 if the corresponding edge exists in all three graphs (X, Y, Z) or it is there only in one of X and Y but not in Z. Finally it is 0 if the corresponding edge is missing from all three graphs, or the edge exists in exactly one of X and Y and is also present in Z. (Therefore if a chord denotes an existing edge in exactly one of X and Y then the entry corresponding to this chord is always 0 or 1.)

Observation 6.1. *Let X, Y and Z be some realizations of a bipartite degree sequence.*

(i) *The row and column sums of $\widehat{M}(X+Y-Z)$ are the same as the row and column sums in M_X (or M_Y or M_Z).*

(ii) *If Z is an intermediate realization between X and Y then $\widehat{M}(X+Y-Z)$ is another realization of the same degree sequence (and all entries are 0 or 1).*

Before we define some further notions we introduce our main tool that we will use later in this paper to illustrate different procedures in our current realizations.

Usually each cycle under processing is small comparing with the full graph, therefore we always consider a "comfortably reordered" adjacency matrix (in other words, we apply a suitable permutation on the vertices) such that the vertices forming the cycle will be associated to an $\ell \times \ell$ submatrices of our adjacency matrices, and our figures will show only these submatrices. The positions $(1,1), \ldots, (\ell,\ell)$ form the **main-diagonal** while the positions right above the main-diagonal as well as the rightmost bottom one (these are $(1,2), (2,3), \ldots, (\ell-1,\ell)$ finally $(\ell,1)$) form the **small-diagonal**. (This placement was our goal using this numbering system for rows and columns. For example, the element $(1,2)$ corresponds to the chord (v_1, u_2). If this is 1, then there is an edge there, otherwise the edge is missing.)

Now we introduce a new tool to give a slightly different view about this "central region". Let Z be a realization such that the symmetric difference of $E(Z)$ and $E(G)$ only involves vertices of C. The new tool is the $\ell \times \ell$ matrix F_Z defined as follows. In $V(C)$ (so at this central region) for $i, j = 1, \ldots, \ell$ we have:

$$F_Z(j,i) = \begin{cases} M_Z(j,i) & \text{if } (j,i) \in \text{ main- or small-diagonals,} \\ \left[M_G + M_{G'} + M_Z\right](j,i) & \text{otherwise.} \end{cases}$$

In that way in the main- and small-diagonal's elements are 0 or 1 while the others (the **off-diagonal** entries) can be $0, 1, 2, 3$. There is an easy algorithm to construct F_Z from the corresponding $\widehat{M}(G+G'-Z)$ and vice versa (please recognize that here we use G and G' instead of X and Y): In the main-diagonal and in the small-diagonal the zeros and ones must be interchanged. Outside of these diagonal entries $-1, 0, 1, 2$ of $\widehat{M}(G+G'-Z)$ become $1, 0, 3, 2$ in F_Z. (In case we need a second realization, similar to Z, we will denote it with Z'.)

Since G and G' coincide outside the alternating cycle C therefore the off-diagonal elements in F_Z are odd when the edge exists in the actual Z and even otherwise. When $Z = G$ then the main-diagonal entries are 1 while the small-diagonal elements are 0. This matrix F_Z will be used in our illustrating figures and also to conduct the construction of our canonical path system.

We are ready now to introduce the central notions of our proof:

Definition 6.2. The **type** of a chord is 1 if it is present in G, and 0 otherwise. Note that a chord is present in G if and only if it is present in G'. Let (v_β, u_α) be a chord so

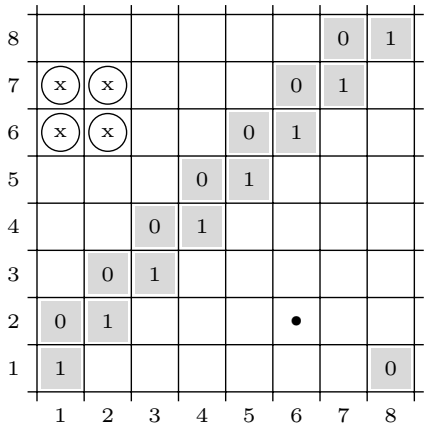
$\delta \notin \{\beta, \beta+1\}$. A chord (v_β, u_α) is a **cousin** of a chord (v_δ, u_ϵ), if the other two corners of the submatrix, which is spanned by this position and the chord are on the main- or on the small-diagonals of F_Z (see Figure 1). We can describe it with formulae as well: this chord (v_β, u_α) is a **cousin** of a chord (v_δ, u_ϵ), if $\alpha \notin \{\beta, \beta+1\}$ and one of the following holds:

$$\begin{cases} \epsilon < \delta, & \alpha \in \{\delta, \delta+1\} \text{ and } \beta \in \{\epsilon-1, \epsilon\}, \\ \epsilon > \delta, & \alpha \in \{\delta-1, \delta\} \text{ and } \beta \in \{\epsilon, \epsilon+1\}. \end{cases}$$

A chord e is **friendly** if at least one of its cousins has the same type as e itself, otherwise it is **unfriendly**. (Please recall that here "chord" also refers to the position itself within the matrix therefore we also say that the position is friendly.)

Now Figure 1 illustrates the cousins of the chord (v_6, u_2) in the initial realization $Z = G$. (They are (v_1, u_6), (v_1, u_7), (v_2, u_6), finally (v_2, u_7) and let's recall that the word chord indicates that the definition does not depend on the actual existence or non-existence of that edge.)

Figure 1: A chord and its cousins



Before the next important definition we introduce a metric on pairs of positions of this matrix: $\|A, \bar{A}\|$ says how many steps are necessary to go from A to $\bar{A}$ if in every step we can move to a (horizontally or vertically) neighboring position, we cannot cross the main-diagonal, finally the position $(i, 1)$ is neighboring to (i, ℓ) and analogously (ℓ, i) is neighboring to $(1, i)$.

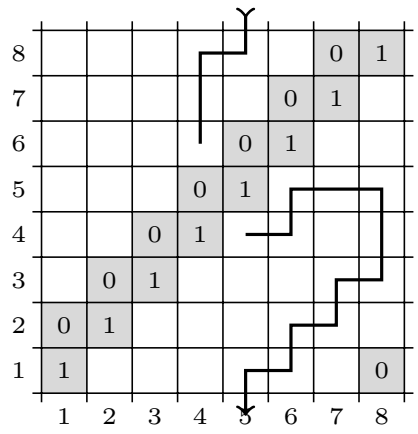
Definition 6.3. A sequence of pairwise distinct positions $A_1, \ldots, A_j$ is a **friendly path** in F_G if

(i) each position is friendly (in the matrix F_G),

(ii) $\|A_h, A_{h+1}\| = 1$,

(iii) the chords e_1 and e_j - defined by the positions A_1 and A_j - are shortest chords and the root of e_1 belongs to G while the root of e_j does not.

Figure 2: A friendly path



A friendly path goes from the main-diagonal to the small-diagonal and it can be quite complicated, and it is important to remark that such a friendly path is NOT a path in a particular graph. Furthermore the friendly path is **fixed** for the entire process determining the swap sequence from realization G to realization G', while the notions of chord or cousin apply for each matrix F_Z along the swap sequence.

The name is justified by the image of the friendly path in the illustration of F_G, shown in Figure 2. (It shows the path itself, but it does not show why the individual elements of the path are friendly.) The figures like this are not for illustration only: whenever we consider a friendly path we always work on the matrix itself.

6.1 The case that a friendly path exists

In this subsection we describe the construction of the path along this cycle in the case that a friendly path exists. Fix one friendly path: if there are more than one, then take, say, the lexicographically smallest one (relative to the subscripts of the positions). Let the chords of the existing friendly path correspond to the positions $A_1, \ldots, A_\Lambda$ where $A_j = (a_j^1, a_j^2)$.

By definition our friendly path has the following properties: (i) $a_j^1 \neq a_j^2$ and $a_j^1+1 \neq a_j^2$, (ii) $\|A_j, A_{j+1}\| = 1$, and finally (iii) A_1 is at distance 1 from the main-diagonal, while A_Λ is at distance 1 from the small-diagonal.

First we introduce two new structures:

Definition 6.4. Let $1 \leqslant \alpha, \beta \leqslant \ell$ with $\beta \notin \{\alpha, \alpha+1\}$. We say an $\ell \times \ell$-matrix $F_Z = (m_{i,j})$ is (α, β)**-OK matrix** iff

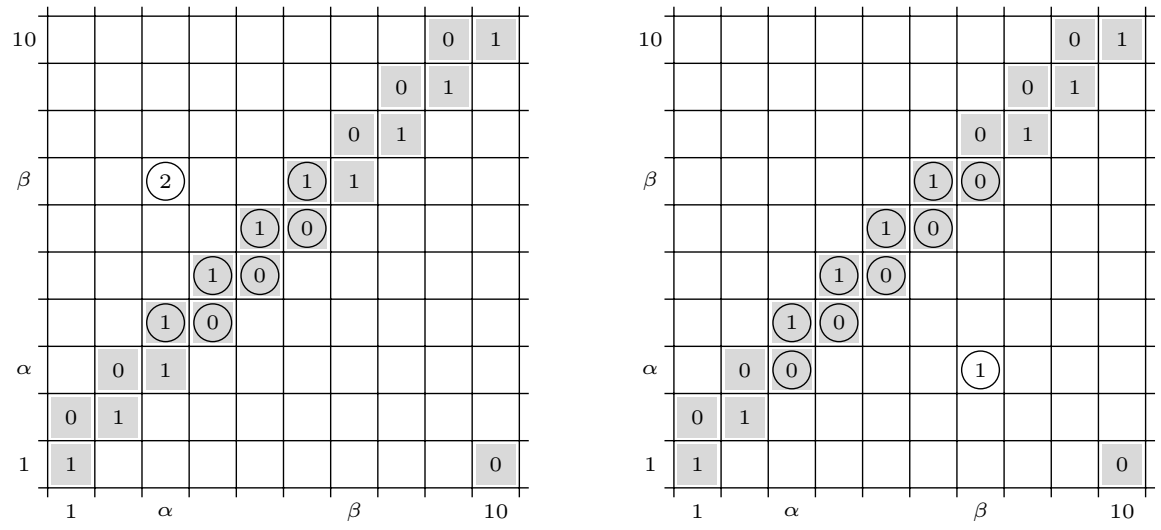
(i) $m_{\alpha,\beta} = 2$,

(ii) $m_{i,i} = \begin{cases} 0 & \text{for } i = \alpha+1, \alpha+2, \ldots, \beta-1, \\ 1 & \text{for } i = \beta, \beta+1, \ldots, \alpha-1, \alpha, \end{cases}$

(iii) $m_{i,i+1} = \begin{cases} 1 & \text{for } i = \alpha, \alpha+1, \ldots, \beta-2, \beta-1, \\ 0 & \text{for } i = \beta, \beta+1, \ldots, \alpha-2, \alpha-1. \end{cases}$

(See the LHS of Figure 3.) Please recall that the entry 2 in F_Z is an edge which is missing from Z but exists in both G and G' (the off-diagonal entries are the same in M_G and $M_{G'}$).

