Uniform Sampling of Graphs with Fixed Degree Sequence under Partition Constraints

Master Thesis

Autor:
Andrin Pelican*

Supervisor:
Prof. Dr. Hochstättler

FernUniversität in Hagen
Fakultät für Mathematik und Informatik

Tomils, 5. April 2019

*Email: andrin.pelican@bluewin.ch, Matrikel-NR.: 9149031

Selbstständigkeitserklärung

Der Verfasser erklärt, dass er die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt hat. Die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind ausnahmslos den wissenschaftlichen Anforderungen entsprechend als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt und auch noch nicht veröffentlicht worden.

Andrin Pelican

Tomils, 17 Februar 2019

Acknowledgement

I want to thank my supervisor Prof. Dr. Winfried Hochstättler for the excellent support and the targeted answers to my questions.

My family supported me in this work and during my studies. Their encouragement and trust in me was a great motivation. I want to take this opportunity to express my sincere thanks.

I am grateful to Fabia Weber, Vincent Johal, Daniele Ballinari and Kaïla A. Munro to proof-read this work.

Contents

1 Introduction

In the analysis of real-world graphs one is interested in inquiring to which extent an observed graph differs from a ensemble of graphs with a given characteristic. This comparison allows to detect deviations from randomness, which are caused by different factors than the characteristics used to construct the ensemble. The most frequent characteristic, which determines the comparison graphs, is the degree sequence. Restricting the comparison ensemble to graphs with a fixed degree sequence accounts for node heterogeneity, which is not considered in the Erdős Rényi model.
The set of characteristics used to construct the comparison ensemble determines a null-model. Under this null-model the only non random features of the graph are these characteristics used to construct the ensemble. The rest is completely random (for a formal description see Section 4). Such null-models are widely used in computational biology [1]. Furthermore, the interest in network analysis is increasing for social sciences. In such a field, there is often the need for controlling different sources of heterogeneity. Consider a network of social relations, where the nodes are persons and an edge is formed if two of them are friends. Certainly one would expect the formation of edges between same sex friendships to be different from the formation of male-female friendship. However, this heterogeneity is not captured by the degree sequence. The main result of this paper is a random draw algorithm which allows to account for such heterogeneity.

As an example we observe the Nyakatoke network. The Nyakatoke network is a social network. It is based on the survey conducted by De Weerdt in Nyakatoke, a small Haya village in the Kagera region of Tanzania [2]. The nodes are the households in that village and the edges are connections of risk-sharing.
The topic of interest is whether risk-sharing links depend on wealth. In particular, we have the

> *Claim:* There are less risk-sharing links between poor and rich households.
>
> *Explanation:* Links have to be formed with consent of both parties. Links are only beneficial for the poor household. Therefore, the probability of a link between these groups is low.

However, we also suspect that education has an impact on the link formation. Since the richer households are usually better educated, the impact of education has to be considered to test the 0-hypothesis.

The ensemble of graphs with a given characteristic is usually by magnitudes too large to be computed explicitly. The method followed in the literature to overcome this problem is to compute a representative sample out of the total ensemble and subsequently compare it to the observation graph. There are two main approaches taken to compute a representative sample:

1. *Importance Sampling:*
   The idea behind importance sampling is to start with an empty graph and iteratively add edges to it until a graph with the required probabilities is constructed. At each step, the edge to be added is chosen randomly but with known probability, such that the importance sampling can be applied to correct for the probability of construction.
   The importance sampling algorithm for graphs with a fixed degree sequence was developed contemporaneously by Blitzstein, Diaconis [3] and Del Genio, Kim, Toroczkai, Bassler [4]. The method was consequently extended to directed graphs [5].

2. *Markov-Chain:*
   The idea behind the Markov-Chain approach is to iteratively apply a random modification on the observed graph. The random modification has certain properties such that after sufficiently many modifications, the distribution of all the graphs is uniform. For a general overview of the Markov-Chain sampling methods, see [6] [7]. The method for bipartite graphs with fixed degree sequence was introduced by Kannan et. all. [8]. Subsequently, the method was generalized to undirected graphs with a fixed degree sequence by Miklos, Podani [9] and for directed graphs by Berger, Hannemann [10].

Although the field of randomly generating graphs is recent, there have been various extensions of the above mentioned methods. Most notable is the curveball method [11], which mixes at least as fast as the classical algorithms [12] [13] but is empirically better.
Fosdick et. all. [14] emphasize the importance of the choice on how to define the ensemble, which the observed graph is compared to. In particular how edge and node numbering of the configuration can influence estimations when applied to multigraphs or graphs with self-loops.
The choice of the characteristics for determining the comparison ensemble can require more than the fixed degree sequence. An obvious choice is to require connectivity, as many observed graphs such as the road network of a city is connected. These properties are inquired in [15] and [16].

The main contribution of this paper is a Markov-Chain sampling algorithm, which considers partition adjacency matrix restrictions. This is a considerable amplification of the configuration models. The correctness of the algorithm is shown. A statistical motivation of the configuration model choice is given. The algorithm is implemented and applied to a real world example[1].

[1] The algorithm can be accessed as python package on PyPi under the name ugd. It includes an API documentation and an overview of the architecture.

Section 2 proposes a Markov-Chain draw algorithm which considers group constraints. In particular, Section 2.1 determines the notation and defines the algorithm. In Section 2.2, the proof of the correctness of the algorithm is developed. Section 2.3 proposes some modifications for efficiency optimization. A discussion of the properties as well as a differentiation to other approaches in the literature is given in Section 2.4. The problem of deciding whether a realization exists to a given set of constraints is examined in Section 3. In a first step a necessary condition is given by a modification of the Havel-Hakimi algorithm in Section 3.1. A literature overview on the existing approaches for a sufficient statistic is given in Section 3.2. Section 4 gives a statistical motivation for the choice of the comparison ensemble. In Section 5, the algorithm is applied to the Nyakatoke network in order to inquire it's properties. The conclusion gives a short summary on the results and makes remarks on further work to be done.

2 The Markov Draw Algorithm

2.1 The Algorithm

We define a degree sequence S as a sequence of integers $(a_1, ..., a_n)$ with $a_i \in \mathbb{N}, i \in \{1, ..., n\}$ where $a_i > 0$. Let $\mathcal{P}$ be a partition of $\{1, ..., n\}$, the elements V_i of the partition have a fixed enumeration. We define the *partition adjacency matrix*, in short PAM, as a symmetric matrix $M \in \mathbb{N}^{|\mathcal{P}| \times |\mathcal{P}|}$.
Let $G = (V, E)$ be a labeled graph without loops and parallel edges and with $|V| = n$. We define the degree-function $d_G : V \to \mathbb{N}$ which assigns to each node $v_i \in V$ the number of incident edges and a cross-edge-function $c_G : \{1, ..., |\mathcal{P}|\} \times \{1, ..., |\mathcal{P}|\} \to \mathbb{N}$ which assigns to $(i, j)$with $i, j \in \{1, ..., |\mathcal{P}|\}$ the number of edges with one node in V_i and the other in V_j.
We denote (S, M) as *graphical* if and only if there exists at least one graph $G = (V, E)$ without any loops or parallel edges which satisfies $d_G(v_i) = a_i$ for all $v_i \in V$ and for every (i, j) with $i, j \in \{1, ..., |\mathcal{P}|\}$ holds $c_G((i, j)) = M_{i,j}$. The decision whether (S, M) is graphical is called the PAM-realization problem. Any such graph G is called *realization* of (S, M). Let $\mathcal{G}$ be the set of all realizations of (S, M).

This paper has the objective to deliver a random algorithm A. A has as input a graph $G \in \mathcal{G}$, node-set-partition $\mathcal{P}$ and a mixing parameter $\tau \in \mathbb{N}$. The mixing parameter τ determines how long the algorithm runs until returning a random graph. A generates a random graph $G' \in \mathcal{G}$, where the probability of G' depends on τ and is denoted with $p_{A,\tau}(G')$. In addition A has the property

$$\forall \epsilon > 0 \ \ \exists T \in \mathbb{N} \ \ \forall \tau > T, \ \ \forall G' \in \mathcal{G} : \ \ \frac{1}{|\mathcal{G}|} - \epsilon < p_{A,\tau}(G') < \frac{1}{|\mathcal{G}|} + \epsilon. \qquad (1)$$

The property (1) will be referred to uniform sampling out of $\mathcal{G}$. The question on how to construct a first realization G, or how to determine whether a realization exists will also be discussed.

For two realizations G, G' of (S, M) the symmetric difference of their edge sets $E(G)$ and $E(G')$ is $(E(G) \setminus E(G')) \cup (E(G') \setminus E(G))$. Consider for example the realizations G and G' with $E(G) := \{\{v_1, v_2\}, \{v_3, v_0\}\}$ and $E(G') := \{\{v_1, v_0\}, \{v_3, v_2\}\}$ consisting of exactly two edges. Then the symmetric difference corresponds to an alternating closed walk $C := (v_1, v_2, v_3, v_0, v_1)$ where $\{v_i, v_{i+1}\} \in E(G)$ and $\{v_{i-1}, v_i\} \in E(G')$ for $i \in 1, 3$ taking indices i mod 4. We define an alternating walk P for a graph $G = (V, E)$ as a sequence $P := (v_1, v_2, ..., v_l)$ of nodes $v_i \in V$ where either ($\{v_i, v_{i+1}\} \in E(G)$ and $\{v_i, v_{i-1}\} \notin E(G)$) or ($\{v_i, v_{i+1}\} \notin E(G)$ and $\{v_i, v_{i-1}\} \in E(G)$) . We call an alternating walk C alternating closed walk if $v_1 = v_l$ and either ($\{v_1, v_2\} \in E(G)$ and $\{v_{l-1}, v_l\} \notin E(G)$) or ($\{v_1, v_2\} \notin E(G)$ and $\{v_{l-1}, v_l\} \in E(G)$) is fulfilled.

