Towards random uniform sampling of bipartite graphs with given degree sequence

István Miklós and Péter L. Erdős and Lajos Soukup
Alfréd Rényi Institute of Mathematics,
Hungarian Academy of Sciences,
Budapest, P.O. Box 127, H-1364 Hungary
`<miklosi,elp,soukup>@renyi.hu`

November 16, 2011

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 *semi-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, [4]) 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 [10].)

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 [13] for a detailed analysis, or the paper [10] 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, [12]) 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 [17].) 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, [9]), 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 semi-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 good multicommodity flow.

2 Basic definitions and preliminaries

Let $G = (U, V; E)$ be a simple bipartite graph (no parallel edges) with vertex classes $U = \{u_1, \dots, u_l\}, V = \{v_1, \dots, v_k\}$. 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** $(u_1, u_2; v_1, v_1)$ 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, [14]).

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 $d(u') \le d(u)$, 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 $\Gamma_{G'}(v) = (\Gamma_G(v) \setminus \{u'\}) \cup \{u\}$.*

Proof: By the pigeonhole principle there exists a vertex $v' \neq v$ such that $(u, v') \in E$ and $(u', v') \notin E$. So 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, $d(u_1) \ge d(u_2) \ge \cdots \ge d(u_k)$ 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 $\Gamma_{G'}(v) = \{u_1, \dots, u_d\}$.*

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

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*

by induction on e. Assume that (✠) holds for $e' < e$. We can assume that $d(u_1) \ge d(u_2) \ge \cdots \ge d(u_k)$. By the Havel-Hakimi Push-up Lemma there is a sequence T_1 (T_2) of $d = d(v_1)$ many swaps which transforms G_1 (G_2) into a G_1' (G_2') such that $\Gamma_{G_1'}(v_1) = \{u_1, \dots, u_d\}$ ($\Gamma_{G_2'}(v_1) = \{u_1, \dots, u_d\}$).

We consider the bipartite graphs $G_1'' = G_1' \setminus \{v_1\}$ and $G_2'' = G_2' \setminus \{v_1\}$, i.e. we remove the vertex v_1 and all the edges connected to v_1. Since $\mathrm{bd}(G_1'') = \mathrm{bd}(G_2'')$ and the number of edges of G_i'' is $e - d$, by the inductive assumption there is a sequence T of $2(e-d)$ many swaps which transforms G_1'' into G_2''.

Now observe 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 the swap sequence T_2 has an inverse T_2' which transforms G_2' into G_2. Hence the sequence $T_1 T T_2'$ is the required swap sequence: it transforms G_1 into G_2 and its length is at most $d + 2(e-d) + d = 2e$. □

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

For a bipartite graphical sequence $\big(\mathbf{a}, \mathbf{b}\big)$ - 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 (and, recall, the inverse swap transforms the second one to the first one as well).
P denotes the *transition probability matrix*, which are defined as follows: if the current realization (status 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*. Furthermore since in each realization (state) our random walk can stay in that state with positive probability (except for the unique degree sequence $((1,1),(1,1))$), therefore our Markov process is clearly aperiodic.

Finally since, as we saw, the jumping probabilities are symmetric, that is $P(G|G') = P(G'|G)$, therefore our 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, so simplify the notations. Let denote P^t the tth power of the transition probability matrix and define

$$\Delta_x(t) := \frac{1}{2} \max_{y \in \mathbb{G}} \left| P^t(y,x) - 1/N \right|,$$

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

$$\tau_x(\varepsilon) := \min_t \{\Delta_x(t') \leq \varepsilon \text{ for all } t' \geq t\}.$$

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

$$\tau_x(\varepsilon) \leq O\Big(\text{poly}\big(log(N/\varepsilon)\big)\Big).$$

for any x in the state space. Consider the different eigenvalues of P in decreasing order:

$$1 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_N \geq -1.$$

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 explicitely stated by Sinclair:

Theorem 4.1 (Sinclair, [15]). $\tau_x(\varepsilon) \leq \tau_{rel} \cdot \text{poly}(\ln \varepsilon^{-1} + \ln N)$. □

So one way to prove rapid convergence is to find a polynomial upper bound of τ_{rel} and we need rapid convergence of the process to the stationary distribution otherwise the method cannot be used in practice.

Kannan, Tetali and Vempala in [9] could prove that the relaxation time of the Markov chain $(\mathbb{G}, P)$ is a polynomial function of the size $n := 2k$ of $(\mathbf{a}, \mathbf{b})$ if it is a regular bipartite degree sequence. Here we extend their proof to show the fast convergence of the process for the semi-regular bipartite case.

There are several different methods to prove fast convergence, here we use - similarly to [9] - Sinclair's *multicommodity flow method* ([15]).

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 jumping probabilities in $\mathcal{M}$ satisfy $P(u|v)P(v|u) \neq 0$. For all $x \neq y \in V(\mathbb{H})$ denote $\Gamma_{x,y}$ a set of paths in $\mathbb{H}$ connecting x and y together with a probability distribution $\pi_{x,y}$ 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

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

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 jumping probability has the property $P(x|y) \geq 1/n^4$ (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:

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

We make one more simplification, namely, we assume that there is some natural number S such that for each x and y we have a path $\varUpsilon(X, Y, s)$ from X

to Y, and $\Gamma_{x,y} = \{\Upsilon(X,Y,s) : s < S\}$. Since it might happen that $\Upsilon(X,Y,s) = \Upsilon(X,Y,s')$ we should define the probabilities $\pi_{x,y}(\gamma)$ as follows:

$$\pi_{x,y}(\gamma) = \frac{|\{s : \gamma = \Upsilon(x,y,s)\}|}{S}.$$

So we simplify RHS of (4.3) as follows:

$$\frac{1}{N}\left(\sum_{\substack{x,y\in V(\mathbb{H})\\ \gamma\in\Gamma_{x,y}\,:\,e\in\gamma}} \pi_{x,y}(\gamma)\right) = \frac{\big|\{(x,y,s) : e \in \Upsilon(x,y,s) : x,y \in V(\mathbb{H}), s < S\}\big|}{N\cdot S}. \tag{4.4}$$

So our task is to find a multicommodity flow $\{\Gamma_{x,y} : x,y \in V(\mathbb{G})\}$ such that

- each path is shorter then an appropriate poly(n) function,
- in (4.4) we have a polynomial upper bound for the right side.

5 Multicommodity flow - general considerations

Our construction method for multicommodity flow commences on the trail of Kannan, Tetali and Vempala ([9]), and Cooper, Dyer and Greenhill ([2]).

We fixed a bipartite graphical sequence $\big(\mathbf{a},\mathbf{b}\big)$, and the vertices of the graph $\mathbb{G}$ are the realizations of $\big(\mathbf{a},\mathbf{b}\big)$. We can assume that if $H \in \mathbb{G}$, then H is a simple bipartite graph $(U_0, U_1; E(H))$, where U_0 and U_1 are fixed finite sets. Let $U = U_0 \cup U_1$.

If $X,Y \in V(\mathbb{G})$ let $E(X \bigtriangleup Y)$ be the symmetric difference of the edge sets $E(X)$ and $E(Y)$, set $E(X-Y) = E(X)\setminus E(Y)$, and $E(Y-X) = E(Y)\setminus E(X)$. The edges in $E(X-Y)$ are the X*-edges* and the edges in $E(Y-X)$ are the Y*-edges*. Since X and Y have the same degree sequence, for every vertex $u \in U$ the number of X-edges adjacent to u is the same as the number of Y-edges adjacent to u.