Figure 3: An (α, β)-OK matrix and an (β, α)-KO matrix



Definition 6.5. Let $1 \leqslant \alpha, \beta \leqslant \ell$ with $\beta \notin \{\alpha-1, \alpha\}$. We say an $\ell \times \ell$-matrix $F_Z = (m_{i,j})$ is (β, α)**-KO matrix** iff

(i) $m_{\beta,\alpha} = 1$,

(ii) $m_{i,i} = \begin{cases} 0 & \text{for } i = \alpha, \alpha+1, \ldots, \beta-1, \beta, \\ 1 & \text{for } i = \beta+1, \ldots, \alpha-1, \end{cases}$

(iii) $m_{i,i+1} = \begin{cases} 1 & \text{for } i = \alpha, \alpha+1, \ldots, \beta-1, \\ 0 & \text{for } i = \beta, \beta+1, \ldots, \alpha-2, \alpha-1. \end{cases}$

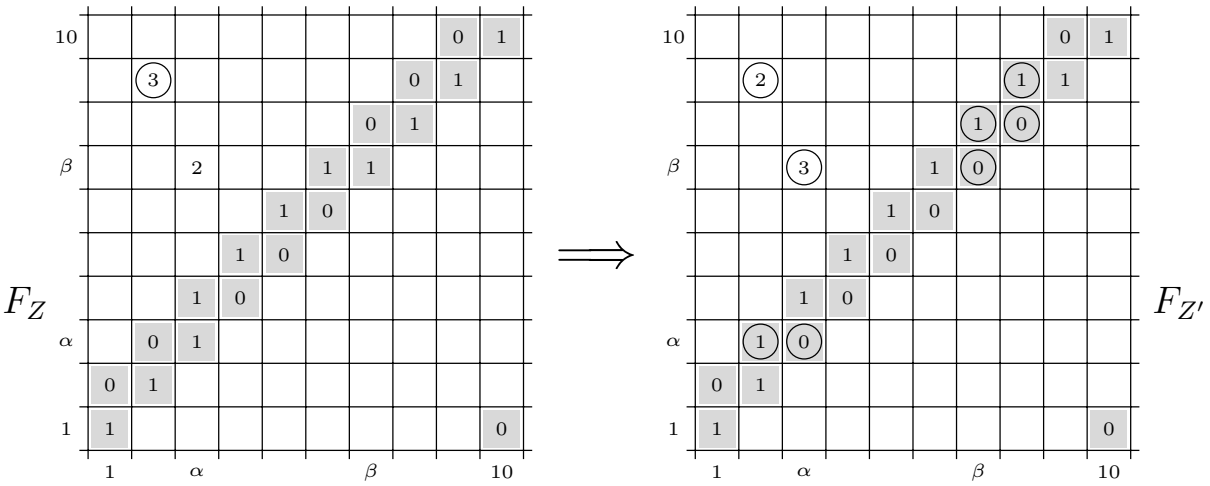
(See the RHS of Figure 3.) Please recall that the entry 1 in F_Z is an edge which exists in Z but missing from both G and G'.

Lemma 6.6. *Let $F_Z = (m_{i,j})$ be an (α, β)-OK matrix and $m_{\alpha-1,\beta+2} = 3$. Assume that $F_{Z'} = (m'_{i,j})$ is an $(\alpha-1, \beta+2)$-OK matrix such that*

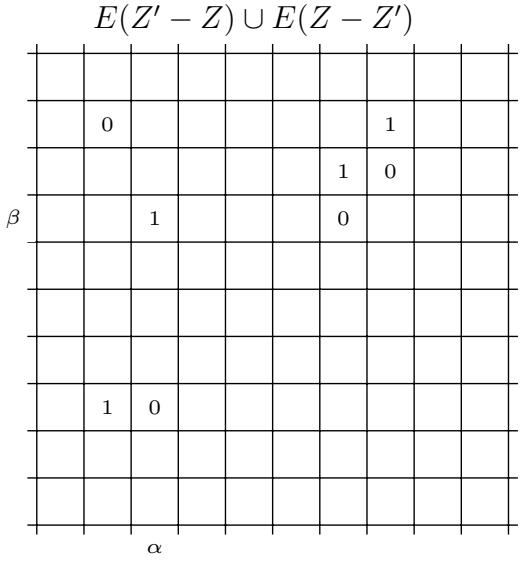
(1) $m'_{\alpha,\beta} = 3$,

(2) $m'_{i,j} = m_{i,j}$ *if* $i \neq j$, $i+1 \neq j$, *and* $(i,j) \neq (\alpha, \beta)$, $(\alpha-1, \beta+2)$.

Then there exists an absolute constant Θ such that one can transform Z to Z' by at most Θ swaps (and, meanwhile, transform F_Z into $F_{Z'}$).



Proof: : It is enough to observe that the symmetric difference of Z and Z' is a single alternating cycle. Indeed, in the next figure the entry 1 indicates edges in $E(Z' - Z)$ and the entry 0 indicates edges in $E(Z - Z')$. (The non-empty positions of this figure are the circled positions in the previous matrix $F_{Z'}$.)



Therefore
$(\alpha - 1, \alpha), (\alpha, \alpha), (\alpha, \beta),$
$(\beta, \beta), (\beta, \beta + 1), (\beta + 1, \beta + 1),$
$(\beta + 1, \beta + 2), (\alpha - 1, \beta + 2)$ is
an alternating cycle of length 8.

So the difference of the realizations lay in the subgraphs induced by $\bar{V}$, which subset contains 8 vertices. The subgraphs $Z[\bar{V}]$ and $Z'[\bar{V}]$ induced by $\bar{V}$ have the same (bipartite) degree sequence and they contain alternately the edges of the cycle. By Theorem 2.2 we know one of them can be transformed by swaps into the other one. Since the cycle contains four-four vertices from both classes, and there are at most 12 edges, therefore the canonical swap sequence (by Corollary 3.1) is at most 2×12 long therefore $\Theta = 24$ is an upper bound on the number of the necessary swaps. □

Clearly the same argument gives the following more general lemma.

Lemma 6.7. *For each natural number u there is a natural number Θ_u with the following property: assume that $F_Z = (m_{i,j})$ is an (α, β)-OK matrix and $m_{\alpha',\beta'} = 3$ where*

$$\left\|(\alpha, \beta); (\alpha', \beta')\right\| = u,$$

furthermore $F_{Z'} = (m'_{i,j})$ is an (α', β')-OK matrix such that

(1) $m'_{\alpha,\beta} = 3$,

(2) $m'_{i,j} = m_{i,j}$ *if* $i \neq j$, $i+1 \neq j$, *and* $(i,j) \neq (\alpha, \beta)$, (α', β').

Then at most Θ_u swaps transform Z into Z' (and along this F_Z is transformed into $F_{Z'}$).

Proof: The only difference is that here the symmetric difference of Z and Z' is a cycle of length at most $2 + 2u$ which alternates between Z and Z'. □

We also have the analogous general result for KO matrices.

Lemma 6.8. *For each natural number u there is a natural number Θ'_u with the following property: assume that $F_Z = (m_{i,j})$ is an (β, α)-KO matrix and $m_{\beta',\alpha'} = 0$ where*

$$\left\| (\beta, \alpha); (\beta', \alpha') \right\| = u,$$

furthermore $F_{Z'} = \left(m'_{i,j}\right)$ is an (β', α')-KO matrix such that

(1) $m'_{\beta,\alpha} = 0$,

(2) $m'_{i,j} = m_{i,j}$ *if* $i \neq j$, $i+1 \neq j$, *and* $(i,j) \neq (\beta, \alpha)$, (β', α').

Then at most Θ'_u swaps transform Z into Z' (and F_Z is transformed into $F_{Z'}$).