Proposition 1. *The symmetric difference of two realizations of (S, M) always decomposes into a collection of alternating closed walks.*

Proof. Let $G = (V, E)$ and $G' = (V, E')$ be two realizations of (S, M). The symmetric difference of G and G' is the graph $G^{\triangle} = (V, E^{\triangle})$ with the edge set

$$E^{\triangle} = E \triangle E' = (E \cup E') \setminus (E \cap E').$$

Consider a node $v \in V$. v has the same degree d in G as in G'. Let E^v be the edges incident to v in both graphs G and G'. Then there are $d - |E^v|$ edges incident to v in G, which are not in G' and there are $d - |E^v|$ edges incident to v in G', which are not in G. In total, there are $2(d - |E^v|)$ edges incident to v in $G^{\triangle}$. v has therefore an even degree in $G^{\triangle}$. Because v was chosen arbitrarily, we conclude that every node has an even degree in $G^{\triangle}$. With the algorithm of Hierholzer the edge set of $G^{\triangle}$ can be decomposed into a collection of alternating closed walks [17]. We can further require the algorithm to iteratively choose edges altering from $E(G)$ and $E(G')$. Then the closed walks can be decomposed into a set of even closed walks, which don't contain another even closed walk. □

Consider a realization G of (S, M). We define the complementary edge-set of G as

$$\bar{E}(G) = \{\{v', v''\} | v', v'' \in V(G) \text{ and } \{v', v''\} \notin E(G)\}.$$

A function $m : \bar{E}(G) \cup E(G) \to \{0, 1\}$ is called an edge marking. We call an edge $\{v', v''\} \in \bar{E}(G) \cup E(G)$ marked if $m(\{v', v''\}) = 1$ and unmarked if $m(\{v', v''\}) = 0$. By the expression "mark an edge $\{v', v''\}$" is meant that the marking function is changed such that $m(\{v', v''\}) = 1$. The expression "unmark" an edge is used analogically. The graph with the edges $\bar{E}(G)$ is denoted with $\bar{G} = (V, \bar{E}(G))$. Let C be a closed alternating walk of even length in G and E^C the edges corresponding to the closed alternating walk C and $E' = (E(G) \setminus E^C) \cup (\bar{E}(G) \cap E^C)$. We say $G' = (V, E')$ is the graph obtained by switching C in G.

Definition 1. *An alternating walk $(v_1, v_2, ..., v_l)$ which uses only unmarked edges and the first edge is in the graph is called a* schlaufe *if one of the following cases is fulfilled:*

1. *If there is a node $v_i \in \{v_1, v_2, ..., v_{l-1}\}$, with $i \neq l$, $v_i = v_l$ and $i - l \mod 2 = 0$. Further it is required that for any two nodes v_k, v_h in $(v_1, v_2, ..., v_{l-1})$ with $v_k = v_h$, $h \neq k$ is $k - h \mod 2 = 1$.*
2. *If at node v_l there is no other node v such that the alternating walk could be extended with the unmarked edge $\{v_l, v\}$.*
3. *If the edge $\{v_{l-1}, v_l\}$ is present in $(v_1, v_2, ..., v_{l-1})$.*

Proposition 2. *In case 1 the schlaufe consists of an alternating walk $(v_1, ..., v_i)$ (which may have length 0) and exactly one even alternating closed walk $C := (v_i, ..., v_l)$ with $v_l = v_i$.*

Proof. Let $(v_1, v_2, ..., v_l)$ be a schlaufe of case 1. By definition there is a v_i such that $i \neq l$, $v_i = v_l$ and $i - l \mod 2 = 0$. $(v_1, v_2, ..., v_l)$ is a schlaufe, therefore in particular an alternating walk. The parts $(v_1, ..., v_i)$ and $(v_i, ..., v_l)$ are alternating walks. $C = (v_i, ..., v_l)$ is closed because $v_i = v_l$ and even because $i - l \mod 2 = 0$. □

For a schlaufe we define the violation matrix $Z \in \mathbb{Z}^{|\mathcal{P}| \times |\mathcal{P}|}$. If the schlaufe is of case 1 then $Z_{i,j}$ is the sum of edges in $E(G) \cap E(C)$ with nodes in V_i and V_j minus the sum of edges in $\bar{E}(G) \cap E(C)$ with nodes V_i and V_j. If the schlaufe is of case 2 or case 3 then Z is zero.

Definition 2. *We call a sequence of schlaufen $\mathcal{R} = (R_1, ..., R_k)$* feasible *if the schlaufen are edge-disjoint, the sum of the violation matrices is zero and for $i < k$ the sum of the violation matrices is not zero.*

For the algorithm two more edge marking functions are needed. One which maps all the nodes V to $\{0, a, b\}$ and one which maps the edges $\bar{E}(G) \cup E(G)$ to $\{0\} \cup \{a^i_j | i, j \in \mathbb{N}\} \cup \{b^i_j | i, j \in \mathbb{N}\}$. The 0 stands for not marked, a for actively marked and b stands for passively marked. The indexing of the a and b is needed to assign a number of the occurrence in the alternating path and the number of the schlaufe. A passively or actively marked node is considered marked. The language use of these functions is analogous to the marking function previously defined.

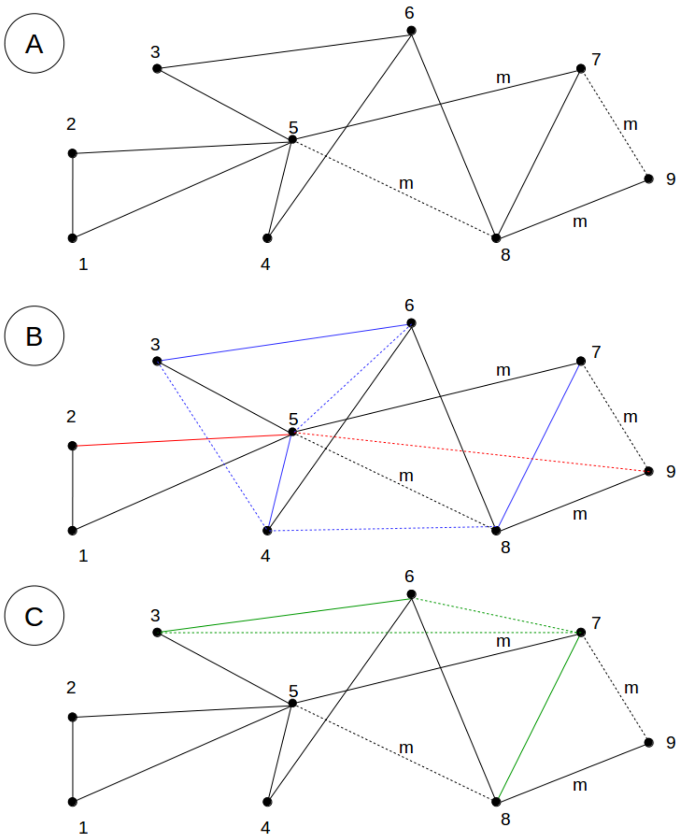


Figure 1: In A a graph G with marked edges is depicted. The edges in $\bar{E}(G)$ are dashed and only drawn if they are in a schlaufe or marked. G has $\{5,7\}, \{5,8\}, \{8,9\}, \{7,9\}$ marked. B shows G with two schlaufen. The blue schlaufe $(7,8,4,5,6,3,4)$ is of case 1. The red schlaufe $(2,5,9)$ is of case 2. The red schlaufe cannot be extended at node 9 because the edge incident with 9 is marked. In C there is a schlaufe $(8,7,3,6,7,8)$ of case 3 marked in green. It does not contain an even alternating closed walk, because the closed walk starting at point 7 is of odd length.

The idea behind the Markov Draw algorithm is to construct randomly schlaufen. If the sum of the violation matrices is 0 we switch the edges in the closed walk of the schlaufen and the process is stopped. Otherwise, we stop the process with a certain probability or construct another schlaufe, which is edge-disjoint to the schlaufen already found. Such a process preserves the correctness of the graph and alters the graph randomly. After a sufficient number of process iterations, the distribution of the random graph approaches the uniform distribution (see section 2.2).

Markov Draw

Input: graph G, node-set-partition $\mathcal{P}$, a mixing parameter τ
Procedure:

```
1. set t := 0

2. choose A with probability $0 < q < 1$, otherwise B
   if case A:
       go to step 4
   else:
       go to step 3

3. find and mark schlaufe
   if sum of violation matrices is 0:
       switch the edges of the even closed alternating walk(s) of
       the schlaufe(n)
       set $G'$ to be the graph with the switched edges
       set $G := G'$
       and unmark edges
       go to step 4
   else:
       choose with probability $\frac{1}{2}$ case C or case D
       if case C:
           unmark edges and go to 4
       else:
           go to step 3

4. set t := t +1
   if $t = \tau$:
       return G
   else:
       go to step 2
```

Output: the graph G

The Markov Draw Algorithm uses as a subroutine find and mark schlaufe. This subroutine is described in the algorithm Schlaufen Detection. The Schlaufen Detection algorithm constructs a random alternating walk. In the construction of the alternating walk, we distinguish between an active visit of a node in the walk, where the next edge in the walk is in the graph, and an passive visit, where the next edge in the walk is not in the graph. At each node it chooses uniformly among the feasible out-edges.

Definition 3. *The set of* feasible out-edges *are*

- *in an active visit: the edges incident with node v in G, which are not already marked in a schlaufe.*
- *in a passive visit: the edges incident with node v in $\bar{G}$, which are not already marked in a schlaufe.*

Schlaufen Detection

Input: a graph G (which may have marked edges in it), node-set-partition $\mathcal{P}$
Procedure:

```
initialization:

  1 choose uniformly a node and go to step 2

active visit:

  2 mark node as active
    choose among all active feasible out-edges one
    if no choice is possible:
        go to step 6
    if the chosen edge is already in the walk:
        go to step 6
    else:
        go to step 3

  3 if new node is marked as passive:
        go to step 6
    else:
        go to step 4

passive visit:

  4 mark node as passive
    choose among all the passive feasible out-edges one
    if no choice is possible:
        got to step 6
    if the chosen edge is already in the walk:
        go to step 6
    else:
        go to step 3

  5 if new node is marked as active:
        go to step 6
    else:
        go to step 2

finalization:

  6 create violation matrix
    mark the edges in the created walk
    unmark all nodes
    return graph and violation matrix
```

Output: the graph and the violation matrix

Proposition 3. *Let $R = (v_1, ..., v_l)$ be a schlaufe in graph G. Let $r^a_G(v_i)$ be the cardinality of the set of feasible out-edges of node v_i in G for an active visit and $r^p_G(v_i)$ for a passive visit. The probability that the algorithm* Schlaufen Detection *marks schlaufe R is strictly positive and equal to:*

$$p_G(R) = \frac{1}{n} \prod_{i=1}^{l-1} \left(\frac{1}{r^a_G(v_i)} [i \mod 2] + \frac{1}{r^p_G(v_i)} [(i-1) \mod 2] \right). \tag{2}$$

Proof. The probability of a schlaufe R can be calculated as follows: in the initialization phase a node is chosen uniformly. The probability that v_1 is chosen equals to $\frac{1}{n}$. The edges are only marked at step 6 in the Schlaufen Detection algorithm. This implies that previous visits of nodes in the walk do not alter the set of feasible out-edges for node v_i. By the definition of a schlaufe, v_{i+1} is a node reachable via a feasible out-edge of v_i and is reached with probability $\frac{1}{r^a_G(v_i)}$ in an active visit and with probability $\frac{1}{r^p_G(v_i)}$ in a passive visit. When v_l is visited then there is no further feasible out-edge or a even alternating walk is closed or the edge $\{v_{l-1}, v_l\}$ is present in the alternating walk $(v_1, ..., v_{l-1})$, in all cases the algorithm Schlaufen Detection goes to step 6 and terminates. The probability of $R = (v_1, ..., v_l)$ is the product of the initializing probability and the probability of each step:

$$p_G(R) = \frac{1}{n} \prod_{i=1}^{l-1} \left(\frac{1}{r^a_G(v_i)} [i \mod 2] + \frac{1}{r^p_G(v_i)} [(i-1) \mod 2] \right).$$

The strict positivity follows directly because it is a product of strict positive numbers. □

If step 3 is reached, case B must have been chosen at step 2 with probability $1-q$. The third step of the Markov Draw Algorithm constructs a schlaufen-sequence $\mathcal{R} = (R_1, ..., R_k)$, which is switched if it has 0 violation. After each schlaufe found R_i, the corresponding schlaufe is marked in the graph. Let G_i be the graph with the schlaufen $(R_1, ..., R_{i-1})$ marked. The probability of finding schlaufe the R_i is $p_{G_i}(R_i)$. The total probability of $\mathcal{R}$ is strictly positive and equals to:

$$p_G(\mathcal{R}) = (1-q) \frac{1}{2^{(k-1)}} \prod_{i=1}^{k} p_{G_i}(R_i). \tag{3}$$

2.2 Correctness

2.2.1 Markov-Chains

In order to prove the correctness of the algorithm, we will use some properties of Markov-Chains. Let $(\Omega, \mathcal{A}, P)$ be a probability space and $(H, \mathcal{H})$ a measurable space.