Before we describe the construction of our multicommodity flow we should recall some definitions:

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

We start our construction process as follows: for each $X,Y \in V(\mathbb{G})$ and for each $u \in U = V(X) = V(Y)$ define $\mathcal{S}\big(u; E(X-Y), E(Y-X)\big)$ as the collection of all one-to-one correspondences between X-edges adjacent to u and Y-edges adjacent to u. Formally:

$$\mathcal{S}\Big(u; E(X-Y), E(Y-X)\Big) = \\ = \Big\{f \cup f^{-1} \mid f : E(X-Y)(u) \to E(Y-X)(u) \text{ is a bijection}\Big\}.$$

We also define

$$\mathbb{S}(X,Y) = \Big\{s : s \text{ is a function, } \mathrm{dom}(s) = U, s(u) \in \mathbb{S}\Big(u; E(X-Y), E(Y-X)\Big)\Big\}$$

and

$$\mathbb{S} = \bigcup \Big\{\mathbb{S}(X,Y) : X, Y \in V(\mathbb{G})\Big\}.$$

For $s \in \mathbb{S}$ let

$$\Delta_s = \bigcup \Big\{\mathrm{dom}\big(s(w)\big) : w \in U \cup V\Big\}.$$

If $s \in \mathbb{S}(X,Y)$ then clearly

$$\Delta_s = E(X \triangle Y). \tag{5.1}$$

If M and M' are $m \times m'$ matrices then let $\mathfrak{d}(M, M')$ be the number of non-zero elements in $M - M'$. For $G \in V(\mathbb{G})$ let M_G be the bipartite adjacency matrix of G. For $X, Y, Z \in V(\mathbb{G})$ write $\widehat{M}_{X,Y,Z} = M_X + M_Y - M_Z$. (These matrices essentially *encode* the paths from X to Y along Z.)

Outline of the construction of the path system.

(A) Fix a total order $\preceq$ on the edges of $E(X) \cup E(Y)$.

To each $s \in \mathbb{S}$ assign an *ordered* circuit decomposition

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

of Δ_s as follows:

(a) Consider the line-graph of Δ_s: this is the graph $\mathcal{G} = (\Delta_s, \mathcal{E})$, where

$$\Big((u,v), (u'v')\Big) \in \mathcal{E} \text{ iff } u = u' \text{ and } s(u)(u,v) = (u,v').$$

(b) $\mathcal{G}$ is a 2-regular graph because $(u,v) \in \Delta_s$ is adjacent to $s(u)(u,v)$ and $s(v)(u,v)$, so $\mathcal{G}$ is the union of vertex disjoint cycles $\{W_i^s : i \in I\}$. The inverse images of these cycles form circuits in $E(X \triangle Y)$, so $\{W_i^s : i \in I\}$ is a decomposition of Δ_s into circuits.

(c) Finally we order the set $\{W_i^s : i \in I\}$ accordingly a suitable order derived $\preceq$ to obtain the sequence $W_1^s, W_2^s \ldots, W_{k_s}^s$. Such an ordering can be, for example, the *lexicographic* order. In this special case, since the circuits are pairwise disjoint, their order coincides with the order of their smallest edges.

(B) As a consequence of (A), every circuit W_i^s is an alternating circuit, i.e. its edges alternate between X-edges and Y-edges.

(C) Decompose each W_i^s into alternating cycles $\{C_j^{s,i} : j \in J\}$. and order the cycles according to $\preceq$ (here, again, we can take the lexicographic order):

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

(D) Let

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

be the short hand notation of the 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 fix independently and arbitrarily a direction of the cycle. If $a, b \in C$ let $[a,b]_C$ be the walk from a to b in C according to the fixed direction.

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

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

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

- (a) $n_{m_s} \le c \cdot n^2$.
- (b) $E(G_{n_{i+1}}) = \big(E(G_{n_i}) \cup (E(Y) \cap E(C_i)\big) \setminus \big(E(X) \cap E(C_i)\big)$,
- (c) if $n_i \le j \le n_{i+1}$ then there are two vertices a and b in C_{i+1} such that

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

  where

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

- (d) for each $i \le n_{m_s}$ there is $T \in V(\mathbb{G})$ such that

  $$\mathfrak{d}\left(\widehat{M}_{X,Y,H_i}, M_T\right) \le \Omega_2,$$

where c, Ω_1 and Ω_2 are fixed "small" natural numbers.

By definition (see equation (5.1)) $E(X \triangle Y) = \Delta_s$. Denote $(2d_1, \ldots, 2d_h)$ the degree sequence of $E(X \triangle Y)$. Put

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

Then $t_\Delta = |\mathbb{S}(X,Y)|$ for each $X, Y \in V(\mathbb{G})$. So we can use the simplified version of Sinclair's theorem from the previous section with parameter $S = t_\Delta$.

Key Lemma 5.2. *If we can assign paths*

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

according to (A)-(F) *then for each* $e \in E(\mathbb{G})$

$$\frac{\left|\Big\{(X,Y,s)\ :\ X,Y \in V(\mathbb{G}),\ s \in \mathbb{S}(X,Y),\ e \in E(\Upsilon(X,Y,s)\Big\}\right|}{t_\Delta \cdot N} \leq \text{poly}(n), \tag{5.2}$$

and so our Markov chain is rapidly mixing.

To prove this statement we need some preparations:

Lemma 5.3. *There is function* Ψ *and a parameter set* $\mathbb{B}$ *such that* $\mathbb{B}$ *has* poly(n) *elements, and for each* $X, Y \in V(\mathbb{G})$*, for each* $\gamma = \Upsilon(X,Y,s) \in \Gamma(X,Y)$ *and for each edge* $e = (Z, Z')$ *from* γ *there is* $B \in \mathbb{B}$ *such that*

$$\Psi\left(Z, \widehat{M}_{X,Y,Z}, s, B\right) = (X,Y,\gamma). \tag{5.3}$$

Proof: By definition s determines the circuit decomposition $(W_1^s, \ldots, W_{k_s}^s)$ of Δ_s, and this circuit decomposition in turn determines a canonical alternating cycle decomposition $(C_1^s, \ldots, C_{m_s}^s)$ by (A) – (D).
Assume that $Z = G_j$ in the path $\Upsilon(X,Y,s)$ such that $n_i \leq j \leq n_{i+1}$. Put

$$\begin{aligned} F \quad = \quad & \bigcup_{i'<i}\Big\{E(C_{i'}) \cap E(Z)\Big\} \ \cup \Big([a,b]_{C_i} \cap E(Z)\Big) \\ & \cup\Big([b,a]_{C_i} \setminus E(Z)\Big) \ \cup \bigcup_{i'>i}\Big\{E(C_{i'}) \setminus E(Z)\Big\}. \end{aligned}$$

Then, by (F)(b) and (c)

$$\Big|\big(E(X) \setminus E(Y)\big) \bigtriangleup F\Big| \leq \Omega_1.$$

So if the parameter B is just the quadruple $\Big(j, a, b, \big(E(X) \setminus E(Y)\big)\Delta F\Big)$, then using this parameter we can determine $E(X) \setminus E(Y)$. Since $j, a, b \leq n$ and $(E(X) \setminus E(Y)) \bigtriangleup F$ is an at most Ω_1 element subset of $U \times U$, the size of the parameter set is polynomial:

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

Since Z and $\widehat{M}_{X,Y,Z}$ determine $E(X) \cap E(Y)$, we can compute $E(X)$. Similarly we can compute $E(Y)$. Then path $\gamma = \Upsilon(X,Y,s)$ is uniquely determined by X, Y and s. □