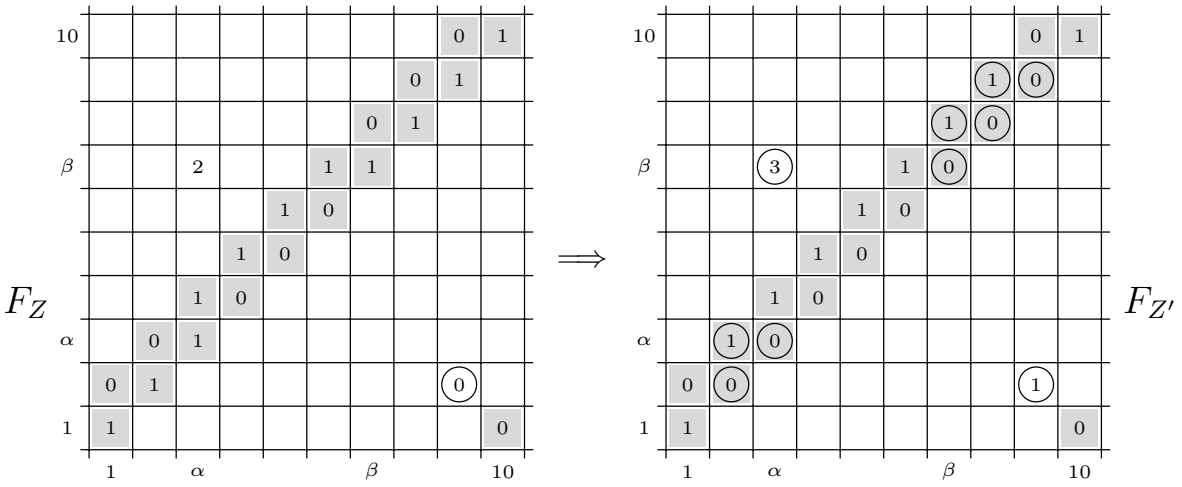
Proof: The proof is very similar to the proof of Lemma 6.7 which is left to the diligent reader. □

Lemma 6.9. *Assume that $F_Z = (m_{i,j})$ is (α, β)-OK matrix and $m_{\beta+2,\alpha-1} = 0$. Assume that $F_{Z'} = (m'_{i,j})$ is a $(\beta + 2, \alpha - 1)$-KO matrix such that*

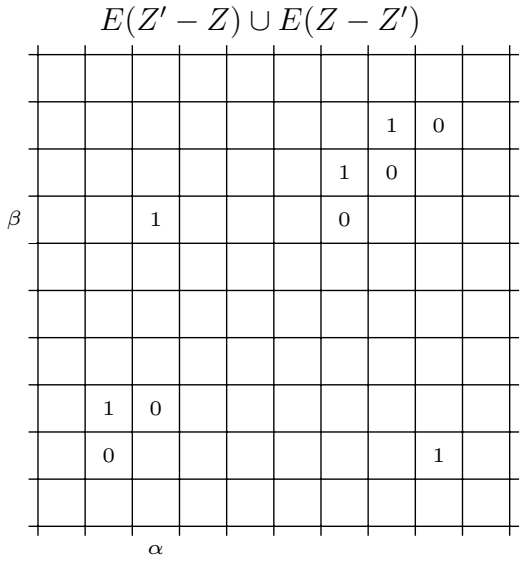
(1) $m'_{\alpha,\beta} = 3$,

(2) $m'_{i,j} = m_{i,j}$ *if* $i \neq j$, $i+1 \neq j$, *and* $(i,j) \neq (\alpha, l)$, $(\beta + 2, \alpha - 1)$.

Then there exists a natural number Ω such that one can transform Z into Z' by at most Ω swaps (and F_Z goes into $F_{Z'}$).



Proof: It is enough to observe that the symmetric difference of Z and Z' is a single alternating cycle. Indeed, in the next figure values 1 indicate edges in $E(Z' - Z)$ and values 0 indicate edges in $E(Z - Z')$.



Therefore
$(\alpha - 1, \alpha - 1), (\beta + 2, \alpha - 1),$
$(\beta + 2, \beta + 2), (\beta + 1, \beta + 2),$
$(\beta + 1, \beta + 1)(\beta, \beta + 1),$
$(\beta, \beta), (\alpha, \beta),$
$(\alpha, \alpha), (\alpha - 1, \alpha)$
is an alternating cycle of length 10.

The proof goes like the proof of Lemma 6.6: The difference of the realizations lies in the subgraphs induced by $\bar{V}$, which subset contains 10 vertices. The subgraphs $Z[\bar{V}]$ and $Z'[\bar{V}]$ induced by $\bar{V}$ have the same (bipartite) degree sequence and they contain alternately the edges of the cycle. By Theorem 2.2 we know one can be transformed by swaps into the other one. Since the cycle contains five vertices from both classes, and there are at most 20 edges, the number of the necessary swaps (by Corollary 3.1) is at most 2×20 therefore there exists a constant upper bound $\Omega \leqslant 40$ on the number of the necessary swaps. □

Lemma 6.10. *For each natural number u there is a natural number Ω_u with the following property: assume that $F_Z = (m_{i,j})$ is (α, β)-OK and $m_{\beta',\alpha'} = 0$ where*

$$\big\|(\alpha, \beta); (\alpha', \beta')\big\| = u,$$

and $F_{Z'} = (m'_{i,j})$ *is a* (β', α')*-KO matrix such that*

(1) $m'_{\alpha,\beta} = 3$,

(2) $m'_{i,j} = m_{i,j}$ *if* $i \neq j$, $i+1 \neq j$, *and* $(i,j) \neq (\alpha, \ell)$, (β', α').

Then at most Ω_u *swaps transform* Z *into* Z' *(and* F_Z *into* $F_{Z'}$*).*

Proof: Similar to Lemma 6.7. □

Now using our friendly path we are going to define a sequence of OK- and KO-matrices, such that we can achieve the required edge changes in G obtaining G' along this sequence, using operations described in the previous Lemmas. At first we define a new sequence $A'_1, \ldots, A'_\Lambda$ from $A_1, \ldots, A_\Lambda$ in the following way:

$$A'_i = \begin{cases} A_i, & \text{if } F_G(A_i) = 0, \\ \text{Cousin}(A_i), & \text{if } F_G(A_i) = 3, \end{cases} \tag{6.1}$$

where Cousin(A) denotes one of the cousins of A. If there are more than one positions of the same type among the corresponding positions, then we choose the lexicographically-least one. We will use the following notation: the *mirror image* of the position (α, β) to the main-diagonal is $\text{Mirror}(\alpha, \beta) = (\beta, \alpha)$.

Observation 6.11. *By definitions,*

(i) *if* $F_G(A_i) = F_G(A_{i+1})$ *then* $\left\|A'_i, A'_{i+1}\right\| \leqslant 3$,

(ii) *if* $F_G(A_i) \neq F_G(A_{i+1})$ *then* $\left\|\text{Mirror}(A'_i), A'_{i+1}\right\| \leqslant 3$.

Definition 6.12. We define the matrix sequence $F_G = L_0, L_1, \ldots, L_\Lambda, L_{\Lambda+1} = F_{G'}$ and the corresponding realizations $Z_1, \ldots, Z_\Lambda$, where $L_i = F_{Z_i}$ for each i as follows:

The matrix L_i $(i = 1, \ldots, \Lambda)$ is defined from the matrix L_{i-1} by the formulae:

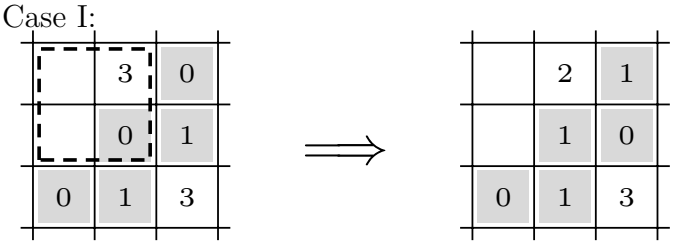
$$L_i = \begin{cases} \text{the } (A'_i)\text{-OK matrix}, & \text{if } L_{i-1}(A_i) = 3, \\ \text{the } (A'_i)\text{-KO matrix}, & \text{if } L_{i-1}(A_i) = 0. \end{cases}$$

Here all positions (u, v) which are NOT determined by the definitions of the OK- and KO-matrices satisfy $L_i(u, v) = L_0(u, v)$. □

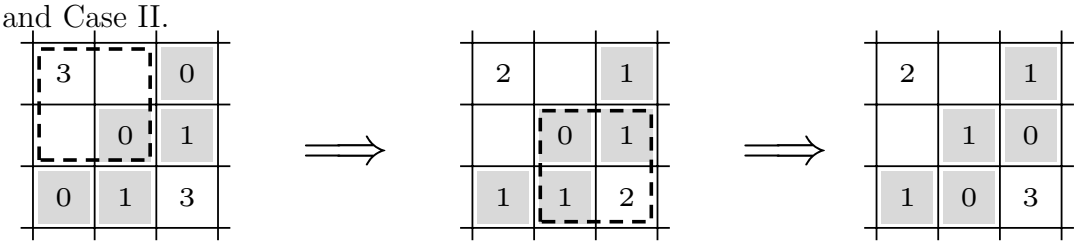
It is quite clear that $(\Lambda - 1)$ consecutive applications of (the appropriate) Lemmas 6.6 - 6.10 will take care the definition of the required swap sub-sequences between L_1 and L_Λ. However, the swap-sequence transforming L_0 into L_1 furthermore the one transforming L_Λ into $L_{\Lambda+1}$ require special considerations:

- If $L_0(A_1) = 3$ then there are two possibilities - depending on the position of the Cousin(A_1). (The squares denoted with dashed lines contain the possible positions of friendly cousins.)

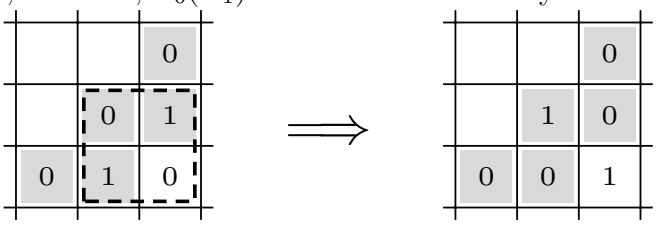
Case I:



and Case II.