Definition 4. *We call a sequence of random variables $(X_i), i \in \mathbb{N}$ a* Markov-Chain, *if $P(X_n = x_n | X_1 = x_1, ..., X_{n-1} = x_{n-1}) = P(X_n = x_n | X_{n-1} = x_{n-1})$*

for all $x_1, ..., x_n \in H$. And the transition probabilities $P(X_n = x'|X_{n-1} = x'')$ for $x', x'' \in H$ do not depend on n.

The sub-indices of the random variables are referred to as time-periods. If H is finite, it is called a discrete Markov-Chain and the elements of H can be enumerated $H = \{x^1, ..., x^{|H|}\}$. In the discrete case, the Markov-Chain can be represented by a directed graph with weighted arrows, which is called a state graph. The state graph is characterized as follows: the vertex set is set H, between two vertices x^i, $x^j \in H$ there is an arrow (x^i, x^j) if $P(X_n = x^j|X_{n-1} = x^i) > 0$, and it has the weight $P(X_n = x^j|X_{n-1} = x^i)$. Similar to the adjacency matrix for directed graphs, there is a matrix representation of the weighted directed graph M. Instead of an indicator for the existence of an arrow, the weight of the arrow (x^i, x^j) is the entry $M_{i,j}$.
We are interested in inquiring how the initial probability distribution of X_0 changes over time (X_t for $t \to \infty$). The distribution of X_0 can be represented as a vector p_0, with p_0^i equal to the probability of element x^i in period 0. The distribution vector p has only positive entries and they sum up to one.
The following proposition and corollary show how the initial distribution maps into the distribution of subsequent periods.

Proposition 4. *Let p_{n-1} be the distribution of X_{n-1}, then the distribution p_n of X_n is*

$$p_n = Mp_{n-1}.$$

Proof. We consider the probability p_n^i of the state i in period n. If the previous state is known to be x^j, then the probability of state x^i in period n is $P(X_n = x^i|X_{n-1} = x^j)$. The probabilities of the states in period $n-1$ are known. The probability of the event $X_{n-1} = x^j$ and $X_n = x^i$ can be calculated and is equal to $P(X_n = x^i|X_{n-1} = x^j)p_{n-1}^j$. The total probability of state i in period n is therefore the sum over all conditional probabilities weighted by the probability of the previous state

$$p_n^i = \sum_{j=1}^{|H|} P(X_n = x^i|X_{n-1} = x^j)p_{n-1}^j.$$

Summing over all elements gives:

$$\begin{aligned}
\sum_{i=1}^{|H|} p_n^i &= \sum_{i=1}^{|H|}\sum_{j=1}^{|H|} P(X_n = x^i | X_{n-1} = x^j) p_{n-1}^j \\
&= \sum_{j=1}^{|H|}\sum_{i=1}^{|H|} P(X_n = x^i | X_{n-1} = x^j) p_{n-1}^j \\
&= \sum_{j=1}^{|H|}\sum_{i=1}^{|H|} P(X_n = x^i | X_{n-1} = x^j) P(X_{n-1} = x^j) \\
&= \sum_{j=1}^{|H|} 1 P(X_{n-1} = x^j) \\
&= 1,
\end{aligned}$$

which shows that p_n is a distribution vector.

□

Corollary 1. *Let p_1 be the distribution of X_1, then the distribution p_n of X_n is*

$$p_n = M^{n-1} p_1$$

We say the state graph of a Markov-Chain has self-loops if every node has a directed self-loop. For the algorithm Markov Draw we are particularly interested in the case, where M is symmetric, the state graph has self-loops and is strongly connected.

Lemma 1. *Consider a Markov-Chain which is symmetric, strongly connected and has self-loops. For any initial distribution p_0, there is a $k \in \mathbb{N}$ such that p_k has only strictly positive entries.*

Proof. There is at least one state x^i which has strictly positive probability in the initial distribution vector, $p_0^i > 0$. $P(X_{n+1} = x^i | X_n = x^i)$ is strictly positive because the state graph has self-loops. The probability of p_n^i is at least $p_0^i P(X_{n+1} = x^i | X_n = x^i)^{n-1}$, which is strictly positive.
Let k be the length of the longest path between two vertices in the state graph. Since the state graph is strongly connected k is well defined. Consider an arbitrary state x^j. There is a path of $(x_i, ..., x_j)$ with length $m_j \leq k$. Therefore the probability of state x^j in period $h \geq m_j$ is at least

$$p_h^j = p_{h-m_j}^i \prod_{(x', x'') \text{ is edge in } (x_i, \ldots, x_j)} P(X_{n+1} = x'' | X_n = x'),$$

which is strictly positive. Note that the transition probabilities do not depend on the period n. The lemma holds, because the probability of state x^j is strictly positive for all h bigger then m_j and k is chosen such that it is bigger then m_j for every $j \in \{1, ..., |H|\}$. □

Definition 5. *We call a distribution vector for a Markov-Chain represented by M* stationary *if*

$$p = Mp.$$

Proposition 5. *Consider a Markov-Chain which is symmetric, strongly connected and has self-loops. There exists at most one stationary distribution vector.*

Proof. By definition a stationary distribution vector is an eigenvector of M with eigenvalue 1. Assume there are two distinct stationary distribution vectors p', p''. Then any linear combination of p' and p'' is also an eigenvector with eigenvalue 1. Because p' and p'' are both distribution vectors and not equal, $p' - p''$ has at least one positive and one negative entry. Therefore

$$k^* = \sup\{k \in \mathbb{R} | p' + k(p' - p'') > 0\}$$

is finite and we can set

$$p^* = p' + k^*(p' - p'').$$

p^* is an intersection point of the affine hull of p', p'' with the borders of the positive orthant. p^* is still a distribution vector, because its elements sum to 1, however p^* has one 0 entry. p^* is also an eigenvector of M with eigenvalue 1. In particular

$$p^* = p^*_k = M^k p^*$$

for any $k \in \mathbb{N}$. If we use p^* as initial distribution vector then there is no $k \in \mathbb{N}$ such that all entries of p^*_k are strictly positive. This contradicts lemma 1. □

Proposition 6. *Let M be the matrix of a Markov-Chain with self-loops. Every eigenvalue of M is in $(-1, 1]$.*

Proof. We denote the identity matrix I. The matrix M is stochastic, which means that the sum of the entries of each column is one. Assume there is an eigenvalue $|\lambda| > 1$, then the matrix $M - \lambda I$ is diagonal dominant. Therefore $M - \lambda I$ is invertible, which contradicts the assumption of λ being an eigenvalue. It remains to show that -1 is not an eigenvalue. Because of the self-loops all the diagonal entries of M are strictly positive. If λ is set equal to -1, $M - \lambda I = M + I$ is also diagonal dominant. □

It is possible to show that there is a stationary distribution and provide its explicit form. The vector with each element equal to one will be denoted with 1.

Proposition 7. *Consider a Markov-Chain which is symmetric, the vector $\frac{1}{n}1$ is a stationary distribution vector.*

Proof. The statement of the proposition is equivalent to the vector $\frac{1}{n}1$ being an eigenvector of M with eigenvalue 1. The column entries sum to one because they represent conditional probabilities. By symmetry, the sum of all the elements in a row sum to one. And therefore:

$$\tfrac{1}{n}1 = M(\tfrac{1}{n}1).$$

□

Theorem 1. *Consider a Markov-Chain which is symmetric, strongly connected and has self-loops. Then for any initial distribution p_0 it holds*

$$\forall \epsilon > 0 \;\; \exists T \in \mathbb{N} \;\; \forall \tau > T, \;\; \forall i \in \{1, ..., n\} : \;\; \frac{1}{n} - \epsilon < p_\tau^i < \frac{1}{n} + \epsilon.$$

Proof. Because M is symmetric, there is a basis of orthogonal eigenvectors $(v_1, ..., v_n)$, with $v_1 = \frac{1}{n}1$. Their eigenvalues are $(\lambda_1, \lambda_2, ..., \lambda_n)$ with $\lambda_1 = 1$. Without loss of generality we assume that λ_2 is the second largest eigenvalue in absolute terms. p_0 can be written as linear combinations of all eigenvectors.

$$p_0 = a_1 v_1 + \sum_{i=2}^{n} a_i v_i$$

By corollary 1 and the fact that $(v_1, ..., v_n)$ are orthogonal eigenvectors we can write the distribution vector in period τ as

$$p_\tau = a_1 \tfrac{1}{n} 1 + \sum_{i=2}^{n} \lambda_i^{\tau-1} a_i v_i.$$

By proposition 6 the eigenvalues $(\lambda_2, ..., \lambda_n)$ are all smaller then one in absolute terms, the summands $\lambda_i^{\tau-1} a_i v_i$ converge to 0 for $\tau \to \infty$. This implies that $a_1 = 1$.
Let q be the largest entry in all eigenvectors and a the absolute value of the largest coefficient a_i. For a given ϵ we set

$$T := -\frac{\log_{10}\left(\frac{naq}{\epsilon}\right)}{\log_{10}(|\lambda_2|)} + 1.$$