Proof of the Key Lemma: 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) each $\widehat{M}$ is in an Ω_2 radius neighborhood of some $K \in V(\mathbb{G})$, so

$$|\mathfrak{M}| \leq \text{poly}(n) \cdot |V(\mathbb{G})| = \text{poly}(n) \cdot N.$$

Using Lemma 5.3, we have a parameter set $\mathbb{B}$ having at most poly(n) elements such that

$$\Big\{(X,Y,\gamma):\ X,Y\in V(\mathbb{G}),\ s\in\mathbb{S}(X,Y),\ e\in\gamma=\varUpsilon(X,Y,s)\Big\}\subset \\ \Big\{\Psi\left(Z;\widehat{M};s,B\right):\ Z\in e,\ \widehat{M}\in\mathfrak{M},\ b\in\mathbb{B},\ e\in\gamma\Big\}.$$

Fix Z, $\widehat{M}$, and B. Let

$$\mathfrak{X}(\widehat{M},B,Z)=\Big\{(X,Y,s):\ \Psi\left(Z,\widehat{M},s,B\right)=(X,Y,\gamma)\wedge e\in\gamma\Big\}.$$

We claim
$$\left|\mathfrak{X}\left(\widehat{M},B,Z\right)\right|\le \mathrm{poly}(n)\cdot t_\Delta. \tag{5.4}$$

Assume that $(X,Y,s)\in\mathfrak{X}(\widehat{M},B,Z)$. Fix j such that $Z=G_j$ and $n_i\le j\le n_{i+1}$.

Since we know $\widehat{M}=M_X+M_Y-M_Z$ and Z, we can compute $\widehat{M}_X+\widehat{M}_Y$ so we know the edge set $\Delta=E(X)\cup E(Y)$.

Observe that s is “almost” in $\mathbb{S}(\Delta\cap E(Z),\Delta\setminus E(Z))$. More precisely there is $s'\in\mathbb{S}(\Delta\cap E(Z),\Delta\setminus E(Z))$ such that

$$\big|\big\{(u,v):u\in U\wedge s(u)(v)\ne s'(u)(v)\big\}\big|\le\theta \tag{5.5}$$

for some constant θ. Indeed, by (F)(c), apart from a few edges, for $i'\ne i$ the cycles $C_{i'}$ alternates between $\Delta\cap E(Z)$ and $\Delta\setminus E(Z))$ by (F)(c). Moreover both $[a,b]_{C_i}$ and $[b,a]_{C_i}$ are “almost alternate”. This proves (5.5) which implies (5.4) because $|\mathbb{S}(\Delta\cap E(Z),\Delta\setminus E(Z))|=t_\Delta$.

Now we can put together the proof of the Key lemma.

$$\Big\{(X,Y,s):\ X,Y\in V(\mathbb{G}),\ s\in\mathbb{S}(X,Y),\ e\in E\big(\varUpsilon(X,Y,s)\big)\Big\}= \\ \bigcup_{M\in\mathfrak{M}}\ \bigcup_{Z\in e}\ \bigcup_{B\in\mathbb{B}}\mathfrak{X}\left(\widehat{M},B,Z\right), \tag{5.6}$$

and so

$$\left|\Big\{(X,Y,s):\ X,Y\in V(\mathbb{G}),\ s\in\mathbb{S}(X,Y),\ e\in E\big(\varUpsilon(X,Y,s)\big)\Big\}\right|\le \\ |\mathfrak{M}|\cdot|e|\cdot|\mathbb{B}|\cdot\left|\mathfrak{X}\left(\widehat{M},B,Z\right)\right|\le\big(poly(n)\cdot N\big)\cdot 2\cdot poly(n)\cdot(poly(n)\cdot t_\Delta)= \\ poly(n)\cdot t_\Delta\cdot N, \tag{5.7}$$

so we verified (5.2). □

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

- Fix $X,Y\in V(\mathbb{G})$.

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

  $$C_0, C_1, \ldots, C_\ell. \tag{5.8}$$

  of $E(X \triangle Y)$.

We want to define a path

$$X = G_0, \ldots, G_i, \ldots, G_m = Y$$

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

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

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

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

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

  $$H_i = G'_0, G'_1, \ldots, G'_{m'} = H_{i+1}.$$

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

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

$$G = G_0, \ldots, G_m = G' \tag{5.9}$$

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

(a) $m \leq c \cdot |C|$,

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

We will carry out this construction in the next sections. The burden of such a construction is to meet requirement (b). In [9] 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 semi-regular then the paths $\Upsilon(X,Y,s)$ satisfy the previous condition (b)
- if our bipartite degree sequence is arbitrary, then $\Upsilon(X,Y,s)$ satisfies (b) provided the symmetric difference of X and Y is a cycle.

Originally we conjectured, that our friendly path method always produces paths which satisfy (b). 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

In this section we describe the construction a path along an alternating cycle C.

In what follows we will imagine our cycles as convex polygons in the plane, and we will denote 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 $(u_1,v_1),(v_1,u_2),\ldots,(u_\ell,v_\ell),(v_\ell,u_1)$ 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.

When we start the process we have realization G, when we finish it we have realization G'. These two have almost the same edge set, except that the X-edges in C belong to G while G' contains the Y-edges. 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 denoted M_G the adjacency matrix of the bipartite realization G where the columns and rows are indexed by the vertices of V and U resp., but the columns are numbered from left to right while the rows are numbered from bottom to the top. (Like in 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 edge is missing from both X and Y but it exists in Z. This is 2 if the edge is missing from Z but exists in both X and Y. It is 1 if the 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 edge is missing from all three graphs, or the edge exists in exactly one of X and Y and 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. *The row and column sums of $\widehat{M}(X+Y-Z)$ are the same as row and column sums in M_X (or M_Y or M_Z).* $\square$

We need some more notions:

Definition 6.2. The **type** of a chord is 1 or 0 depending whether it is an edge in G or - what is the same - in G'. A chord $f=(u_\alpha,v_\beta)$ is a **cousin** of a chord $e=(u_\epsilon,v_\delta)$, if $0\le\delta-\alpha\le 1$ and $0\le\epsilon-\beta\le 1$. A chord e is **friendly** if at least one of its cousins has the same type as e itself.

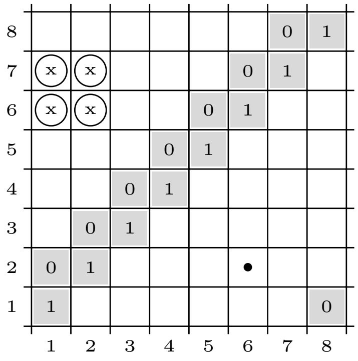
Since the notion of the cousins is not a very natural one we want to illustrate it. Therefore now we introduce our main tool what we will use later in this paper to illustrate different procedures in our current realizations. This tool $F(G+G'+Z)$ (or F_Z for short, and we use F for "figure") is a matrix again, similar to the matrix $M_G+M_{G'}+M_Z$. This always illustrate the subgraph in Z spanned by the vertices of one of the alternating (therefore of an even cardinality) cycle C_i. 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.)

The matrix F_Z's entries in the main-diagonal and the entries in the small-diagonal come from the adjacency matrix of M_Z. The other entries come from the corresponding entries of $M_G+M_{G'}+M_Z$. 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 to the opposite ones. Outside of these diagonal entries $-1,0,1,2$ of $\widehat{M}(G+G'-Z)$ become $1,0,3,2$ in F_Z. Since in this alternating cycle G and G' contain the same chords 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.