- If, however, $L_0(A_1) = 0$ then there is only one case:



The connecting swap-sequence from the matrix L_Λ to $L_{\Lambda+1}$ (which is $F_{G'}$) can be defined analogously to the previous one. This completes the definition of the canonical path $\Gamma(X, Y, s)$.

Next we will analyze the behavior of the current matrices $\widehat{M}(G, G', Z)$ along these sub-sequences. At first we consider those Z's which correspond to matrices L_i.

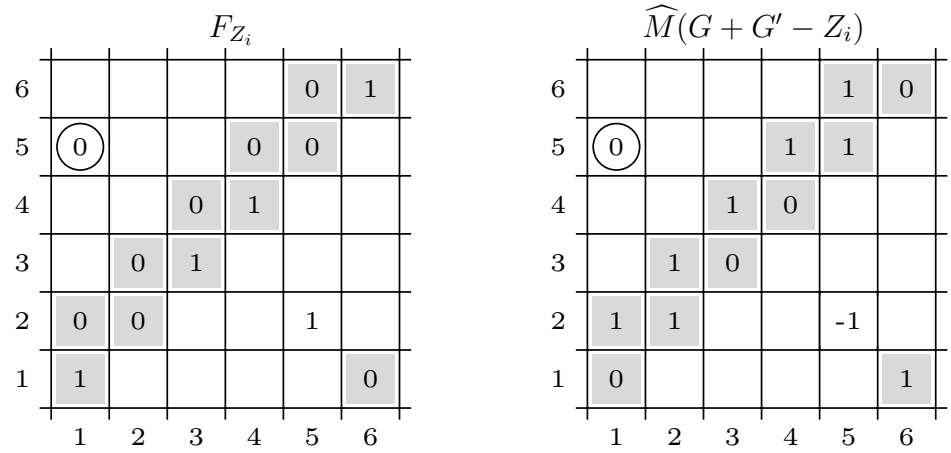
Let M be an integer matrix and let M' be a 2×2 submatrix of it. If we add 1's to the values of the positions of one diagonal in M' and -1's to the values of the positions of the other diagonal, then the acquired matrix has the same row and column sums as M had. Such an operation is called a **switch**. When our matrix M is the adjacency matrix of a degree sequence realization, then any swap clearly corresponds to a switch of that matrix. We say that the two matrices are in **switch-distance** 1 from each other. It is clear that bounded switch-distance between two matrices also means bounded Hamming distance between them (as it was required in (F)(d)).

The following lemma is an auxiliary result, which help us to handle the numbers of different paths (in our canonical path system) which cover the same edge. It has no role in the definition of our path system, but it helps to show that this path system obeys the rules outlined in (A) – (F) is Section 5.

Lemma 6.13. *For $i = 1, \dots, \Lambda$ there exist realizations $G_1, \dots, G_\Lambda$ in $V(\mathbb{G})$ for which M_{G_i} is in switch-distance 1 from the matrix $\widehat{M}(G + G' - Z_i)$ for $i = 1, \dots, \Lambda$.*

Proof: We show here the statement for such an L_i where $L_i(A_i) = 0$ therefore L_i itself is an (A_i')-KO matrix, and where - by definition - $A_i = A_i'$ (the other case is similar).

Due to the definitions A_i originally is not an edge either in G or in G'. It belongs to the friendly path, therefore we also know that $F_G(\text{Cousin}(A_i)) = F_{G'}(\text{Cousin}(A_i)) = 0$ hold. In L_i this value is 1, so A_i is an edge in Z_i. Therefore L_i which is $= F_{Z_i}$ looks like the matrix to the left in the following figure (the circled element is the cousin of A_i). The corresponding $\widehat{M}(G + G' - Z_i)$ is shown on the right hand side:



It is clear that adding 1 to the values of the positions A_i and $\text{Cousin}(A_i)$ of $\widehat{M}(G+G'-Z_i)$ and subtracting 1 from the other two corners of the spanned submatrix constitutes the required switch. □

(In the figures above A_i is $(5, 2)$. Here one can also recall that outside our $\ell \times \ell$ submatrix every entry is 0 or 1 and after the switch the same applies inside the submatrix. Therefore, due to the row- and column-sum conditions, the acquired matrix is a realization indeed.)

Lemma 6.14. *The realization G can be transformed into the realization G' through realizations Z_i ($i = 1, \ldots, \Lambda$) in such a way that the lengths of the swap sub-sequences leading from each Z_i to Z_{i+1} (where $0 = 1, \ldots, \Lambda$) can be bounded from above by the absolute constant $\max\{\Theta_3, \Theta'_3, \Omega_3\}$. In this process, each arisen matrix $\widehat{M}(G + G' - Z_i)$ is within a constant switch-distance from some vertex in $V(\mathbb{G})$ (that is some realization of the bipartite degree sequence).*

Proof: By Observation 6.11 for each i the positions A'_i and A'_{i+1} or $\text{Mirror}(A'_i)$ and A'_{i+1} are at most distance 3. Therefore for each i (where $i = 2, \ldots, \Lambda$) the corresponding process chosen among Lemma 6.7, Lemma 6.8 and Lemma 6.10 will describe the desired swap sub-sequences. The length of any such swap-subsequence is bounded from above by $\max\{\Theta_3, \Theta'_3, \Omega_3\}$.

Furthermore when in the process the current realization Z_i corresponds to an $F_{Z_i} = L_i$, then Lemma 6.13 applies, and matrix $\widehat{M}(G + G' - Z_i)$ has switch-distance 1 from the adjacency matrix of some realization $\in V(\mathbb{G})$.

Let now Z be a realization in the process, say, on the path between the matrices L_i and L_{i+1}: then $\widehat{M}(G+G'-Z_i)$ can be transformed through swaps into $\widehat{M}(G+G'-Z_{i+1})$ (assume, this end is the closer one to Z). As we know all swaps are specialized switches, and they keep the row and column sums. Combining this with the previous paragraph, we

have for every Z that $\widehat{M}(G+G'-Z)$ is at most $\left\lceil \frac{1}{2}\max\{\Theta_3,\Theta_3',\Omega_3\}\right\rceil+1$ switch distance from some realization $\in V(\mathbb{G})$. □

Key problem

One can say that we are very close to proving the rapidly mixing property of our Markov process on all bipartite degree sequences: we should prove, that in the case when there exists a friendly path from G to G' then for each intermediate Z the matrix $\widehat{M}(X+Y-Z)$ is in a constant distance from some realization $\in V(\mathbb{G})$. If we can manage this then we must handle the cases when there are no friendly paths. It is somewhat surprising that this second requirement can be satisfied successfully (as it will be shown in Subsection 6.2).

However, we cannot manage to prove the first requirement. The problem is the following: we can try to repeat the proof of Lemma 6.14, but, unfortunately, it is not true anymore that for each graph Z, corresponding to a particular matrix L_i, the matrix $\widehat{M}(X+Y-Z)$ is also in distance 1 from some realization in $V(\mathbb{G})$.

In the realizations G and G' all chords have the same types, but this is not the case for realizations X and Y. The edges in $E(X-Y)\cup E(Y-X)$ belong to only one of them. Therefore if a swap turns an entry to 2 in $\widehat{M}(G+G'-Z)$ then this entry originally was 1: the edge belonged to G and G' and Z as well. Therefore its cousin bears the entry 1 (also belonged to G and G' and Z as well). So this entry was appropriate to perform a switch to turn the matrix under investigation into the adjacency matrix of a realization. However, if the cousin entry is 0 in $\widehat{M}(X+Y-Z)$ (this edge belongs only to one of realizations X and Y, say, it belongs to X only), then the required switch cannot be performed. (The value -1 can cause a similar problem and can be handled similarly as this case.)

A good solution for this particular problem would probably end up in a complete proof of the rapidly mixing property.

The following observation is enough to handle the switch-distance problem for $\widehat{M}(X+Y-Z)$ in half-regular bipartite degree sequences. Recall, a bipartite degree sequence $(\mathbf{a},\mathbf{b})$ is *half-regular* if in $\mathbf{a}$ all degrees are the same, while the entries in $\mathbf{b}$ can be anything.

Lemma 6.15. *Assume that our bipartite degree sequence* $(\mathbf{a},\mathbf{b})$ *is half-regular and the matrix* F_G *under investigation contains a friendly path. Then the statement of Lemma* 6.14 *applies for the matrices* $\widehat{M}(X+Y-Z_i)$ *as well.*

Proof: We follow the proof of Lemma 6.14. To do so the only requirement is to show (somewhat loosely) that the matrices $\widehat{M}(X+Y-L_i)$ are in a constant switch-distance from the adjacency matrix of some realizations. As we know any of these matrices contains exactly one entry of value different from 1 and 0. So consider a particular L_i and assume that this "extra" value in this case is a 2. If the switch, described in the proof of Lemma 6.13, is also a possible switch in $\widehat{M}(X+Y-Z)$ then we are done. If this not the case then the entry (with value 1 in matrix $\widehat{M}(G+G'-L_i)$) has value 0 in $\widehat{M}(X+Y-L_i)$. (In this case, as we discussed it previously, the corresponding edge is missing from Y.)

Let this corresponding edge be (u, v), then this entry in $\widehat{M}(X + Y - L_i)$ is 0. Since the column sums are fixed in these matrices, they are the same (and equal to entries in **a**).