For $\tau > T$ we have as upper bound on p_τ

$$
\begin{aligned}
p_\tau &= \tfrac{1}{n}1 + \sum_{i=2}^{n} \lambda_i^{\tau-1} a_i v_i \\
&\leq \tfrac{1}{n}1 + \sum_{i=2}^{n} |\lambda_i|^{\tau-1} |a_i||v_i| \\
&\leq \tfrac{1}{n}1 + \sum_{i=2}^{n} |\lambda_2|^{T-1} |a_i||v_i| \\
&\leq \tfrac{1}{n}1 + \sum_{i=2}^{n} |\lambda_2|^{-\frac{\log_{10}\left(\frac{naq}{\epsilon}\right)}{\log_{10}(|\lambda_2|)}} |a_i||v_i| \\
&= \tfrac{1}{n}1 + \sum_{i=2}^{n} 10^{\log_{10}(|\lambda_2|)\frac{-\log_{10}\left(\frac{naq}{\epsilon}\right)}{\log_{10}(|\lambda_2|)}} |a_i||v_i| \\
&= \tfrac{1}{n}1 + \sum_{i=2}^{n} 10^{\log_{10}\left(\frac{\epsilon}{naq}\right)} |a_i||v_i| \\
&= \tfrac{1}{n}1 + \sum_{i=2}^{n} \frac{\epsilon}{naq} |a_i||v_i| \\
&= \tfrac{1}{n}1 + \sum_{i=2}^{n} \frac{\epsilon}{n}\frac{|a_i|}{a}\frac{1}{q} |v_i| \\
&\leq \tfrac{1}{n}1 + \sum_{i=2}^{n} \frac{\epsilon}{n} 1 \\
&\leq \tfrac{1}{n}1 + \epsilon 1.
\end{aligned}
$$

Reading the equation element wise gives the wanted upper bound $p_\tau^i < \frac{1}{n} + \epsilon$. The lower bound can be derived analogously. □

2.2.2 Uniform Draw

Definition 6. *We define the* state graph *of the* Markov Draw *as* $\Phi = (V_\Phi, A_\Phi)$. *Its underlying vertex set* V_Φ *is the set of all realizations of* (S, M). *For a realization* G, *we denote by* V_G *the corresponding vertex in* V_Φ. *It contains double arrows, the arrow set* A_Φ *is defined as follows:*

1. *For all vertices we set a directed loop* (V_G, V_G) *with probability* q.
2. *Let* G' *be another realization. For each feasible schlaufen-sequence* $\mathcal{R}$, *which edge-set of the even closed alternating walks is equal to* $E(G)\Delta E(G')$, *we set an arrow* $(V_G, V_{G'})$ *and assign the probability* $p_G(\mathcal{R})$.
3. *We set a directed loop* (V_G, V_G) *if the probability of all arrows leaving* V_G

according to point 1 or 2 do not sum to 1. Its probability is 1 minus the sum of the probability of all leaving arrows according to point 1 or 2.

The probability of any edge $a \in A_\phi$ we denote by $p(a)$.

Lemma 2. *For any two vertices $V_G, V_{G'}$ the transition probability $(V_G, V_{G'})$ in Φ is equal to the transition probability $(V_{G'}, V_G)$.*

Proof. Let $A_{G,G'}$ be the set of edges from the vertex V_G to the vertex $V_{G'}$. We construct a bijection $\varphi : A_{G,G'} \to A_{G',G}$. Then we show that the probability of an edge $p(a)$ is equal to $p(\varphi(a))$. If that is proven, the probability of a transition from V_G to $V_{G'}$ can be transformed in the following manner:

$$\begin{aligned} \sum_{a \in A_{G,G'}} p(a) &= \sum_{a \in A_{G,G'}} p(\varphi(a)) \\ &= \sum_{\varphi^{-1}(a') \in A_{G,G'}} p(a') \\ &= \sum_{a' \in \varphi(A_{G,G'})} p(a') \\ &= \sum_{a' \in A_{G',G}} p(a') \end{aligned}$$

which is the probability for a transition from $V_{G'}$ to V_G.

For the construction of the bijection, consider that each edge $A_{G,G'}$ corresponds uniquely to a feasible schlaufen-sequence $\mathcal{R} = (R_1, ..., R_k)$. If $R_i = (v_1, ..., v_i, ..., v_l)$ is a schlaufe of case 1 with v_i as closed alternating walk start, we define $\bar{R}_i := (v_1, ..., v_{i-1}, v_l, v_{l-1}, ..., v_{i+1}, v_i)$. Note that by definition $v_i = v_l$. If R_i is a schlaufe of case 2 or case 3, we set $\bar{R}_i := R_i$. We define $\bar{\mathcal{R}} := (\bar{R}_1, ..., \bar{R}_k)$.

Note that the R_i, $i \in \{1, ..., k\}$ are edge disjoint. As soon as the closed walk of $(R_1, ..., R_i)$ are switched, $\bar{R}_i$ is a schlaufe in the graph obtained by the switching. The violation matrix of $\bar{R}_i$ is the negative violation matrix of R_i. This implies that if $\mathcal{R}$ is a feasible schlaufen-sequence for G, which defines an edge in $A_{G,G'}$, then $\bar{\mathcal{R}}$ is a feasible schlaufen-sequence for G' and defines an edge $A_{G',G}$.

We now define φ as the function, which maps an edge in $A_{G,G'}$ with schlaufen-sequence $\mathcal{R}$ to the edge in $A_{G',G}$ with schlaufen-sequence $\bar{\mathcal{R}}$. By construction, φ is injective, which implies $|A_{G,G'}| \leq |A_{G',G}|$. By symmetry, we conclude that $|A_{G',G}| \leq |A_{G,G'}|$, which implies $|A_{G',G}| = |A_{G',G}|$ and that φ is bijective.

It remains to show that the probability of an edge $p(a)$ is equal to $p(\varphi(a))$. Let $\mathcal{R} = (R_1, ..., R_k)$ be the feasible schlaufen-sequence corresponding to a. For every node, there are equally many feasible active / passive out-edges in G as in G'. This can be seen as follows, with no marked edges the number of feasible out-edges is directly derived from the degree of the node. Recall that with G_i we denote the graph with the schlaufen $(R_1, ..., R_{i-1})$ marked and with G'_i the graph with the schlaufen $(\bar{R}_1, ..., \bar{R}_{i-1})$. If for a node v, one (active or passive) edge is marked due to an edge in R_i, then one (active or passive) edge for node v is marked due to an edge in $\bar{R}_i$. Therefore, we conclude that for every node there are equally many feasible active / passive out-edges in G_i as in G'_i for all $i \in \{1, ..., k\}$. Therefore, $r^a_{G'_i}(v)$ is equal to $r^a_{G_i}(v)$ and $r^p_{G'_i}(v)$ is equal to $r^p_{G_i}(v)$ for $i \in \{1, ..., k\}$. Looking at equation (2), the $p_G(R_i)$ is only different from $p_{G'}(\bar{R}_i)$ with respect to the numbering of the factors. But in a closed walk of a schlaufe the start node v_i and the end node v_l are such that $i - l \mod 2 = 0$. The reordering leaves even indices even and odd indices odd. Therefore, $p_G(R_i)$ $= p_{G'}(\bar{R}_i)$. From equation (3), it follows directly that $p_G(\mathcal{R}) = p_{G'}(\bar{\mathcal{R}})$, which completes the proof.

□

Lemma 3. *The state graph Φ is strongly connected.*

Proof. The symmetric difference of two realizations of (S, M), which we denote by G and G', is a set of alternating closed walks. Alternating closed walks are in particular schlaufen (of case 1). We order the alternating schlaufen $\{R_1, ..., R_k\}$ arbitrarily to obtain the sequence $(R_1, ..., R_k)$. The sum of the violation matrices is 0. Therefore, $(R_1, ..., R_k)$ is either a feasible schlaufen-sequence or a concatenation of feasible schlaufen-sequences. In the first case, there is an edge from V_G to $V_{G'}$. In the second case, all the feasible schlaufen-sequences define an edge to a new vertex, resulting in a directed path starting at V_G and ending in $V_{G'}$. Thus between any two vertices in Φ there is a directed path.

□

Theorem 2. *The algorithm* Markov Draw *is a random walk on the state graph Φ which samples uniformly from the set all realizations of (S, M) for $\tau \to \infty$.*

Proof. Every time the algorithm arrives at step 4, a new edge of the state graph is crossed. If at step 2 A is chosen, it follows a loop edge of type 1 with probability q. Else, it proceeds to step 3. In step 3, a schlaufen-sequence $\mathcal{R}$ is constructed. The probability with which $\mathcal{R}$ is constructed in the algorithm Schlaufen Detection is given by equation (3). If the violation matrices sum up to 0, the closed walks are switched and an edge of type 2 is followed with probability $p_G(\mathcal{R})$. If the violation matrices do not sum up to 0, then an edge of type 3 is followed. The probability of the cases of type 3 equals to 1 minus the probability of the cases of type 1 and of type 2. Therefore algorithm 1 is a random walk on the state graph Φ.
According to lemma 2, Φ is (weighted) symmetric and according to lemma 3, it is strongly connected. Φ has self-loops. According to theorem 1, the limit distribution is uniform. □

2.3 Hard Constraints

Erdős et al. inquired in "Graph Realizations Constrained by Skeleton Graphs" the PAM-realization problem for the case that the partition restriction have special topological characteristics. The crucial idea in dealing with the PAM-realization problem is to consider hard constraints. Hard constraints are 0 entries in the PAM. The PAM can be interpreted as an adjacency matrix, where an edge exists for each nonzero entry. In this way they define a skeleton graph, which captures the topology of the constraints [18].

In practice, hard constrained partitions occur often. The most prominent examples are k-partite graphs, where there are no edges within the k edge groups. The case $k = 2$ corresponds to a bipartite graph. Another example are graphs with node groups corresponding to geographical areas. If only nodes belonging to neighboring areas can be connected, then the resulting partition adjacency matrix contains many hard constraints.

Apart from the edges not possible due to the hard partition constraints, it is also clear that an alerting path never passes a node with degree 0 or a node with degree $|V| - 1$. This is the case because the alternating walk contains for every node except the start node one edge in G and one edge in $\bar{G}$. This observation motivates the definition of forbidden edges.

Definition 7. *Let (S, M) be the degree sequence and the partition adjacency matrix to a node set V . Let*

$$E = \{\{v', v''\}|v', v'' \in V, v' \neq v''\}$$

be the set of possible edges. We call an edge $e \in E$ forbidden *if and only if*

- *e is incident to a node v with degree 0 or degree $|V| - 1$.*
- *e is incident to two nodes, v' belonging to node group i and v'' belonging to node group j, which are hard constrained, $M_{i,j} = 0$.*

Proposition 8. *A forbidden edge occurs either in all the realizations of (S, M) or occurs in no realization of (S, M)*

Proof. Let e be a forbidden edge.

Case 1: e is incident to a node v with degree 0. The node v has degree 0 in all the realizations of (S, M). Therefore, e it cannot occur in any realization.

Case 2 : e is incident with a node v with degree $|V| - 1$. The node v is adjacent to all other vertices in V. Because the degree of v is the same in all the realizations of (S, M), v is incident to all other vertices in all the realization. In particular the edge e occurs in all the realizations.