Now Figure 1 illustrates the cousins of the chord (u_6, v_2) in the initial Z. (They are (u_1, v_6), (u_1, v_7), (u_2, v_6), finally (u_2, v_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: An edge and its cousins



Definition 6.3. A sequence of pairwise distinct chords $e_1, \ldots, e_j$ is a **friendly path** if

(i) each chord is friendly,

(ii) each e_h has precisely one common endpoint with e_{h+1} (therefore the other end points are in distance 2),

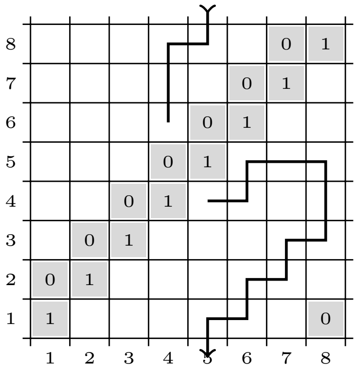
(iii) finally e_1 and e_j are shortest chords and their roots have different types.

A friendly path can be quite complicated, and it is important to remark that such a friendly path is NOT a path in a particular graph. The name is justified by the image of the friendly path in the illustration of F_Z what we are going to show next. (It shows the path itself, but it does not show why the individual elements of the path are friendly - see Figure 2.) This figure is not for illustration only: whenever we consider a friendly path we always work on the matrix itself.

6.1 There exists a friendly path in our cycle

In this subsection we describe the construction of the path along this cycle in the case that a friendly path (going from the main-diagonal to the small-diagonal) exists. Let the chords of the existing friendly path correspond to the positions $A_1, \ldots, A_\Lambda$ where $A_j = (a_j^1, a_j^2)$. Usually each cycle under process is small comparing with the full graph, therefore we always consider a "comfortably reordered" adjacency matrix (in other words, we apply a suitable permutation

Figure 2: A friendly path



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 other vertices do not play roles at the actual phases of our algorithm.) It is **important** to remark, that all descriptions of this subsection use the language of the matrices F_Z (that is the entries can be $0, 1, 2, 3$), unless it is announced otherwise.

Before we start 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 $(1, i)$.

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) $\|A_j, A_{j+1}\| = 1$. Finally (iii) A_1 is in distance 1 from the main-diagonal, while A_Λ is in distance 1 from the small-diagonal.

First we introduce two new structures:

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

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

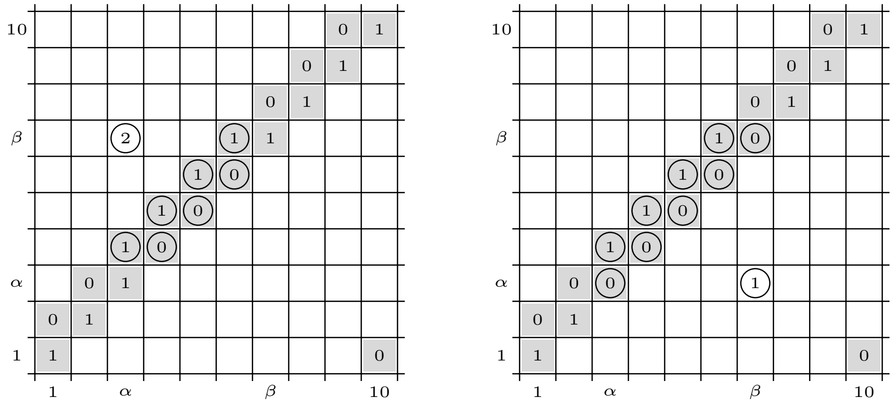
(ii) $m_{i,i} = 0$ for $\alpha < i < \beta$, and $m_{ii} = 1$ for $i \leq \alpha$ and $\beta \leq i$,

(iii) $m_{i,i+1} = 1$ for $\alpha \leq i < \beta$, and $m_{i,i+1} = 0$ for $i < \alpha$ and $\beta \leq i$

(See 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 $M_{G'}$).

Definition 6.5. Let $1 \leq \alpha, \beta \leq \ell$, $\alpha + 1 < \beta$. We say an $\ell \times \ell$-matrix $M = (m_{i,j})$ is (β, α)**-KO matrix** iff

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



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

(ii) $m_{i,i} = 0$ for $\alpha \leq i \leq \beta$, and $m_{ii} = 1$ for $i < \alpha$ and $\beta < i$,

(iii) $m_{i,i+1} = 1$ for $\alpha \leq i < \beta$, and $m_{i,i+1} = 0$ for $i < \alpha$ and $\beta \leq i$

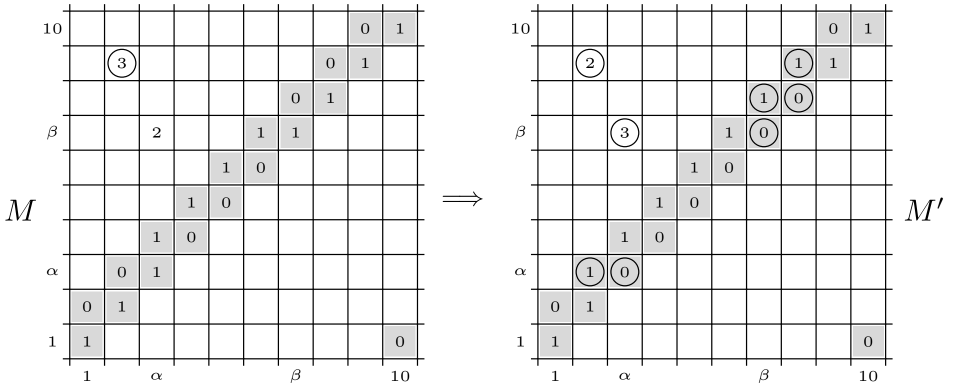
(See Figure 3) Please recall that the entry 1 in F_Z is an edge which exists in Z but missing from both G and G'.

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

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

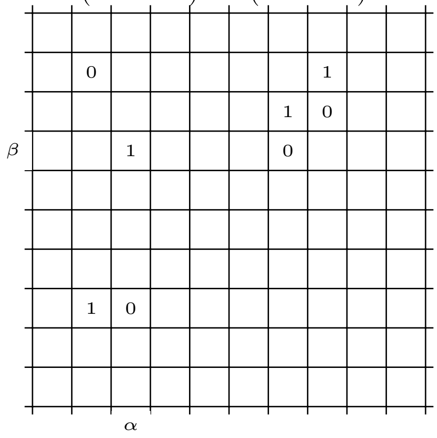
(2) $m'_{i,j} = m_{i,j}$ *if* $i \neq j$, $i + 1 \neq j$, *and* $(i, j) \neq (\alpha, \beta)$, $(\alpha - 1, \beta + 2)$.

Then there exists an absolute constant Θ such that one can transform M into M' with at most Θ swaps.



Proof: : It is enough to observe that the symmetric difference of M and M' is a cycle which alternates between M and M'. Indeed, in the next figure (1) indicates edges in $E(M'-M)$ and (0) indicates edges in $E(M-M')$. (The non-empty positions of this figure are the circled positions in the previous matrix M'.)

$E(M'-M) \cup E(M-M')$



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

So the difference of the realizations lay within in a subgraph with 8 vertices $\bar{V}$. The subgraphs in the two original graphs spanned 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 four vertices from both classes, and there at most 12 edges, therefore the number of the necessary swaps (by Corollary 3.1) is at most 2×12 therefore $\Theta = 24$ is an upper bound of 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 $M = (m_{i,j})$ is an (α,β)-OK matrix and $m_{\alpha',\beta'} = 3$ where*

$$\|(\alpha,\beta);(\alpha',\beta')\| < u,$$

furthermore $M' = (m'_{i,j})$ 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 M into M'.