Now vertex v has degree at least 2 (it is a vertex on cycle C and it also end point of at least one chord of C in X). Therefore the column v contains some 1s. One of them is (w, v) (this w cannot be the row of the 2, since the entry there is 0 due that it belongs to the originally intended switch). Now by the pigeonhole principle (since all row sums are the same) there is a column z such that $\widehat{M}(w, z) = 0$ and $\widehat{M}(u, z) = 1$. Therefore the $u, w; v, z$ switch (actually this is a swap) will change $\widehat{M}(u, v)$ into 1, and now the original switch finishes the job. The matrix $\widehat{M}(X + Y - L_i)$ is in switch-distance at most 2 from the adjacency matrix of some realization. □

6.2 The case that no friendly path exists

In the previous subsection we discussed the situation when – processing one by one the cycles in the canonical decomposition of the symmetric difference – the cycle under investigation possesses a friendly path. All definitions, statements, reasonings were valid for any arbitrary bipartite degree sequence – except the situation described in the Key Problem and in Lemma 6.15 where we have to use the half-regularity condition.

Here we discuss the case where there exists no friendly path in the cycle under investigation. Nothing that we define here, state here or prove here requires the half-regularity condition. So here our general assumptions are: we have realizations G and G' of the same (arbitrary) bipartite degree sequence, where the symmetric difference of the two edge sets forms exactly one cycle, which, in turn does not possesses a friendly path.

Our plan is this: at first we show that the non-existence of the friendly paths yields a strong structural property of the matrix F_G. Using this property we can divide our problem into two smaller ones, where one of the smaller matrices possesses a suitable friendly path. So we can solve our original problem in a recursive manner.

This recursive approach must be carried out with caution: a careless "greedy" algorithm can increase the switch-distances very fast. We will deal with this problem using a simple "fine tuning" (which is described at the end of this subsection).

We start with some further notions and notations.

Definition 6.16. In an $\ell \times \ell$ matrix the *sequence* of positions $(i + 1, i - 1), (i + 2, i - 2), \ldots, (i + \lfloor \ell/2 \rfloor - 1, i - \lfloor \ell/2 \rfloor + 1)$ form the ith **down-line** of the matrix. (The arithmetic operations are thought to be considered modulo ℓ, that is, for example, $1 - 3 = \ell - 2$. Therefore if the down-line reach the edge of the matrix at position, say, (ℓ, k) then the next position is $(1, k - 1)$. Similarly, if the position on the edge is $(k, 1)$ then the next position is $(k + 1, \ell)$. If $2i > \ell$ then the first case applies, in case of $2i < \ell$ the second case applies. Finally if, by chance, $2i = \ell$ then the positions in questions are $(\ell, 1)$ and $(1, \ell)$. Analogously the *sequence* of positions $(i-1, i+1), (i-2, i+2), \ldots, (i - \lceil \ell/2 \rceil + 1, i + \lceil \ell/2 \rceil - 1)$ form the ith **up-line** of the matrix. (Let us mention that in case of even ℓ the length of the up-lines and the down-lines are equal. However, in case of odd ℓ the up-lines are longer by one position.)

Since the lines are sequences therefore by definition they have orientations along which the algorithm will traverse them. Also, by definitions, in case of even ℓ, the ith down-line equals the jth up-line (for some j) as sets. However, as sequences, they are of course different.

Definition 6.17. A set T of positions of an $\ell \times \ell$ matrix is called **rook-connected** if a chess rook, staying inside T, can visit all elements of T. Here the chess rook is allowed to wrap around cyclically on the rows and columns (that is the rook is moving on a torus). We use the expression **king-connected** analogously.

The following lemma is a well-known version of the classical Steinhaus lemma (see [13]).

Lemma 6.18. *Assume that the off-diagonal positions of an $\ell \times \ell$ matrix are arbitrarily colored white and black. Then either the rook has a white path which starts at distance 1 from the small-diagonal and ends at distance 1 from the main-diagonal, and avoids both diagonals, or there is a king-connected set T of black positions which intersects all rook's paths from the main-diagonal to the small-diagonal.* □

We use the previous result without proof. The set T of black positions, which was identified in the previous lemma, will be called a *Steinhaus set*. So a Steinhaus set is a king-connected set of positions which intersects all rook's paths from the main-diagonal to the small-diagonal.

Definition 6.19. The **cousin-set** $\mathfrak{C}(u, v)$ is the set of the off diagonal cousins of the position (u, v). If T is a set of positions, then the cousin set $\mathfrak{C}(T)$ is defined as $\bigcup\{\mathfrak{C}(e) : e \in T\}$.

We will use Lemma 6.18 as follows: We color the off-diagonal positions of the matrix F_G with white and black: the friendly positions are white while the unfriendly ones are black. If the matrix does not posses any friendly path, then there exists no white rook's path from the small diagonal to the main diagonal (since that would be in fact a friendly path). Consequently, by Lemma 6.18, there exists an all-black Steinhaus set T (so it consists of unfriendly positions only). While the Steinhaus set T is (only) king-connected, but as we will show in Lemma 6.20 the derived cousin-set $\mathfrak{C}(T)$ is rook-connected, moreover the type of all positions in the cousin-set $\mathfrak{C}(T)$ are the same. Furthermore, by Lemma 6.21, $\mathfrak{C}(T)$ intersects all down-lines and up-lines. Finally we will put together these observations in Lemma 6.22 and Corollary 6.23 to find a smaller submatrix which possesses a friendly path.

Lemma 6.20. *Assume that T is a king-connected set in the matrix F_G. Then the cousin-set $\mathfrak{C}(T)$ is rook-connected. Moreover, if the positions in T are all unfriendly, then the type of all positions in $\mathfrak{C}(T)$ are the same, and all positions in T have the opposite type.*

Proof: Let P be a position in T. For each other position P' in T, which can be reached from P in one king step, the cousin sets $\mathfrak{C}(P)$ and $\mathfrak{C}(P')$ have at least one common position. Therefore the neighboring cousin sets are rook-connected, so $\mathfrak{C}(T)$ is rook-connected.

Assume now that the positions in T are all unfriendly. W.l.o.g. we may assume that a position P in T has type 0, then all positions in its cousin-set must have type 1. However, for each other position P' in T, which can be reached from P in one king step, the cousin sets $\mathfrak{C}(P)$ and $\mathfrak{C}(P)$ have at least one common position, therefore all types in those two cousin sets must be the same (1), therefore both positions P and P' have the same type (0) as well. □

Lemma 6.21. *Let T be a Steinhaus set in F_G, then its cousin-set $\mathfrak{C}(T)$ intersects all down-lines and up-lines.*

Proof: Actually we prove more: namely that any king-path U from the main-diagonal to the small-diagonal intersects the cousin-set $\mathfrak{C}(T)$.

Indeed, by Lemma 6.20, the set $\mathfrak{C}(U)$ is rook-connected, so $\mathfrak{C}(U)$ intersects the Steinhaus set T. However, if $P \in \mathfrak{C}(U) \cap T$, then $P \in \mathfrak{C}(P')$ for some $P' \in U$. But then $P' \in \mathfrak{C}(P)$, so $\mathfrak{C}(T)$ intersects U.

Finally it is clear, that every down- and up-line forms a required king-path. □

Lemma 6.22. *We assume that in the matrix F_G there is no friendly path. Then for each i ($i = 1, \ldots, \ell$) there exists a $\mathbf{t} \in \{0, 1\}$ and a pair of indices $j, j' \in \{1, \ldots \lceil \ell/2 \rceil - 1\}$ such that one of the following holds:*

- *entries $(i+1, i-1), \ldots, (i+j, i-j)$ and $(i-j', i+j')$ have the same type $\mathbf{t}$, furthermore the entries $(i-1, i+1), \ldots, (i-j'+1, i+j'-1)$ have the type $1-\mathbf{t}$, and all entries belong to a down- or up-line;*
- *entries $(i-1, i+1), \ldots, (i-j, i+j)$ and $(i+j', i-j')$ have the same type $\mathbf{t}$, furthermore the entries $(i+1, i-1), \ldots, (i+j'-1, i-j'+1)$ have the type $1-\mathbf{t}$, and all entries belong to a down- or up-line.*

Proof: Color the friendly positions white, and the unfriendly positions black. Since a white rook path from the small diagonal to the main diagonal would be a friendly path, by Lemma 6.18 there is a Steinhaus set T containing only black, i.e. unfriendly positions. Now, by Lemma 6.20, the type of all positions of the cousin-set $\mathfrak{C}(T)$ are the same. Moreover, by Lemma 6.21, the cousin-set $\mathfrak{C}(T)$ of T intersects all down-lines and up-lines.

Assume for a contradiction that there is no such j for a particular i. W.L.O.G. we may assume, that $F_G(i+1, i-1) = 0$. Then, by the assumption, $F_G(i-1, i+1) = F_G(i-2, i+2) = 3$ must hold. Then, again by our assumption, $F_G(i+2, i-2) = 0$ must hold, etc. All entries along the down-line are 0, while all entries along the up-line must be 1. However, as we observed in the previous paragraph, both lines intersect the cousin-set $\mathfrak{C}(T)$ of the Steinhaus set T. But, by Lemma 6.20, all its entries have the same type. A contradiction. □

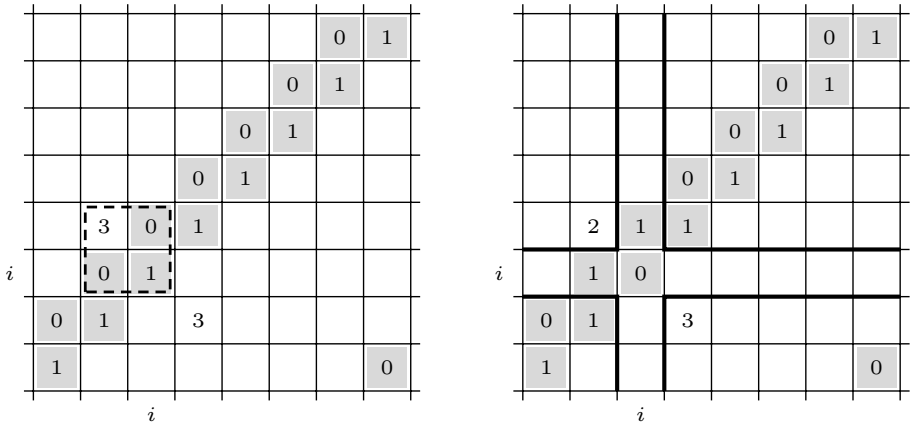
Corollary 6.23. *If conditions of Lemma 6.22 hold, and $j' \geqslant 2$ for a particular i, the submatrix spanned by $(i+j, i-j)$ and $(i-j, i+j)$ contains at least one friendly path. (Let us recall that the bottom-right position $(i+j, i-j)$ belongs to the small-diagonal by definition.)*