Case 3: e is incident to two nodes, v' belonging to node group i and v'' belonging to node group j, which are hard constrained, $M_{i,j} = 0$. Then e cannot occur in any realization of (S, M) because it would violate the partition constraints. □

The above algorithm deals with forbidden edges by rejecting to switch schlaufen containing them or by creating schlaufen of case 2 with no closed alternating walk. In a partition with many hard constraints or in a graph with many isolated nodes, the probability that a schlaufe contains a forbidden edge is high. The forbidden edges are known from the start of the algorithm. It is sensible to consider them at construction time. This is done by adjusting the set of feasible out nodes.

Definition 8. *The set of* strictly feasible out-edges *is equal to the set difference of the set of feasible out-edges according to definition 3 minus the set of forbidden edges.*

We now adjust the Schlaufen Detection by replacing the feasible out-edges in step 2 and 4 with the strongly feasible out-edges. We call the new schlaufen detection algorithm Improved Schlaufen Detection.

Proposition 9. *Let $R = (v_1, ..., v_l)$ be a schlaufe in graph G. Let $\tilde{r}^a_G(v_i)$ be the cardinality of the set of strictly feasible out-edges of node v_i in G for an active visit and $\tilde{r}^p_G(v_i)$ for a passive visited. The probability that the algorithm* Improved Schlaufen Detection *marks schlaufe R is equal to:*

$$p_G(R) = \begin{cases} 0 & \textit{if } R \textit{ contains a forbidden edge} \\ \frac{1}{n}\prod_{i=1}^{l-1}\left(\frac{1}{\tilde{r}^a_G(v_i)}[i \mod 2] + \frac{1}{\tilde{r}^p_G(v_i)}[(i-1) \mod 2]\right) & \textit{else} \end{cases} \tag{4}$$

Proof. **Case R contains a forbidden edge:**
Let the forbidden edge be $\{v_i, v_{i+1}\}$. When the node v_i is reached by the algorithm Improved Schlaufen Detection, $\{v_i, v_{i+1}\}$ is not among the strictly feasible out-edges. Therefore the probability that v_{i+1} comes after v_i in the schlaufen construction is 0.

Case R does not contain a forbidden edge:
The probability of a schlaufe R can be determined similarly to the proof of proposition 3 : in the initialization phase a node is chosen uniformly. The probability that v_1 is chosen is $\frac{1}{n}$.
The edges are only marked at step 6 in the Schlaufen Detection algorithm. Additionally, because of proposition 8, the set of forbidden edges never changes during the algorithm. This implies that previous visits of nodes in the walk do not alter the set of feasible out-edges for node v_i. By the definition of a schlaufe and the fact that the schlaufe does not contain a forbidden edge, v_{i+1} is a node

reachable via a strictly feasible out-edge. v_{i+1} is reached with probability $\frac{1}{\tilde{r}^a_G(v_i)}$ in an active visit and with probability $\frac{1}{\tilde{r}^p_G(v_i)}$ in a passive visit. When v_l is visited then there is no further strictly feasible out-node or an even alternating walk is closed or the edge $\{v_{l-1}, v_l\}$ is present in the alternating walk $(v_1, ..., v_{l-1})$, in all cases the algorithm Improved Schlaufen Detection goes to step 6 and terminates. The probability of $R = (v_1, ..., v_l)$ is the product of the initializing probability and the probability of each step:

$$\tilde{p}_G(R) = \frac{1}{n} \prod_{i=1}^{l-1} \left(\frac{1}{\tilde{r}^a_G(v_i)} [i \mod 2] + \frac{1}{\tilde{r}^p_G(v_i)} [(i-1) \mod 2] \right).$$

□

The probability of a schlaufen-sequence $\mathcal{R} = (R_1, ..., R_k)$ is:

$$\tilde{p}_G(\mathcal{R}) = (1-q) \frac{1}{2^{(k-1)}} \prod_{i=1}^{k} \tilde{p}_{G_i}(R_i). \tag{5}$$

For every schlaufen-sequence $\mathcal{R}$ the probability $p_G(\mathcal{R})$ is strictly positive. However, this is not the case with $\tilde{p}_G(\mathcal{R})$.

Definition 9. *We call a sequence of schlaufen $\mathcal{R} = (R_1, ..., R_k)$* strictly feasible *if no schlaufe contains a forbidden edge, the schlaufen are edge-disjoint, the sum of the violation matrices is zero and for $i < k$ the sum of the violation matrices is not zero.*

Proposition 10. *For each strictly feasible schlaufen-sequence $\mathcal{R} = (R_1, ..., R_k)$ the probability $\tilde{p}_G(\mathcal{R})$ is strictly positive.*

Proof. Due to equation (5) it is sufficient to show that for each schlaufe R_i, $i \in \{1, ..., k\}$ the value $\tilde{p}_{G_i}(R_i)$ is strictly positive. Because the schlaufen-sequence is strictly feasible every schlaufe $R_i = (v_1, ..., v_l)$ does not contain any forbidden edge. Because of proposition 9, $\tilde{p}_{G_i}(R_i)$ is equal to

$$\frac{1}{n} \prod_{j=1}^{l-1} \left(\frac{1}{\tilde{r}^a_{G_i}(v_j)} [j \mod 2] + \frac{1}{\tilde{r}^p_{G_i}(v_j)} [(j-1) \mod 2] \right),$$

which is a product of strictly positive values.

□

We can now define

Definition 10. *We define the* state graph *of the of the* Markov Draw*, which uses the subroutine* Improved Schlaufen Detection *as $\tilde{\Phi} = (V_{\tilde{\Phi}}, A_{\tilde{\Phi}})$. Its underlying vertex set $V_{\tilde{\Phi}}$ is the set of all realizations of (S, M). For a realization G, we denote by V_G the corresponding vertex in $V_{\tilde{\Phi}}$. It contains double arrows, the arrow set A_Φ is defined as follows:*

1. *For all vertices we set a directed loop (V_G, V_G) with probability q.*
2. *Let G' be another realization. For each feasible schlaufen-sequence $\mathcal{R}$, which circle edge-set is equal to $G \Delta G'$ we set an arrow $(V_G, V_{G'})$ and assign the probability $\tilde{p}_G(\mathcal{R})$.*
3. *We set a directed loop (V_G, V_G) if the probability of all arrows leaving V_G according to point 1 or 2 do not sum to 1. Its probability is 1 minus the sum of the probability of all leaving arrows according to point 1 or 2.*

The probability of any edge $a \in A_{\tilde{\Phi}}$ we denote by $\tilde{p}(a)$. Note the only difference of Φ and $\tilde{\Phi}$ are the probabilities of the edges.

Lemma 4. *For any two vertices $V_G, V_{G'}$ the transition probability $(V_G, V_{G'})$ in $\tilde{\Phi}$ is equal to the transition probability $(V_{G'}, V_G)$.*

Proof. Let $A_{G,G'}$ be the set of edges from the vertex V_G to the vertex $V_{G'}$. Because the vertex set of Φ and $\tilde{\Phi}$ is the same, the existence of a bijection $\varphi : A_{G,G'} \rightarrow A_{G',G}$ follows analogically to the proof of lemma 2. We show that the probability of an edge $\tilde{p}(a)$ is equal to $\tilde{p}(\varphi(a))$.

For any node, there are equally many feasible active / passive out-edges in G as in G'. Because the set of forbidden edges doesn't change, there are equally many forbidden active/passive out-edges and therefore equally many strictly feasible out-edges in G as in G'. If for a node v, one to v incident edge is marked due to an edge in R_i, then, one to v incident edge is marked due to an edge in $\bar{R}_i$. Therefore, $\tilde{r}^a_{G'_i}(v)$ is equal to $\tilde{r}^a_{G_i}(v)$ and $\tilde{r}^p_{G'_i}(v)$ is equal to $\tilde{r}^p_{G_i}(v)$. Looking at equation (4), the $\tilde{p}_{G_i}(R_i)$ is only different from $\tilde{p}_{G'_i}(\bar{R}_i)$ with respect to the numbering of the factors. But in a closed walk of a schlaufe the start node v_i and the end node v_l are such that $i - l \mod 2 = 0$. The reordering leaves even indices even and odd indices odd. Therefore, $\tilde{p}_G(R_i) = \tilde{p}_{G'}(\bar{R}_i)$. From equation (5), it follows directly that $\tilde{p}_G(\mathcal{R}) = \tilde{p}_{G'}(\bar{\mathcal{R}})$. The equation

$$\sum_{a \in A_{G',G}} \tilde{p}(a) = \sum_{a' \in A_{G,G'}} \tilde{p}(a')$$

follows also analogically as in the proof of lemma 2. □

Lemma 5. *The state graph $\tilde{\Phi}$ is strongly connected.*

Proof. The proof is analogous to the proof of lemma 3. However we have to show that the schlaufen $(R_1, ..., R_k)$ in the symmetric difference of G and G' all have a positive probability. The schlaufen $(R_1, ..., R_k)$ can be split into a concatenation of schlaufen-sequences. By proposition 10 we have to show that the schlaufen-sequences are all strictly feasible. By construction all edges in all the Schalufen are switched. By proposition 8, forbidden edges cannot occur in exactly one graph of G' and G. Therefore all schlaufen don't contain a forbidden edge, which concludes the proof. □

Theorem 3. *Algorithm* Markov Draw*, which uses the subroutine* Improved Schlaufen Detection *is a random walk on the state Graph $\tilde{\Phi}$ which samples uniformly a feasible graph for $\tau \to \infty$.*

Proof. The proof is analogous to the proof theorem 2. The symmetry and strong conceitedness is ensured by lemma 4 and lemma 5. □

2.4 Discussion

Other approaches have been developed in order to draw out of a set with more complex constraints. Most notable are the k-switching procedure from Tabourier [19]. The idea is to choose k edges at random, choose a random permutation which defines how the edges have to be switched. Then check if the new graph fulfills the double-edge, self-loops and complex constraints. If it fulfilled the constraints, the edges are swapped. This approach has two disadvantages:

For any two graphs fulfilling the constraints, there is a k, such that one can be transformed into another by an edge-swap of k edges. However, this does not imply connectivity of the Markov-Chain graph. It is not clear how to choose the k. To illustrate this problem two example are given. The first one is given in figure 2 . It has nodes $\{1, ..., 8\}$ all with degree 1, 4 edges and as additional constraint no edge from the group $\{1, ..., 4\}$ to $\{5, ..., 8\}$. There are 4 graphs satisfying this constraints. With the choice of $k = 2$, the Markov-Chain can reach all of them. With a four edges switch, the Markov-Chain always performs a switch in both groups and therefore, can reach only 2 of the 4 feasible graphs. This problem can be overcome by randomly choosing subsamples of the k-edges, which are to be swapped.