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

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 $M = (m_{i,j})$ is an (β, α)-KO matrix and $m_{\beta',\alpha'} = 0$ where*

$$\| (\beta, \alpha); (\beta', \alpha') \| < u,$$

furthermore $M' = \big(m'_{i,j}\big)$ 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 M into M'.

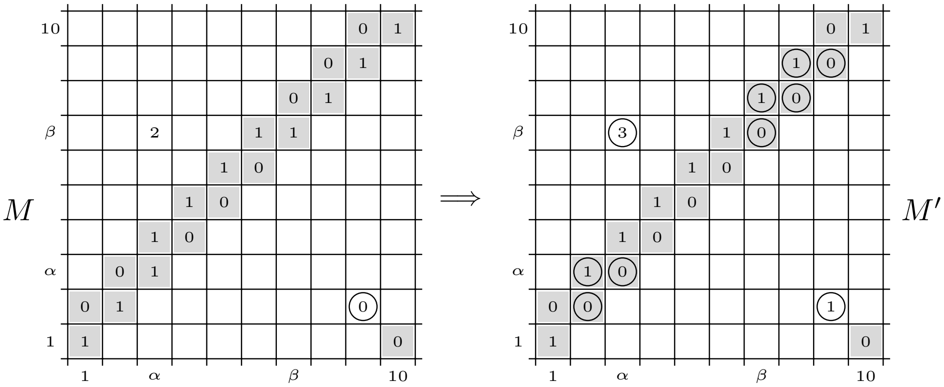
Proof: The proof is very similar to the proof of Lemma 6.7 which is left to the diligent reader. □

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

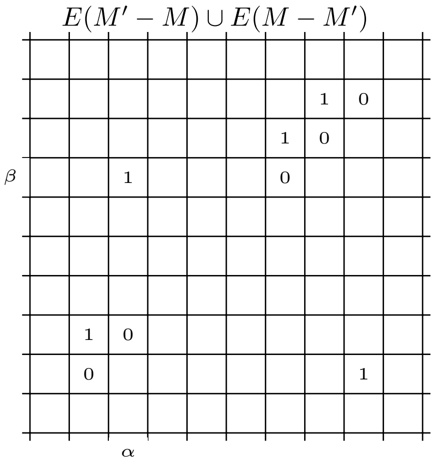
(1) $m'_{\alpha,\beta} = 3$

(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 M into M' with at most Ω swaps.



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



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 lay within in a subgraph with 10 vertices $\bar{V}$. The subgraphs in the two original graphs spanned 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 $\Omega \le 40$ of 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 $M=(m_{i,j})$ is (α,β)-OK and $m_{\beta',\alpha'}=0$ where*

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

and $M'=(m'_{i,j})$ 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 M into M'.

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 getting G' along this sequence, using operations described in the previous Lemmas. At first we define a new sequence $A'_1,\ldots,A'_\Lambda$ from $A_1,\ldots,A_\Lambda$ with the following way:

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

where Cousin(A) denote one position where A has a cousin. (So $M_G(\text{Cousin}(A)) = M_G(A)$.)

Observation 6.11. *By definitions,*

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

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

Next we define the matrix sequence $M_G = L_0, L_1, \ldots, L_\Lambda, L_{\Lambda+1} = M_{G'}$ as follows:

Definition 6.12. The matrix L_i $(i = 1, \ldots, \Lambda)$ is defined from the matrix L_{i-1} by the formulae:

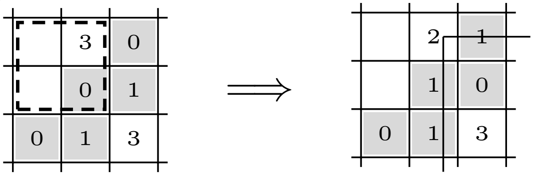
$$L_i = \begin{cases} \text{the } (A'_i)\text{-OK matrix}, & \text{if } L_i(A_i) = 3; \\ \text{the } (A'_i)\text{-KO matrix}, & \text{if } L_i(A_i) = 0. \end{cases}$$

Here all positions (u, v) which are NOT determined by the definitions of the OK- and KO-matrices satisfy $L_i(u, v) = L_{i-1}(u, v)$. ☐

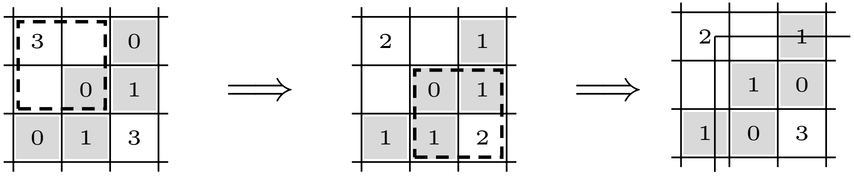
It is quite clear that $(\Lambda - 1)$ consecutive applications of (the appropriate) Lemmas 6.6 - 6.10 will take care the definition of the required swap sub-sequences between L_1 and L_Λ. However, the swap-sequence transforming L_0 into L_1 furthermore the one transforming L_Λ into $L_{\Lambda+1}$ require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

- If $L_1(A_1) = 3$ 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.)

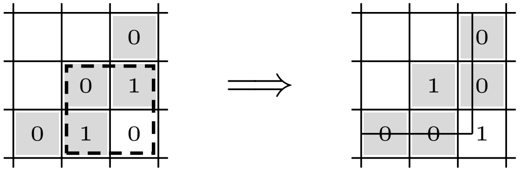
Case I:



and Case II.



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



The connecting swap-sequence from the matrix L_Λ to $L_{\Lambda+1}$ (what is $M_{G'}$) can be defined analogously to the previous one.

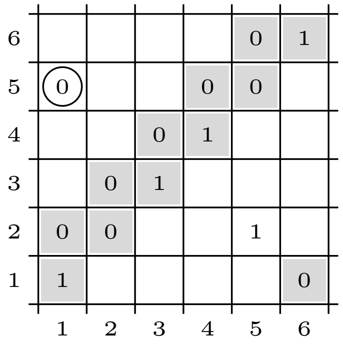
Next we will analyze the behavior of the current matrices $\widehat{M}(G, G", Z)$ along this sub-sequences. At first we consider those Z's which correspond to matrices L_i.

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. The two matrices are in *switch-distance* 1 from each other.

Lemma 6.13. *All adjacency matrices $L_1, \ldots, L_\Lambda$ are in switch-distance* 1 *from the adjacency matrix of some appropriate realizations.*

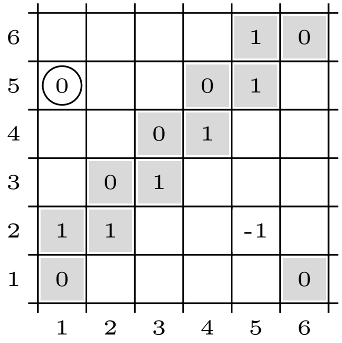
Proof: We show here the statement for such an L_i where $M(A_i) = 0$ therefore L_i itself is an (A'_i)-KO-matrix, and where - by definition - $A_i = A'_i$ (the other case is similar). Due to the definitions A_i originally is not an edge either in G or in G'. It belongs to the friendly path, therefore we also know that Cousin$(A_i) = 0$ also holds in M_G, M'_G. In L_i this value is 1, so A_i is an edge in Z. Therefore L_i which is $= F_Z$ looks like the matrix to the left in the following figure (the circled element is the cousin of A_i). The corresponding $\widehat{M}(G + G' - Z)$ is shown on the right hand side:

$F_Z = M_G + M_{G'} + M_Z$



6					0	1
5	⓪			0	0	
4			0	1		
3		0	1			
2	0	0			1	
1	1					0
	1	2	3	4	5	6

$\widehat{M}(G + G' - Z)$



6					1	0
5	⓪			0	1	
4			0	1		
3		0	1			
2	1	1			-1	
1	0					0
	1	2	3	4	5	6

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

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

Proof: By Observation 6.11 any two 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 subsequences. 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 corresponds to an L_i, then Lemma 6.13 applies, and matrix $\widehat{M}(G+G'-Z)$ has switch-distance 1 from the adjacency matrix of some realization $\in V(\mathbb{G})$.