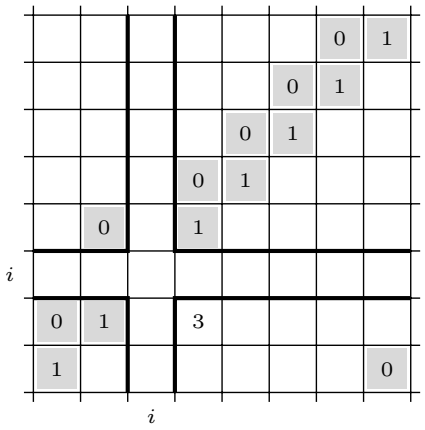
Proof: We argue by contradiction: assume that the submatrix does not contain a friendly path. Then - due to Lemma 6.18 - it contains a Steinhaus set T. Due to Lemma 6.20, in its cousin-set $\mathfrak{C}(T)$ - which intersects all down- and up-lines - all positions have the same type. But this contradicts to the fact, that in the ith down-line all positions have type $\mathbf{t}$, while in the ith up-line all positions have type $1-\mathbf{t}$. A contradiction, again. □

That finishes the preliminaries that are needed to describe our recursive algorithm, which is essentially a divide and conquer approach. Due to the previous fact here we should handle separately two possibilities: when $j'=1$ (and $j=1$ as well) and when $j'\geqslant 2$. For sake of simplicity we will assume that in our cycle the first condition described in Lemma 6.22 holds. We start with the

First possibility: assume that, for a particular i, we have $j'=1$. Then we also have $j=1$. We should take care of two cases:

Case 1: If both $F_G(i+1,i-1)$ and $F_G(i-1,i+1)=3$ (that is both chords belong to both G and G') then we are in an easily handleable situation: at first we swap the quartet $u_i,u_{i+1};v_i,v_{i+1}$. (The dashed square in our illustration. Here we use the matrix F_G.) Denote the resulting realization by Z_1. In Z_1, the entries $(i-1,i),(i,i)$ and $(i,i+1)$ have the required types. However, entry $F_{Z_1}(i-1,i+1)=2$ therefore during the procedure we should take care to change the entry $(i-1,i+1)$ of Z_1 back to its original value. Indeed, during the procedure we should guarantee that the matrix $\widehat{M}(X+Y-Z)$ contains at most Ω_2 many 2-s and -1-s, i.e. the corresponding F-matrix contains at most Ω_2 many 1-s and 2-s (see (F)(d) from Section 5.3). Since we want to apply our method recursively, therefore during the procedure we should take care to change it back to its original value otherwise the 2-s could accumulate in the F-matrices. We will handle this problem during the "fine tuning".



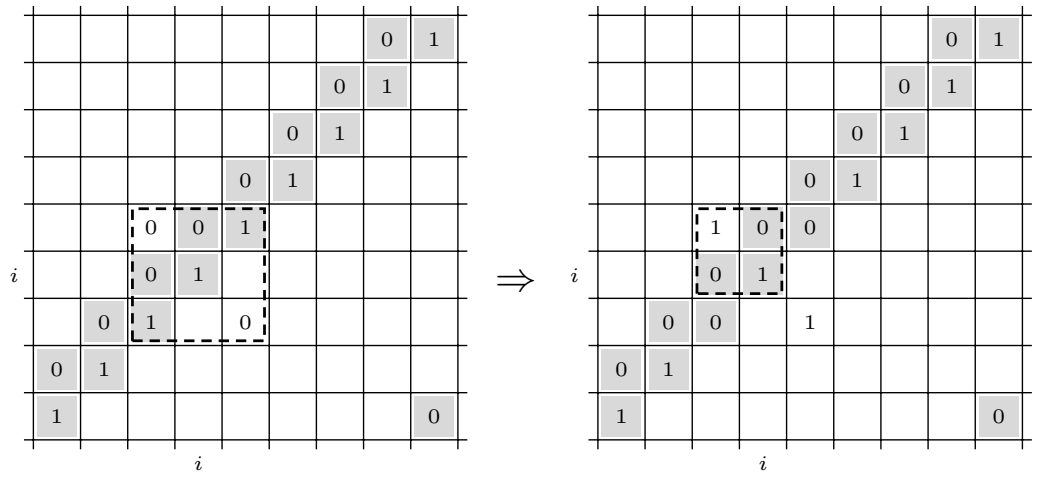


The remaining subproblem, indicated by thick black lines, fortunately is already in the required form. Indeed: its main-diagonal contains only 1s, while its small-diagonal is full with 0s. (We have to keep it in our mind that the shown matrix of the remaining smaller subproblem is F_Z where the element of the main- and small-diagonals came from Z.)

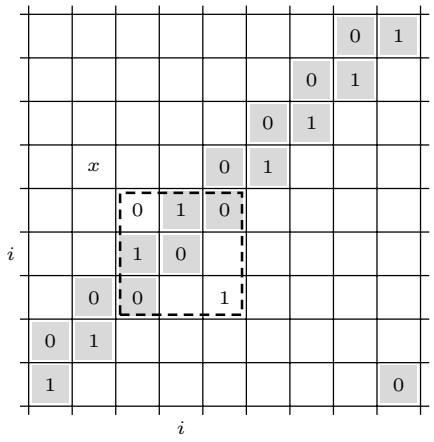
Denote the alternating cycle of this smaller problem by C'. The (recursive) subproblem C' may contain a friendly path which will process it completely in one step, and will switch the value of $F_G(i-1, i+1)$ automatically back to 1. If, however, it does not contain a friendly path, then the recursive procedure can use any down- and up-lines, including $(i-1, i+1)$ (see Lemma 6.21), therefore we can take care that this switch-back will happen in the next recursion.

It is important to recognize that matrices $\widehat{M}(G+G'-Z)$ and $\widehat{M}(X+Y-Z)$ may contain 2 at the position $(i-1, i+1)$. Fortunately this "problematic" entry will be present only along one recursive step. Furthermore this entry will increase the switch-distance of the current $\widehat{M}$ by at most one: the positions $(i-1, i), (i, i)$ and $(i, i+1)$ (outside of our subproblem), provide a suitable switch to handle the entry 2 at position $(i-1, i+1)$.

Case 2: Now we have $F_G(i+1, i-1) = 0$ and $F_G(i-1, i+1) = 0$. Here we perform two swaps, the places of the swaps are denoted (shown below) with dashed squares:



If $\widehat{M}(X+Y-Z)(i+1, i-1) = -1$ holds, then it increases the switch-distance of the current $\widehat{M}$ by at most one (since it can be directly back-swapped). The result of the second swap (after which the previous problem is just solved automatically), together with our further strategy is shown below:



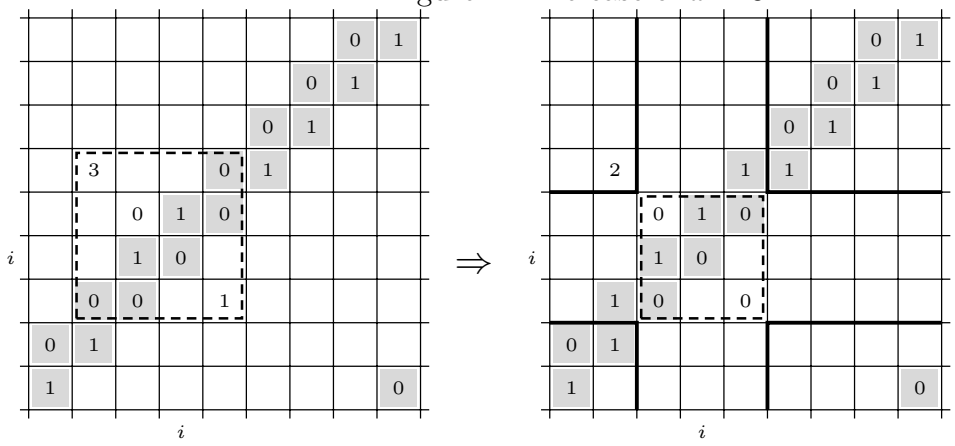
Here we distinguish between two cases, according to the value

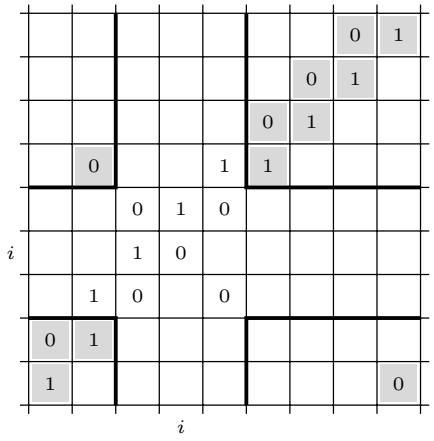
$$F_Z(i-2, i+2) = x.$$

This value can be $x = 3$ or $x = 0$.

In the case of $x = 3$ we perform one more swap, which results in a subproblem with a friendly path (the swap shown on the left side of Figure 4, while the right hand side indicates the two new subproblems):

Figure 4: The case of $x = 3$





Here we have redrawn the RHS of Figure 4 to show that one subproblem is already solved: the cross shaped midsection is in the required state: the main-diagonal (within the midsection) contains all 0's, while each entry in the small-diagonal is 1 and the off-diagonal positions are in their original states. Here is nothing more to do.