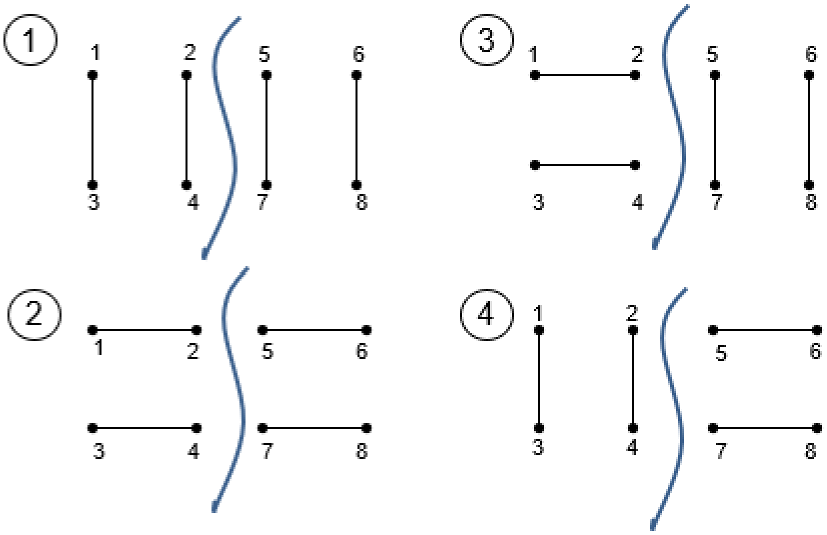


Figure 2: Example graph: The four realizations of the first example-graph. If 4 edges are swapped in one step, it is not possible to reach all the examples.

The second example has for the node-set $\{0, ..., 2n-1\}$ where the nodes 0 to $n-1$ have degree 3 and the nodes n to $2n-1$ have degree 1. We define n groups. In group i are all the nodes j for which $j \equiv i \mod n$. The group constraints are: the number of cross-edges between two groups is 2 if $i - j \equiv 1 \mod n$. Any realization of this graph has no edges between nodes from $\{n, ..., 2n-1\}$, because adding an edge to these two nodes would isolate them from the rest of the graph and the degree-sequence of the residual graph would not be graphical anymore. Every node of $\{1, ..., n-1\}$ has to be adjacent to at least one node of $\{n, ..., 2n-1\}$. By the pigeonhole principle, it follows that every node of $\{1, ..., n-1\}$ has to be adjacent to exactly one node of $\{n, ..., 2n-1\}$ and therefore any realization has to have the edge-set $A = \{\{i, i+1 \mod n\} | i \in \{0, ..., n-1\}\}$. Every node in $\{0, ..., n-1\}$ has to be connected with one node of $\{n, ..., 2n-1\}$. Due to the double arrow and partition restrictions, there is only one node of $\{n, ..., 2n-1\}$ feasible in each one of the two neighboring groups. If one is chosen, then the others are implied by this choice. Therefore, there are only two options for the other edges either $B = \{\{i, (i+1 \mod n) + n\} | i \in \{0, ..., n-1\}\}$ or $B = \{\{i, (i-1 \mod n) + n\} | i \in \{0, ..., n-1\}\}$. Figure 3 shows the two possible realizations.

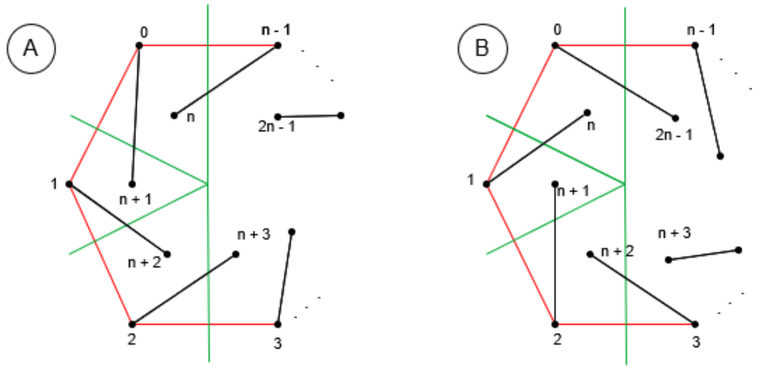


Figure 3: Example graph: The two realizations of the second example-graph. In order to reach the other realization by a swap, n edges have to be swapped simultaneously.

The symmetric difference of the two possible realization has $2n$ edges. For the k-switch algorithm to work for this case, k = n has to be chosen in advance. The proposed algorithm solves this problem by not having to define in advance how many edges have to be swapped.

Another problem, which the k-switching procedure has, is that for high k, there is a high rejection of constraints. As Tabourier puts it:

> *"The reason why large values of k are not necessarily advisable actually lies in the possibility of k-switch failures, i.e. such that the resulting graph does not anymore belong to the set of feasible graphs and thus the walk stays on the same graph at the next step. Odds of such failure depend in a complicated way on k.[..] In practice, given an a priori fixed number of trials, we observe that the number of successful alterations tends to decrease sharply for large values of k."* [2]

This is in particular a problem when the graph is dense, due to the double-arrow restriction. The example above showed a case, where for correctness, k had to be chosen as half of the edge-set. In a practical example, choosing k as half of the edge-set will almost never find a possible switch. The proposed algorithm alleviates this problem, because by construction of the switching edge set, there is neither double arrow nor loop violation. Further partition restriction which are 0 can be considered while constructing the schlaufen, which is particular interesting when looking at graphs with hard constraint such as bipartite graphs or the graph in figure 3.

A common problem, which all the Markov-Chain sampling methods face, is how long it is necessary to run the algorithm until the generated sample is from a uniform distribution. In other words, how fast is the Markov-Chain mixing.

[2] from section 2.4 in [19]

This is for the general edge-swap algorithm an open problem. There are mixing results for regular graphs [20], which where subsequently amplified to a broader class of graphs [21]. The mixing properties of the proposed algorithm are to be inquired for future work.

2.5 Implementation

The package *Uniform Graph Draw* implements the Markov Draw algorithm and provides a simple API. It is written in Python. The source code is openly available under the MIT Licence at:

> `https://github.com/AndrinPelican/ugd`.

A documentation on the API, architecture, and testing is included. The package is published on the Python Package Index (PyPI):

> `https://pypi.org/project/ugd/`

and is installable via pip.

3 The Realization Problem

The algorithm Markov Draw is able to draw uniformly out of the feasible set $\mathcal{G}$ if an element of $\mathcal{G}$ is provided. If one is interested in inquiring a specific observed graph, then the observed graph is the element for the input to the algorithm. Otherwise a natural question is whether such an element exists and how to construct it. We first provide a necessary condition for the existence of a graph. We then discuss the steps and challenges of extending it to a sufficient condition.

3.1 A Necessary Condition

Consider the node-set $\{1, ..., n\}$, the degree-sequence S, the node groups V_i and the partition matrix $M \in \mathbb{N}^{|\mathcal{P}|\times|\mathcal{P}|}$. Assume there exists a realization G of (S, M). The nodes in group V_1 fulfill the degree sequence. For each other node group V_j, $j \in \{2, ..., |\mathcal{P}|\}$ there are $M_{1,j}$ edges from V_1 to V_j. Each node in V_1 is connected to at most $|V_j|$ nodes in V_j, since otherwise a double edge constraint is violated. We now derive a multigraph H^1 from G by shrinking the node-groups $\{V_2, ..., V_{|\mathcal{P}|}\}$. This gives the motivation for the following definition:

Definition 11. *For a graph G satisfying (S, M) we define for $i = 1, ..., |\mathcal{P}|$ the to i* corresponding multigraph *$H^i = (V, E)$. The node-set V is equal to $V_i \cup \{V_1, ..., V_{i-1}, V_{i+1}, ..., V_{|\mathcal{P}|}\}$. The nodes $\{V_1, ..., V_{i-1}, V_{i+1}, ..., V_{|\mathcal{P}|}\}$ are referred to as outer nodes. For each edge e in G that is incident with an node* *in V_i we create an edge in H^i as follows. If e is incident to two nodes in V_i, it is incident to the same two nodes in H^i. If e is incident with a node in V_i and a node in group V_j, it is in H^i incident with the same node in group V_i and it is incident to the node V_j.*

The next definition clarifies how the to i corresponding multigraph relates to the (S, M) restrictions.

Definition 12. *We call the to i corresponding multigraph H^i* feasible *if the following condition are fulfilled. There are no edges between $\{V_1, ..., V_{i-1}, V_{i+1}, ..., V_{|\mathcal{P}|}\}$ and no parallel edges within V_i. For node V_j and a node in V_i there are at most $|V_j|$ parallel edges. V_j has degree $M_{1,j}$.*

The to i corresponding feasible multigraph is therefore characterized by the degree-sequence of S restricted on V_i and the tuples $(a_j, m_j) = (M_{i,j}, |V_j|)$ for $j \in \{1, ..., i-1, i+1, ..., |\mathcal{P}|\}$.

Proposition 11. *If there exists a graph G satisfying (S, M), then for each $i = 1, ..., |\mathcal{P}|$ there exists a to i corresponding feasible multigraph $H^i = (V, E)$.*

Proof. Definition 11 gives an explicit construction from G to a to i corresponding multigraph H^i. By definition there are no edges between the nodes $\{V_1, ..., V_{i-1}, V_{i+1}, ..., V_{|\mathcal{P}|}\}$, no parallel edges within V_i. The nodes in V_i have degrees of the restricted degree-sequence S^{V_i}. And the outer nodes $V_j \in \{V_1, ..., V_{i-1}, V_{i+1}, ..., V_{|\mathcal{P}|}\}$ have a_j incident edges and at most m_j parallel edges.
Therefore there exists a to i corresponding feasible multigraph H^i. □

Corollary 2. *If it is not possible to construct a feasible multigraph H^i for each node-group V_i, then there is no realization of (S, M).*

Corollary 2 delivers a necessary condition for the existence of a realization. For the construction of H^i it is convenient to consider another type of restricted, called *feasible extended graph F^i*.

Definition 13. *We define a* to i corresponding feasible extended graph *$F^i = (V, A)$. The node set V is equal to $\bigcup_{k=1}^{|\mathcal{P}|} V_k$. The nodes are enumerated by their respective group V_k, $k \in \{1, ..., |\mathcal{P}|\}$. In particular, v_j^k is the node number j of group V_k. The nodes v_j^k, $k \neq i$ are referred to as outer nodes.*
The degree of v_j^i equals the corresponding degree in S. For the calculation of the degree of v_j^k, $k \neq i$ we determine $b \in \mathbb{N}$ and $r < m_k$ such that $a_k = bm_k + r$. If $k \leq r$ the degree of v_j^k is $b + 1$. Otherwise the degree is b. Further it is required that no two outer nodes are adjacent.

The relationship between the feasibility of a multigraph and the feasibility of an extended graph is given by the following lemma.