Let now Z be an intermediate realization in the process, say, between the matrices L_i and L_{i+1}: then $\widehat{M}(G+G'-Z)$ can be transformed through swaps into $\widehat{M}(G+G'-L_{i+1})$ (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 $\frac{1}{2}\max\{\Theta_3, \Theta'_3, \Omega_3\}+1$ 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 chord have the same values, 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 be cause a similar problem and can be handled similarly as this case.)

A good solution for this particular problem would probably ends 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 semi-regular bipartite degree sequences: $(\mathbf{a},\mathbf{b})$ is *semi-regular* if in $\mathbf{a}$ all degrees are the same, while in $\mathbf{b}$ can be anything.

Lemma 6.15. *Assume that our bipartite degree sequence* $(\mathbf{a},\mathbf{b})$ *is semi-regular. Then the statement of Lemma* 6.14 *applies for the matrices* $\widehat{M}(X+Y-Z)$ *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 ready. 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 $\mathbf{a}$).

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 row v contains some 1s. One of them is (w,v) (this w cannot be the column of the 2, since the entry there is 0 due that it belongs to the originally intended switch). Now by the pigeonhole principle there is a row 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. $\square$

6.2 There is no friendly path in our cycle

In this subsection we discuss the case where there is no friendly path in M_G. Our plan is this: at first we show that the non-existence of the friendly paths yields a strong structural property of the matrix M_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. Let M be an $\ell \times \ell$ matrix, then the positions $(i,i),(i+1,i-1),(i+2,i-2),\ldots,(j,j+1)$ form the ith **down-line** of the matrix. (The arithmetic operations are thought to be considered modulo ℓ, that is, for example, $\ell + 1 = 1$ while $1 - 3 = \ell - 2$. The positions $(i,i), (i-1, i+1), (i-2, i+2), \ldots, (j', j'-1)$ form the ith **up-line** of the matrix.

The lines have orientations: the down-lines connect the main-diagonal to the small-diagonal, while an up-line connects the small-diagonal to the main-diagonal.

Definition 6.17. A set T of positions of an $\ell \times \ell$ matrix called **connected** if a chess king, staying inside T, can visit all elements of T.

The following lemma is a well-known version of the classical Steinhaus lemma (see [16]).

Lemma 6.18. *Assume that the positions of an $\ell \times \ell$ matrix M are colored for white and black. Then either the king has a white path from the main diagonal to the small diagonal, or there is a connected set T of black positions which intersects all rook's path from the main diagonal to the small diagonal.* □

We use the previous result without proof. The set T, which was identified in the previous lemma, will be called a *Steinhaus set.*

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 $\cup\{\mathfrak{C}(e) : e \in T\}$.

Lemma 6.20. *Assume that in the matrix M_G there is a connected set T of non-friendly positions. Then the type of all positions in the cousin-set $\mathfrak{C}(T)$ are the same (all chords have the same type). All edges in T have the opposite type.*

Proof: 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 common position(s). Therefore all type 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 of unfriendly positions in M_C, then its cousin-set $\mathfrak{C}(T)$ intersects all down-lines and up-lines.*

Proof: Actually we can prove more: namely that any king-path from the main-diagonal to the small-diagonal intersects the cousin-set $\mathfrak{C}(T)$. We will argue by contradiction, assume that there exists a king-path which does not cross the cousin-set. For position P let denote $I(P)$ be those positions whose cousin-sets contain P. Then if $P \in$ king path, then $I(P)$ is disjoint from $\mathfrak{C}(T)$ (otherwise the king-path intersects T). But any two neighboring positions in the king-path have overlapping $I(P)$ sets. Therefore we have an overlapping sequence of $I(P)$ sets, connecting the main-diagonal to the small-diagonal. And such an overlapping sequence clearly contains a rock-path, connecting the main-diagonal to the small-diagonal. A contradiction. Finally it is clear, that every down- and up-line forms a required king-path. □

Lemma 6.22. *We assume that in the matrix M_C there is no friendly path. Then for each i $(i = 1, \dots, \ell)$ there exists a pair of j, j' s.t. $0 \le j' - j \le 1$ and all entries $(i+1, i-1, \dots, (i+j, i-j)$ have the same type $\mathbf{t}$, furthermore the entries $(i-1, i+1), \dots, (i-j'+1), i+j'-1)$ have type $1-\mathbf{t}$ while $(i+j', i-j')$ has type $\mathbf{t}$ again.*

Proof: Assume indirectly, there is no such j for a particular i. W.O.L.G. we may assume, that $M(i+1, i-1) = 0$. Then, by the assumption, $M(i-1, i+1) = M(i-2, i+2) = 1$ must hold. Then, again by our assumption, $M(i+2, i-2) = 0$ must hold, etc. All entries along the down-line are 0, while all entries along the up-line must be 1. However both lines intersect (see Lemma 6.21) 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. $\square$

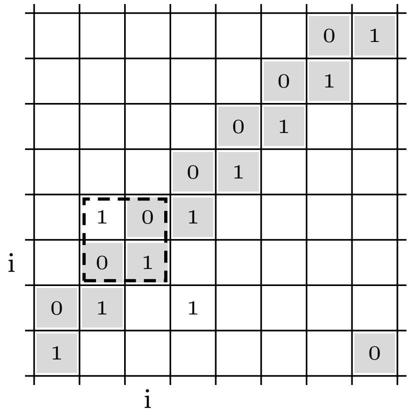
Corollary 6.23. *If conditions of Lemma 6.22 hold, and $j' \ge 2$ for a particular i (in which case $j' = j+1$), the submatrix spanned by $(i+j, i-j)$ and $(i-j, i+j)$ contains friendly path(s).*

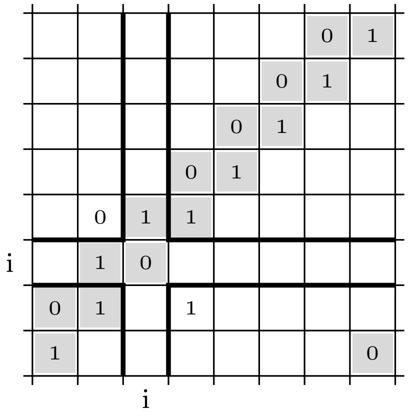
Proof: We argue by contradiction: assume that the submatrix does not contain a friendly path. Then - due to Lemma 6.18 - it contains a Steinhaus set. Due to Lemma 6.20, in its cousin-set $\mathfrak{C}(T)$ - which intersects all down- and up-lines - all positions have the same type. But it contradicts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

That finishes the preliminaries what 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 when $j' \ge 2$. We start with the

First possibility: assume that for a particular i our $j' = 1$. We should take care for two cases:

Case 1: If $M_G = (i+1, i-1) = M_G(i-1, i+1) = 1$ 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 dash-ed square in our illustration. Here we use the adjacency matrix M_G.) The entries $(i-1, i), (i, i)$ and $(i, i+1)$ have the required types. However, entry $M(i-1, i+1) = 0$ therefore along the procedure we should take care to change it back to its original value.