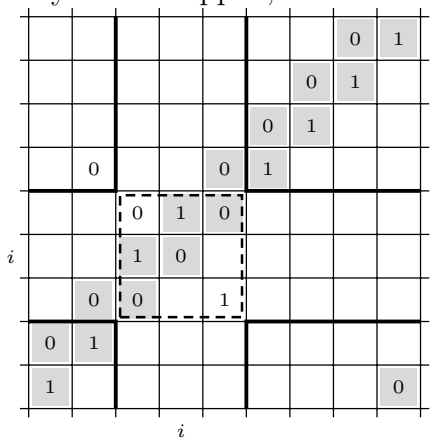
The second subproblem (indicated with the thick black lines, the four pieces fit together to a square matrix) is in the right form for further processing. The position $(2, 6)$ changed

into 0 since we described the subproblem in the language of (the now smaller) F_Z: the positions in the small-diagonal depend on the edges of Z only.

We will process this second subproblem along the up-line, containing position $(i-2, i+2)$, so the only currently improper entry will have the right value at the end of the next recursion step (that is it will be swapped back to its original value). (Here we can argue the same way as in Case 1.)

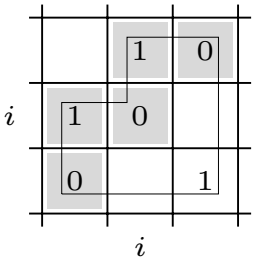
As it happened before $\widehat{M}(X+Y-Z)$ may contain 2 at the position $(i-2, i+2)$. Again this increases the switch-distance by at most one, since the positions $(i-2, i-1)$, $(i+1, i-1)$ and $(i+1, i+2)$ are not in our subproblem.

Finally it can happen, that $x = 0$. Then we can define the following subproblem:

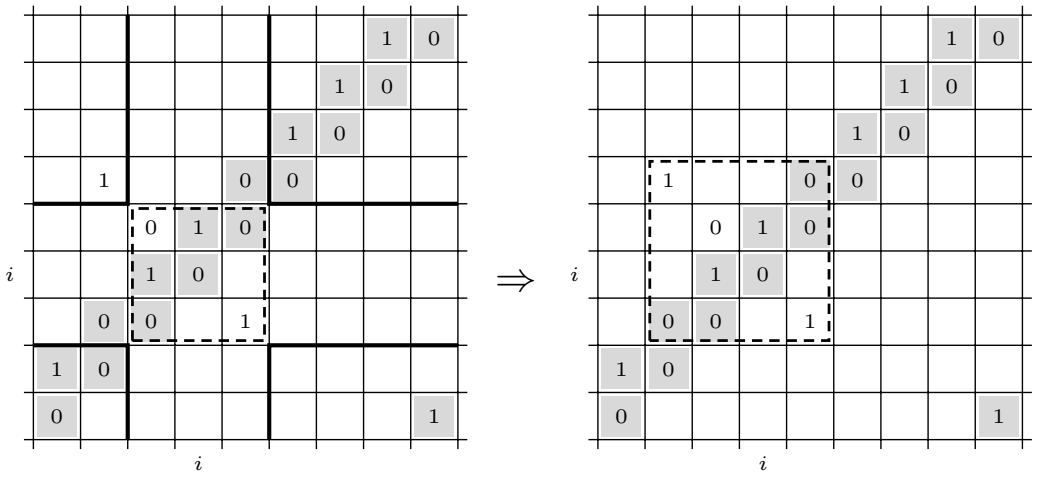


This figure shows the new subproblem (indicating with thick black lines) is in the right form (for further processing) again. We will process the subproblem along the up-line, containing position $(i-2, i+2)$ (so the only currently improper entry will have the right value at the end of the next recursion step).

Here, again, we may confront the fact, that $\widehat{M}(X+Y-Z)(i+1, i-1) = -1$. Then we should consider the alternating cycle shown in the figure. All elements of the cycle, except $(i+1, i-1)$, are in the main- and small-diagonal, therefore along this cycle we can swap that entry into range within a small number (say δ) of steps. This will increase the switch-distance of $\widehat{M}$ by at most δ.

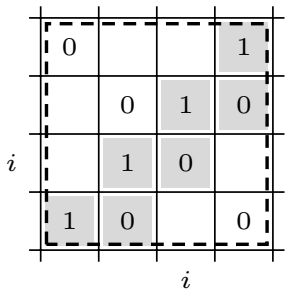


We run the first recursion on the subproblem along the ith up-line, therefore the sub-subproblem with friendly path will contain the position $(i-2, i+2)$. Therefore when we finish the first recursion, our matrix F_G will be in the following form: (the figure on the left):



We have seen how one can handle the switch-distance of our matrix, if position $(i+1, i-1)$ is problematic with $\widehat{M}(i+1, i-1) = -1$ but position $(i-2, i+2)$ is correct. On the other hand if $\widehat{M}(X+Y-Z)(i-2, i+2) = -1$ then the swap on the positions $(i-2, i-1), (i+1, i+2); (i+1, i-1), (i-2, i+2)$ changes both $(i+1, i-1)$ and $(i-2, i+2)$ into 0. For $(i+1, i-1)$ that was the original type - so it cannot be wrong in $\widehat{M}$.

After that we perform the swap on the positions $(i-2, i-1), (i+1, i+2); (i+1, i-1), (i-2, i+2)$ (these are the corners of the dashed square in the figure on the upper right). The result is shown to the right:



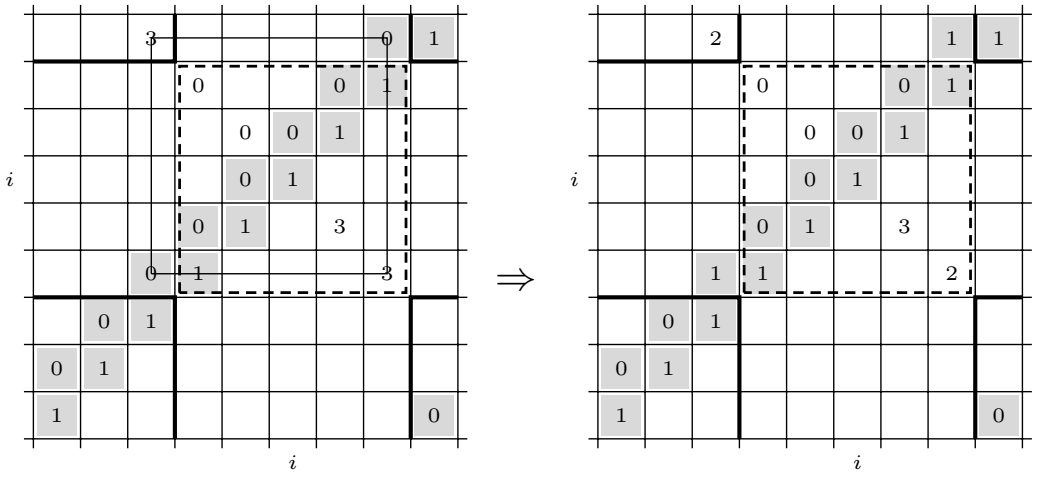
This completes our handling on the First Possibility, that is when for our i we have the value $j' = 1$. Now we turn to the other (and probably more common) configuration:

Second Possibility: We have $j' \geqslant 2$. Unfortunately, the situation can be more complicated in this case due to the possible switch-distances of $\widehat{M}$. We overcome this problem by showing at first the general structure of the process, and later we give the necessary fine-tuning to ensure the bounded switch-distance. (Recall again, that the bounded switch distance is necessary to have a good upper bound on the number of different matrices $\widehat{M}$ appearing along the algorithm.)

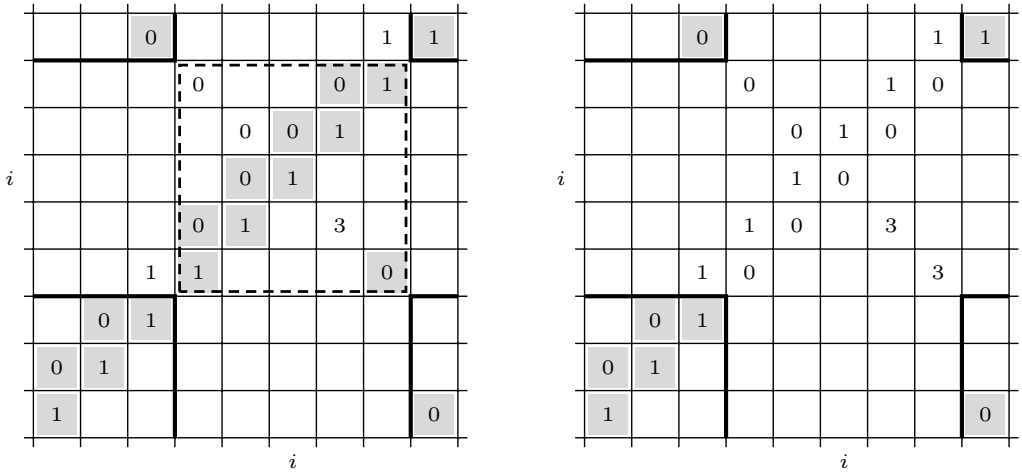
In our current alternating cycle (lying in the symmetric difference of G and G') there is no friendly path, therefore there is a Steinhaus set T in F_G. Now fix a particular i and assume that the j' corresponding to this i is $\geqslant 2$. We should distinguish between two cases: where the down-line starts with the value $t = 3$ or with $t = 0$.