Lemma 6. *There exists a to i corresponding feasible extended graph if and only if there exists a to i corresponding feasible multigraph.*

Proof. $\Rightarrow$:
There exists a to i corresponding feasible multigraph H^i. Consider the outer node V_k. We enumerate the incident edges. For the enumeration we start with an edge, which is among the ones with the most parallel edges. Then we enumerate all its parallel edges. This procedure is iterated until all the edges incident with V_k are enumerated. Consider the m_k nodes $v_1^k, ..., v_{m_k}^k$ in F^i. For edge h leave the node in V_i unaltered and replace V_k with node $v_{h \mod m_k}^k$. There will be no double arrows because in the multigraph the node V_k has at most m_k parallel edges.
This can be done for all outer nodes in H^i. The edges in H^i with only incident nodes in V_i map in a natural manner to edges in F^i with only nodes in V_i.
This gives a feasible realization of F^i.
$\Leftarrow$:
We start with the to i corresponding feasible extended graph. Since in each outer node-group V_k there are at most m_k nodes, contracting the nodes in the node-group k to a single node will lead to a multigraph with node k having at most m_k parallel edges. □

3.1.1 The Extended Havel-Hakimi Algorithm

We are now able to define the Extended Havel-Hakimi. It can be applied to each node group. The nodes in the node-group, which it is applied to, are called inner nodes. The restriction of the degree sequence S on the inner nodes is refereed to a S^{in}.

Extended Havel-Hakimi:

Input: The degree sequence of the inner nodes S^{in}, the sequence $(a_1, m_1), ..., (a_{|\mathcal{P}|}, m_{|\mathcal{P}|})$ of tuples with the degree of the outer nodes and maximal parallel edges.
Procedure:
Outer node expansion phase

```
1. Start an empty degree series, it is called the outer degree sequence
   $S^{out}$

2. For each outer node tuple $(a_k, m_k)$
       Calculate $r, b$ such that $a_k = bm_k + r$, with $b \in \mathbb{N}$ and $r < m_k$
       Add $m_k - r$ entries to the outer degree sequence with value $b$
       Add $r$ entries to the outer degree sequence with value $b + 1$
```

Cross-edge addition phase

```
1. Start an empty graph with $length(S^{in}) + length(S^{out})$ nodes.

2. For each outer node $v_{out}$:
       For i in $degree(v_{out})$:
               Find inner node $v_{in}$ with highest residual degree not
               adjacent to $v_{out}$
           If degree of $v_{in}$ is 0:
               Return FALSE
           Else:
               Add the edge $\{v_{out}, v_{in}\}$ to the graph
```

Havel-Hakimi phase

```
1. Add inner edges to the graph using the Havel-Hakimi algorithm on
   the residual inner degree-sequence.
   If possible:
       Return TRUE and the realization
   Else:
       Return FALSE
```

Output: TRUE and the realization/FALSE.

Theorem 4. *The algorithm* Extended Havel-Hakimi*, returns TRUE if and only if there exists a feasible multigraph.*

Proof. The phase *outer node expansion phase* transforms the constraints of the sequence $(a_1, m_1), ..., (a_{|\mathcal{P}|}, m_{|\mathcal{P}|})$ of tuples with the degree of the outer nodes and the maximal parallel edges to the constraints of the extended graph.
Assume: the *cross-edge addition phase* and *Havel-Hakimi phase* construct a realization of the extended graph if and only if a feasible extended graph exists. Then the algorithm returns TRUE if and only if there exists a feasible extended graph. Lemma 6 ensures that the algorithm Extended Havel-Hakimi, returns TRUE if and only if there exists a feasible multigraph.

It remains to show that: The *cross-edge addition phase* and *Havel-Hakimi phase* construct a realization of the extended graph if and only if a feasible extended graph exists.
$\Leftarrow$:
If the algorithm returns TRUE, then it also delivers a realization of the extended graph. During the *cross-edge addition phase* an edge is only added if it does not form a double edge and the residual degrees of the two nodes are not 0. Further no edges between outer nodes can be added. In the *Havel-Hakimi phase* only edges between inner nodes are added, which do not form double edges or loops and are incident to nodes with residual degree bigger then 0. Therefore the constructed realization of the extended graph is feasible.
$\Rightarrow$:
We assume the algorithm returns FALSE, despite the existence of a correct graph.
Let $(e_1, e_2, ..., e_h)$ be the edges added by the algorithm until it returns FALSE. Let $\mathcal{G}_i$ for $0 \leq i \leq h$ be the set of admissible extended graphs sharing the first i edges of $(e_1, e_2, ..., e_h)$. By assumption $\mathcal{G}_0 \neq \emptyset$. Since the algorithm returns FALSE, $\mathcal{G}_h$ is empty. Therefore there exists a unique index $0 \leq j < h$, such that $\mathcal{G}_j \neq \emptyset$ and $\mathcal{G}_{j+1} = \emptyset$.

case 1: e_{j+1} is added in the *outer node expansion phase.*
Let G be a graph in $\mathcal{G}_j$. We construct from G a graph $\tilde{G} \in \mathcal{G}_{j+1}$ leading to the desired contradiction. e_{j+1} has the shape $\{o, s\}$ and let $\{o, \tilde{s}\}$ be an edge in G, which is not in $(e_1, ..., e_j)$.
By construction of the algorithm the residual degree of $\tilde{s}$ is smaller or equal to the residual degree of s. Since $\{o, \tilde{s}\}$ is in G, there is at least one edge $\{k, s\}$, which is not in $(e_1, ..., e_j)$, such that k is not adjacent (and not equal) to $\tilde{s}$ in G.
$\{k, s\}, \{o, \tilde{s}\}$ are in G and $\{k, \tilde{s}\}, \{o, s\}$ are not in G. We define $\tilde{G}$ by deleting $\{k, s\}, \{o, \tilde{s}\}$ from G and adding $\{k, \tilde{s}\}, \{o, s\}$ to G. The nodes s, $\tilde{s}$ are both inner-nodes. Therefore independent of the node-group of o and k there are equally many cross-group edges in G and $\tilde{G}$. $\tilde{G}$ is still admissible, because the degree-sequence is the same as of G and there are no double edges nor self loops in $\tilde{G}$. However $e_{j+1} = \{o, s\}$ is in $\tilde{G}$, hence $\tilde{G} \in \mathcal{G}_{j+1}$. A contradiction.

case 2: e_{j+1} is added in the *Havel-Hakimi phase.*
Let e_{l+1} be the first edge added in the *Havel-Hakimi phase.* When e_{l+1} is added only inner-nodes have a positive residual degree. Because $l < j$ there exists a feasible graph $G \in \mathcal{G}_l$ sharing all the edges, which were added up to e_l. Because G satisfies the degree-sequence, the residual inner degree-sequence is graphical. Due to the correctness of the Havel-Hakimi algorithm the Extended Havel-Hakimi algorithm would return TRUE. A contradiction.

□

3.1.2 Examples

In order to illustrate the above derived procedure we give two examples. The first is a positive example, where there exists a realization of the graph with partition constraints. The second shows for a given partition constraint that there exists no realization of a graph fulfilling the constraint.

First example:
The restrictions of the graph are:

- Degree sequence: $S = (3, 1, 1, 3, 1, 1)$.
- Partitions: $\mathcal{P} = \{V_1, V_2, V_3\}$ with $V_1 = \{1, 2\}$, $V_2 = \{3, 4\}$ and $V_3 = \{5, 6\}$.
- Partition matrix:
$$M = \begin{pmatrix} 0 & 3 & 1 \\ 3 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

In order to check the necessary condition we construct for each node group V_i a feasible multigraph H^i. For the illustration we give the conditions for the feasible multigraph H^1:

- Degree sequence: $S^{V_1} = (3, 1)$.
- The outer edge tuple for V_2 is $(3, 2)$ and for V_3 is $(1, 2)$.
- No edges between outer nodes are allowed.

The conditions for the feasible extended graph F^1 are:

- Degree sequence: $S = (3, 1, 2, 1, 1, 0)$, were the entries $3-6$ belong to outer nodes.
- No edges between the outer nodes are allowed.

The figure 4 shows example realization of G, H^1 and F^1. The algorithm Extended Havel-Hakimi would construct F^1.

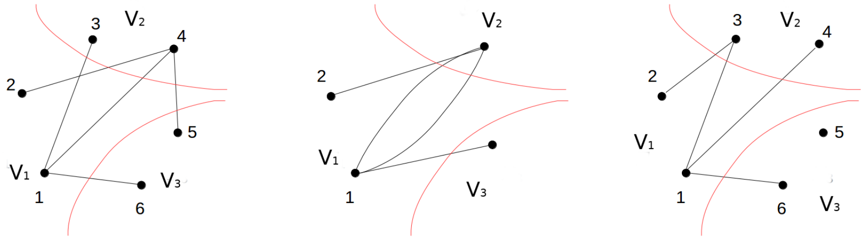


Figure 4: Realizations of G, H^1 and F^1 for the restrictions of the first example

Second example:
The restrictions of the graph are:

- Degree sequence: $S = (3, 1, 1, 3, 1, 1)$.
- Partitions: $\mathcal{P} = \{V_1, V_2, V_3\}$ with $V_1 = \{1, 2\}$, $V_2 = \{3, 4\}$ and $V_3 = \{5, 6\}$.
- Partition matrix :
$$M = \begin{pmatrix} 0 & 4 & 0 \\ 4 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

In order to check the necessary condition we construct for each node group V_i a feasible multigraph H^i. For the example we give the conditions for the feasible multigraph H^1:

- Degree sequence: $S^{V_1} = (3, 1)$.
- The outer edge tuple for V_2 is $(4, 2)$ and for V_3 is $(0, 2)$.
- No edges between outer nodes are allowed.

The conditions for the feasible extended graph F^1 are:

- Degree sequence: $S = (3, 1, 2, 2, 0, 0)$, were the entries $3-6$ belong to outer nodes.
- No edges between the outer nodes are allowed.

The algorithm Extended Havel-Hakimi would construct the restriction of the feasible extended graph F^1 in the the *outer node expansion phase*.
In the *cross-edge addition phase* it adds the edges $\{3, 1\}$, $\{3, 2\}$ and $\{4, 1\}$. After the edges were added, the residual degree of node 4 is not zero. The only node in V_1 not adjacent to node 4 is node 2. Node 2 has residual degree 0. Therefore the Extended Havel-Hakimi returns FALSE. No realization of the restriction of example 2 is possible.
Note that the degree sequence $(3, 1, 2, 2, 0, 0)$ is graphical. The restriction, that the outer nodes cannot be connected, imply that there is no feasible extended graph.

3.2 Approaches for Sufficiency

For a sufficient condition it is necessary to answer the question, whether there exists a graph with the degree sequence S, respecting the partition constraints given by the partition matrix M. This problem is often referred to as the Partition Adjacency Matrix realization problem. In short PAM-realization problem. The PAM realization problem is conjectured to be $np-$complete [18],[22].