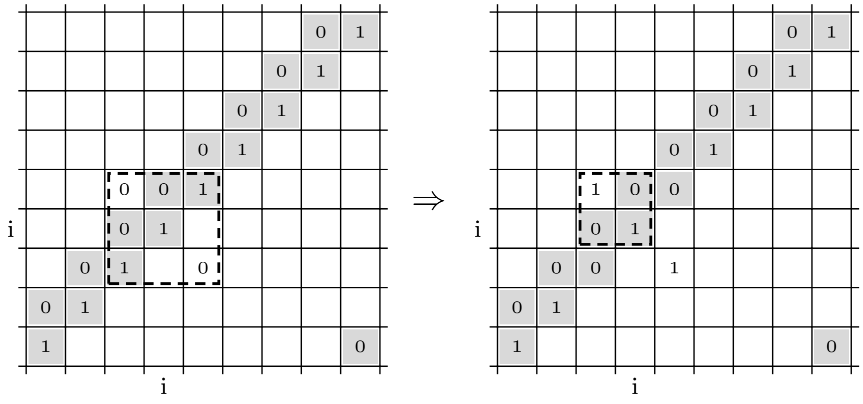




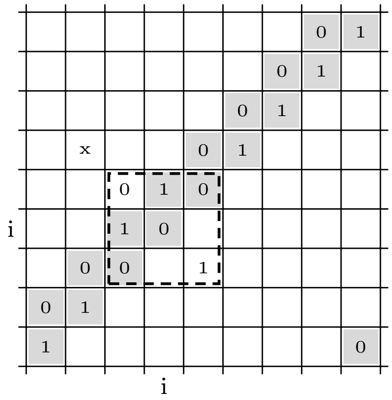
The remaining subproblem, indicated by thick black lines (for a formal description see case $j' \geq 2$), fortunately is already in the required form. Indeed: its main-diagonal contains only 1s, while its small-diagonal is full with 0s. Denote the alternating cycle of this smaller problem by C'. The (recursive) solution of the subproblem C' will switch the value of $M(i-1, i+1)$ automatically back to 1. Since the recursive procedure can use any down- and up-lines (see Corollary 6.23), 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 have 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 $\widehat{M}$ with at most one: the positions $(i-1, i), (i, i)$ and $(i, i+1)$ (outside of our subproblem), provides a suitable switch to handle the entry 2 at position $(i-1, i+1)$.

Case 2: Now we have $M_G = (i+1, i-1) = M_G(i-1, i+1) = 0$. Here we perform two swaps, as it is shown below (the places of the swaps are denoted with dash-ed squares):



If $\widehat{M}(X+Y-Z)(i+1,i-1)=-1$ holds, then it increases the switch-distance of the current $\widehat{M}$ with 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:



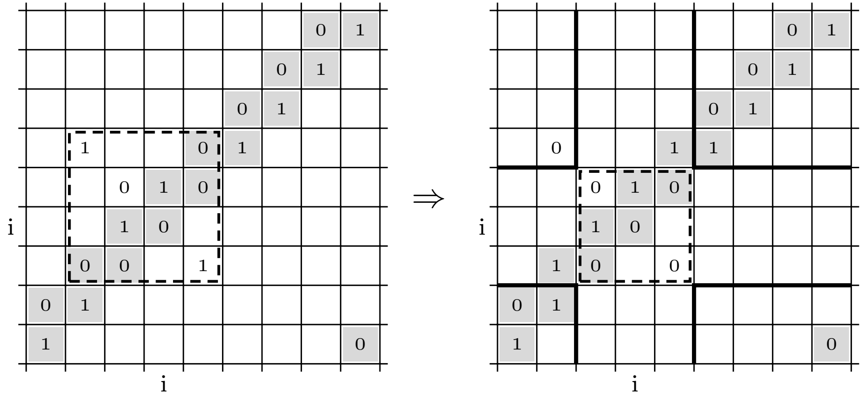
Here we distinguish between two cases, according to the value

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

This value can be $x=1$ and $x=0$.

In the case of $x=1$ we perform a second 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):

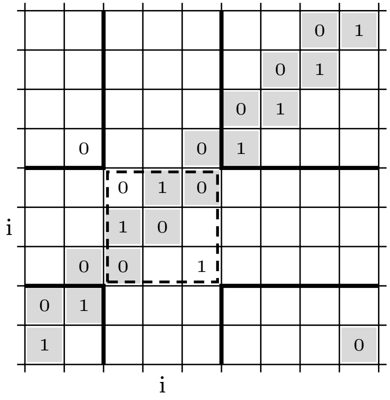
Figure 4: The case of $x=1$



On the RHS of Figure 4, the dash-ed submatrix is in the *right form*: the main diagonal contains all 1's, while each entry in the small-diagonal is 0. The second subproblem (indicated with the thick black lines, the four pieces fit together to a square matrix, again in the right form. We will process the second ("dash-ed") 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 used again Corollary 6.23.)

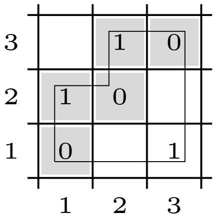
As it happened before $\widehat{M}(X+Y-Z)$ may contain 2 at the position $(i-2,i+2)$. Again this increases the switch-distance with at most one, since the positions $(i-2,i-1),(i+1,i-1)$ and $(i+1,i+2)$ are not in our subproblem.

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

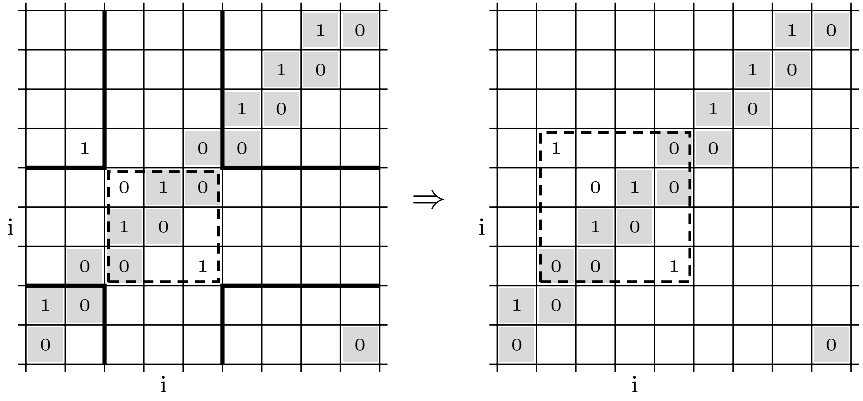


This figure shows the new subproblem (indicating with thick black lines) is in the right form again. We will process the subproblem along the up-line, containing position $(i-2,i+2)$ (so the only currently improper entry will have the right value at the end of the next recursion step).

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 $(i+1,i-1)$, is 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 $\widehat{M}$ with at most δ.