Case 1: $t = 3$ The first figure below shows the structure of matrix F_G. The dashed square is the first subproblem to deal with, while the thick black lines indicate the second

subproblem. However, before we start the processing the subproblems, we have to perform a swap. The corners of the thin black square shows the positions of the swap.



After this swap (see the figure above, right), the first subproblem (indicated by the dashed square) is in the right form. Indeed, the left figure below shows the two separate subproblems.

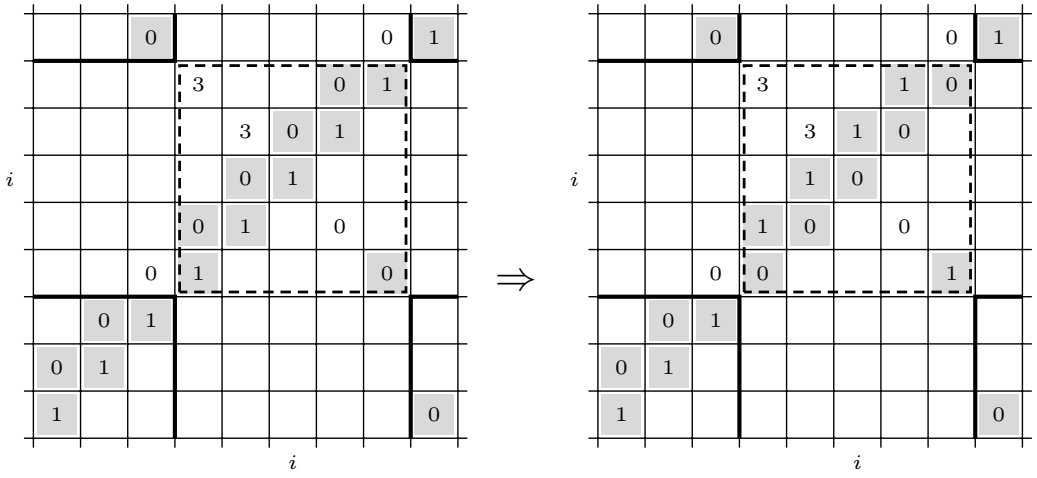


Finishing the first subproblem, we have the F_Z matrix (above, right). As it can be seen, after the first phase, all entries in the midsection are in their required types: the small-diagonal consists of 1s (including position $(i+2, i-2)$ which in that way is back to its original type), while the main-diagonal consists of only 0s.

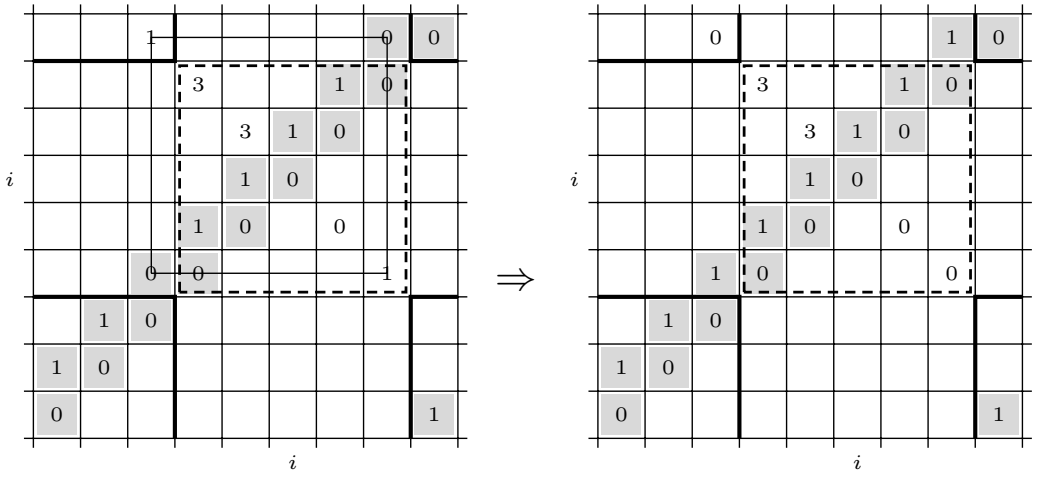
The second subproblem (indicated by the thick black lines) in the right form now to process (including position $(i-3, i+3)$ which is sitting on the small-diagonal).
After completing the solution of the black subproblem, all entries in the matrix will be in exactly the required type. We start processing the black subproblem on the ith up-line, therefore the actual types of positions $(i-3, i+3)$ and $(i+2, i-2)$ can be described as follows: Position $(i+2, i-2)$ has opposite type after the very first swap, then while processing the dashed subproblem it may change between 0 and 1. Finishing the dashed subproblem, it will be in the same type as it starts.

Position $(i-3, i+3)$ will be in type 0 all the way in the dashed phase, while within the black phase it will change between 0 and 1. At the end, as we already mentioned, is 1.

Case 2: $t = 0$ The first figure below shows the structure of matrix F_G. The dashed square is the first subproblem to deal with, while the thick black lines indicate the second subproblem. They can process without any preprocessing.



At the end everything will be in the right type, except the four positions, showed by the thin black square (below, left side). We can finish the process with that swap.



While the overall structure of our plan is clear, we may meet problems along this procedure. The reason is that we must be able to control the switch-distance of our $\widehat{M}(X+Y-Z)$ (we will use here simply $\widehat{M}$) from the adjacency matrix of some realization. There are two neuralgic points: both the positions $(i+j, i-j)$ and $(i-j', i+j')$ may contain -1, or both may contain 2. When we start a new subproblem, then their types always provide a suitable switch for the control (as it was seen before). However, when we proceed along our subproblem, then it can happen that one of the problematic positions changes its value, while the other does not. But in this case the switch which was previously

available is not useable anymore. Next we describe how we can fine tune our procedure to avoid this trap.

As we know the first subproblem contains a friendly path (by Corollary 6.23), and for easier reference let call its problematic position P_1. We also know that second subproblem contains a problematic position, P_2, and probably we have to divide this subproblem into two smaller ones. If so, then the first of them becomes the new second subproblem, which contains P_2 and possesses a friendly path, while the third subproblem contains another problematic position, P_3.

Fine tuning:

1. We begin our swap sequence along the first subproblem but we stop just before we face the swap which changes the value of P_1.

2. Next we continue with the swap sequence of the second problem and we stop before we should perform a swap on P_2.

3. Now we finish the swap sequence of the first subproblem.

4. After that we focus on the second subproblem. Dealing effectively with this, we need to prepare the third subproblem similarly as we did with the second one, when we were working on the first one. Therefore we begin the swap sequence of the third subproblem but we stop it before the first swap would be carried out on P_3.

5. And if now we just rename our two active subproblems as first and second subproblem, we are back to a situation, which is equivalent to the beginning of the third stage.

Along this algorithm, at each point we have two "active" subproblems. When a subproblem has a friendly path, then along this path we define the necessary swap sequence (as described in Subsection 6.1) and we have an upper bound on its length. When the subproblem is without a friendly path, then we divide it into two, and one (or both) of them have a friendly path, etc. The sum of the sizes of the subproblems is at most the size of the original cycle. Finally we put together the final swap sequence from these swap sequences and some short sequences we get from the (sometimes) necessary preprocessing. Since we have bounded switch distances all along (one or two at preprocessing stages, and those given in Subsection 6.1), therefore all together we have a good control of the overall number of used $\widehat{M}$'s. □

Acknowledgement

The authors would like to thank to the anonymous referee, whose comments and suggestions improved the manuscript significantly. We are most grateful to Catherine Greenhill for her tremendous help to prepare this manuscript.

References

[1] Bollobás, B.: A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European J. Comb.* **1** (1980), 311–316.

[2] Cooper, C. - Dyer, M. - Greenhill, C.: Sampling regular graphs and a peer-to-peer network, *Comb. Prob. Comp.* **16** (4) (2007), 557–593.

[3] Erdős, Paul - Gallai, T.: Gráfok előírt fokú pontokkal (Graphs with prescribed degree of vertices), *Mat. Lapok*, **11** (1960), 264–274. (in Hungarian)

[4] Erdős, Péter L. - Király, Z. - Miklós, I.: On graphical degree sequences and realizations, manuscript (2012).

[5] Hakimi, S.L.: On the realizability of a set of integers as degrees of the vertices of a simple graph. *J. SIAM Appl. Math.* **10** (1962), 496–506.

[6] Havel, V.: A remark on the existence of finite graphs. (in Czech), *Časopis Pěst. Mat.* **80** (1955), 477–480.

[7] Kannan, R. - Tetali, P. - Vempala, S.: Simple Markov-chain algorithms for generating bipartite graphs and tournaments, *Rand. Struct. Alg. 14* (4) (1999), 293–308.

[8] Hyunju Kim - Toroczkai, Z. - Erdős, P.L. - Miklós, I. - Székely, L.A.: Degree-based graph construction, *J. Phys. A: Math. Theor.* **42** (2009) 392001 (10pp)

[9] Molloy, M. - Reed, B.: A critical point for random graphs with a given degree sequence, *Rand. Struct. Alg.* **6** (2-3) (1995), 161–179.

[10] Newman, M.E.J. - Barabasi, A.L. - Watts, D.J.: *The Structure and Dynamics of Networks* (Princeton Studies in Complexity, Princeton UP) (2006), pp 624.

[11] Ryser, H. J.: Combinatorial properties of matrices of zeros and ones, *Canad. J. Math.* **9** (1957), 371–377.

[12] Sinclair, A.: Improved bounds for mixing rates of Markov chains and multicommodity flow, *Combin. Probab. Comput.* **1** (1992), 351–370.

[13] Steinhaus, H.: *Mathematical Snapshots*, Oxford University Press, New York, 1950. pp 30.

[14] Wormald, N.C.: Generating random regular graphs, *J. Algorithms* **5** (1984), 247–280.