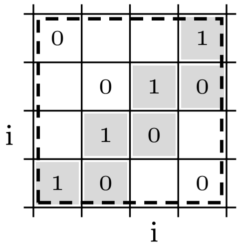


We run the first recursion on the subproblem along the ith up-line, therefore the sub-subproblem with friendly path will contain the position $(i-j',i+j')$. Therefore when we finish the first recursion, our adjacency matrix M_G will be in the following form: (the figure on the left):



We have seen how one can handle the switch-distance of our matrix, if position $(i+1,i-1)$ is problematic (in $\widehat{M}$ it is a -1 there) but position $(i-2,i+2)$ is O.K. On the other, if $\widehat{M}(X+Y-Z))(i-2,i+2)=-1$ then the swap on the positions $(i-2,i-1),(i+1,i+2);(i+1,i-1),(i-2,i+2)$ change 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 dash-ed square in the figure on the upper right). The result is shown to the right:

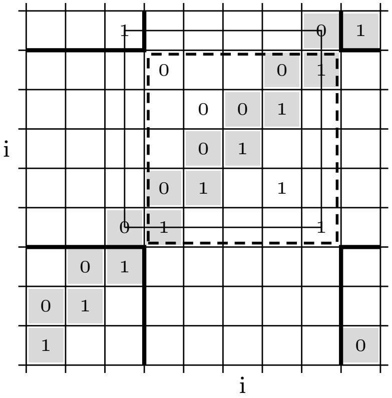


This completes our handling on the First Possibility, that is when for our i we have the value $j'=1$. Now we turn to the other (and probably more common) configuration:

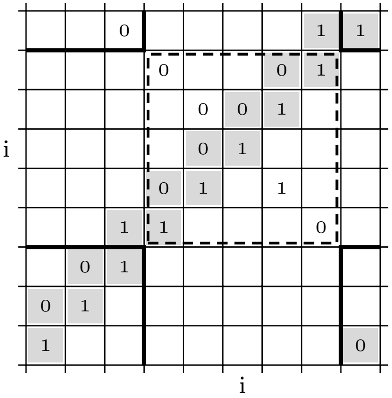
Second Possibility: We have $j' \geq 2$. Unfortunately, the situation can be more complicated in this case due to the switch-distance of $\widehat{M}$. We overcome this problem by showing at first the general structure of the process, and later we give the necessary fine-tuning to ensure the low switch-distance.

In our current alternating cycle (lying in the symmetric difference of G and G') there is no friendly path, therefore there is a T Steinhaus set in the subgraph of Z, spanned by the vertices of the cycle. Now fix a particular i and assume that the j' belongs to this i is ≥ 2. We should distinguish between two cases: where the down-line start with the value $t=1$ or with $t=0$.

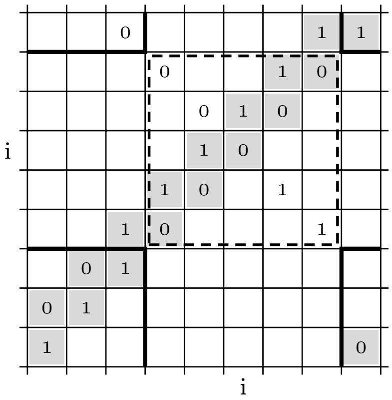
Case 1: $t=1$ The first figure below shows the structure of matrix M_G. The dash-ed square is the first subproblem to deal with, while the thick black lines indicate the second subproblem. However, before we start the processing the subproblems, we have to perform a swap. The corners of the thin black square shows the positions of the swap. After that, the first subproblem (indicated by the dash-ed square) is in the right form. (See the figure on the right.)



$\Rightarrow$



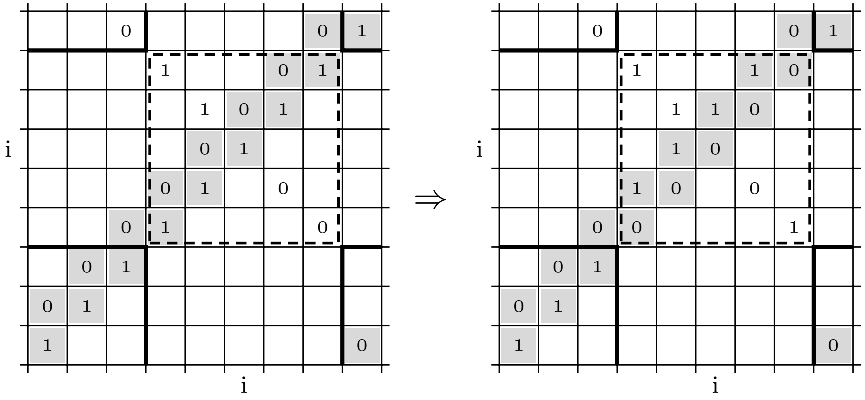
Finishing the first subproblem, we have the following $M_{C'}$ adjacency matrix:



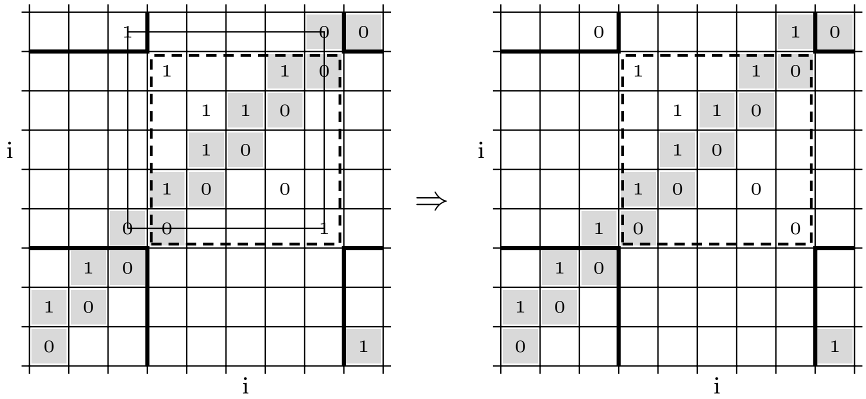
As it can be seen, after the first phase, all entries in the dash-ed are in their required types: the small-diagonal consists of 1s (including position $(i+2, i-2)$ which in that way is back to its original type), while the main-diagonal consists of only 0s.
The second subproblem (indicated by the thick black lines) in the right form now (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 dash-ed subproblem it may changes between 0 and 1. Finishing the dash-ed subproblem, it will be in the type as it starts.

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

Case 2: $t = 0$ The first figure below shows the structure of matrix M_G. The dash-ed 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 position, 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. Its 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 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 position changes its value, while the other does not. But in this case the switching which was previously available are not useable anymore. Next we describe how we can fine tuning our procedure to avoid this trap.

As we know the first subproblem contains a friendly path, 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.

Doing this refined algorithm we have the opportunity that at every particular moment we can control with a small number of swaps the $\widehat{M}$ values at positions P_i and P_{i+1}. □

7 Acknowledgement

The authors would like to thank to the anonimous referee, whose comments and suggestions improved the manuscript significantly. The project was supported by OTKA PD 84297.

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] Diaconis, P. - Strook, D.: Geometric Bounds For Eigenvalues of Markov. Chains, *Annals Applied Probability* **1** (1991), 36–61.

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

[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] Hammersley, J.M. - Morton, K.W.: Poor man's Monte Carlo, *J. Royal Stat. Society B* **16(1)** (1954), 23–38.

[8] Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57** (1) (1970), 97–109.

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

[10] 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)

[11] Metropolis, N. - Rosenbluth, A.W. - Rosenbluth, M.N. - Teller, A.H. - Teller, E.: Equations of state calculations by fast computing machines, *J. Chem. Phys.* **21** (6) (1953), 1087–1091.

[12] Molloy, M. - Reed, B.: A Critical Point for Random Graphs with a Given Degree Sequence, *Rand. Struct. Alg.* **6** (2-3) (1995), 161–179.

[13] 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.

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

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

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

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