It turned out to be helpful to consider the following help graph T. We denote the degree of a node v in the degree sequence S with $d(v)$. The node set of T is

$$V(T) = \{v^u | v, u \in V, u \neq v\} \cup \{a_1^v, ..., a_{n-1-d(v)}^v | v \in V\}$$

and the edge set of T is:

$$E^{in}(T) = \{\{v^u, a_i^v\} | v, u \in V, u \neq v, i = 1, ..., n-1-d(v)\}$$

$$E^{out}(T) = \{\{v^u, u^v\} | v, u \in V, u \neq v\}$$

$$E(T) = E^{out}(T) \cup E^{in}(T)$$

We call T the Tutte gated. The above graph is a special case of a help graph, which Tutte originally introduced in 1954 in order to solve the f-factor problem [23].

Proposition 12. *T has the following property: There is a graph with degree sequence S if and only if T has a perfect matching. From a perfect matching in T we define the edge set*

$$E = \{\{u, v\} | \{v^u, u^v\} \textit{ is in the matching}\}$$

Proof. See [23] [18]. □

We illustrate the statements in figure 5.

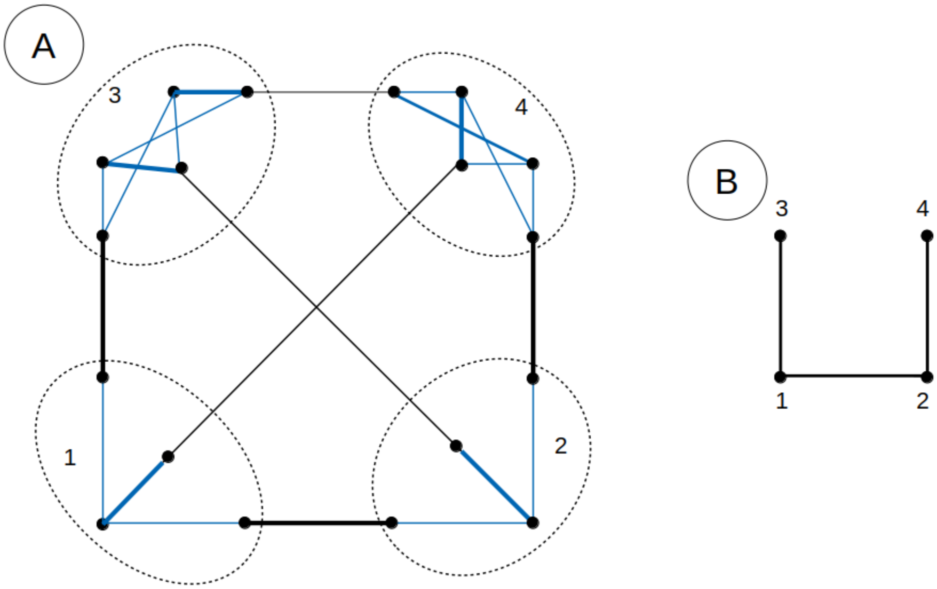


Figure 5: A shows the Tutte gadget of the degree sequence $S = (2, 2, 1, 1)$. The edges in $E^{in}(T)$ are colored blue and the edges in $E^{out}(T)$ are colored black. There is a perfect matching, and the edges in the matching are marked bold. B shows the realization of the degree sequence, which corresponds to the matching.

3.2.1 The Two Group Case

Erdős et al. had the following idea to solve the case of two node groups [18]. They propose to find a realization of the degree sequence with the maximum number of cross edges G^{max} and a realization with the minimal number of cross edges G^{min}. Let $\mathcal{P} = \{V_1, V_2\}$ be the partition of the node set with only two node groups. Finding a graph with the maximum number of crossing edges is equivalent to the problem of finding a perfect matching in T with the maximum number of edges in

$$E^{cross}(T) = \{\{v^u, u^v\} | v \in V_1, u \in V_2\}.$$

We assign the weight 1 to the edges in E^{cross} and weight 0 to the edges in $E(T) \setminus E^{cross}(T)$ and then apply Edmond's algorithm for the maximal weighed matching [24] on T in order to find a maximal perfect matching. A maximum perfect matching corresponds to a graph G^{max} satisfying the degree sequence and having the maximal number of cross edges. When we assign weight 0 to all edges in E^{cross} and weight 1 to the edges in $E(T) \setminus E^{cross}(T)$, we find in a similar way a realization G^{min} of the degree sequence with the minimal possible crossing edges.

The symmetric difference of G^{max} and G^{min} is an Eulerian graph and the edge set decomposes in a collection of closed altering walks. It is possible to iteratively apply a $2-$edge swap on G^{min} in order to transform it into G^{max} [10]. This procedure gives a graph-sequence $(G^{min}, G_1, ..., G_{k-1}, G^{max})$. Let a be the number of crossing edges in G^{min} and b be the number of crossing edges in G^{max}. A $2-$edges swap changes the number of crossing edges by 0 or by 2. Therefore for any c in $\{a, a+2, a+4, ..., b-2, b\}$ there is a graph in $(G^{min}, G_1, ..., G_{k-1}, G^{max})$ with c crossing edges. Let h be an odd number smaller then $b-a$. There cannot be a realization of the degree sequence with $a+h$ crossing edges. Assume there is, then we can remove all the crossing edges in a realization and look at the remaining graph with only the nodes in the first group. The sum of its degrees would be odd, a contradiction. Therefore the set $\{G^{min}, G_1, ..., G_{k-1}, G^{max}\}$ contains for each possible number of crossing arrows one realization.

3.2.2 An Algebraic Monte-Carlo Approach

The Tutte gadget allows to reduce the degree sequence problem to a matching problem. In the PAM-realization problem, we are interested in a particular matching. All possible edges between two node groups correspond to a subset E' of the edges in $E^{out}(T)$. We want to find a matching which has exactly as many edges in E' as the number of desired crossing edges. Because any realization of the Tutte gadget corresponds to a realization of the degree sequence, it is sufficient to require at least as many edges in E' as the number of desired crossing edges. By the pigeonhole principle any realization which has at least as many edges in E' as the number of desired crossing edges, has exactly as many edges in E' as the number of desired crossing edges. This motivates the Dominating Matching Problem.

Dominating Matching Problem: Given a graph G, disjoint subsets $E'_1, ..., E'_k \subset E(G)$, integers $(m_1, ..., m_k)$, is there a perfect matching in G, which uses at least m_j edges from the edge-set E'_j for all $j \in \{1, ..., k\}$?

Czabarka et al. proposed an algebraic Monte-Carlo algorithm in order to solve the dominating matching problem [22] . The idea is based on another result from Tutte described in *The Factorization of Linear Graphs* [25]. We briefly state the main results:

Definition 14. *Let A be a skew-symmetric matrix, with an even number of rows $2h$. Let Q be the set of all partitions of $\{1, 2, ..., 2h\}$ into pairs. An element $q \in Q$ can be written as $q = \{(i_1, j_1), (i_2, j_2), ..., (i_h, j_h)\}$ with $i_1 < j_1, i_2 < j_2, ..., i_h < j_h$ and $i_1 < i_2 < ... < i_h$. For the element $q \in Q$ the corresponding permutation is defined as*

$$\pi_q = \begin{bmatrix} 1 & 2 & 3 & 4 & ... & 2h-1 & 2h \\ i_1 & j_1 & i_2 & j_2 & ... & i_h & j_h \end{bmatrix}$$

The pfaffian *of* A *is*

$$\mathrm{Pf}(A) = \sum_{q \in Q} \mathrm{sign}(\pi_q) A_{i_1,j_1} A_{i_2,j_2} \cdot A_{i_h,j_h}$$

where $\mathrm{sign}(\pi_q)$ *is the signature of* π_q.

For a graph G the matrix A is defined as:

$$A_{i,j} = \begin{cases} x_{i,j} & \text{if } \{i,j\} \in E(G) \text{ and } i < j \\ -x_{i,j} & \text{if } \{i,j\} \in E(G) \text{ and } i > j \\ 0 & \textit{else} \end{cases}$$

where the $x_{i,j}$ are indeterminate variables.

Theorem 5. *A graph G has an perfect matching if and only if* $\mathrm{Pf}(A)$ *is not the 0 polynomial.*

Proof. See [25]. □

The key idea to consider the edge sets $E'_1, ..., E'_k$ in the dominating matching problem is to modify the matrix A. Given a graph G and disjoint edge-subsets $E'_1, ..., E'_k \subset E(G)$ a matrix $\tilde{A}$ is defined as:

$$\tilde{A}_{i,j} = \begin{cases} x_{i,j} z_l & \text{if } \{i,j\} \in E(G) \text{ and } i < j \text{ and } \{i,j\} \in E'_l \\ -x_{i,j} z_l & \text{if } \{i,j\} \in E(G) \text{ and } i > j \text{ and } \{i,j\} \in E'_l \\ x_{i,j} & \text{if } \{i,j\} \in E(G) \text{ and } i < j \text{ and } \{i,j\} \notin \cup_{l=1}^{k} E'_l \\ -x_{i,j} & \text{if } \{i,j\} \in E(G) \text{ and } i > j \text{ and } \{i,j\} \notin \cup_{l=1}^{k} E'_l \\ 0 & \textit{else} \end{cases}$$

where the $x_{i,j}$ and z_l are indeterminate variables. With the matrix $\tilde{A}$ it is possible to state the following proposition.

Proposition 13. *Given a graph G, disjoint subsets $E'_1, ..., E'_k \subset E(G)$, integers $(m_1, ..., m_k)$. There is a perfect matching in G, which uses at least m_j edges from the edge-set E'_j for all $j \in \{1, ..., k\}$ if and only if the polynomial* $\mathrm{Pf}(\tilde{A})$ *has a term in which the exponent of z_l is at least m_l for every $l \in \{1, ..., k\}$.*

Proof. See [22]. □

Let W be space space of multivariate polynomials in the variables $z_1, z_2, ...$. We define the difference operator $\nabla_{z_1} : W \to W$ as

$$\nabla_{z_1} f = f(z_1, z_2, ...) - f(z_1 - 1, z_2, ...).$$

Note that the exponent of z_1 is strictly smaller in $\nabla_{z_1} f$ than in f. The composition of h times the ∇_{z_1} operator is denoted with $\nabla_{z_1}^{h}$ and the of different operators we use the product notations $\prod_{i=1}^{2} \nabla_{z_i} = \nabla_{z_1} \nabla_{z_2}$. The composition is commutative [22]. Recall that $Pf(\tilde{A})$ is a polynomial in the variables z_l for $l \in \{1, ..., k\}$ and $x_{i,j}$ for $i, j \in \{1, ..., n\}$. We now can state

Theorem 6. *Given a graph G, disjoint subsets $E'_1, ..., E'_k \subset E(G)$, integers $(m_1, ..., m_k)$. There is a perfect matching in G, which uses at least m_j edges from the edge-set E'_j for all $j \in \{1, ..., k\}$ if and only if the polynomial*

$$\left(\prod_{l=1}^{k} \nabla_{z_l}^{m_l} \right) \mathrm{Pf}(\tilde{A}) \tag{6}$$

is not the 0 polynomial.

Proof. See [22]. □

Theorem 6 gives a sufficient condition for the existence of a dominating matching. The polynomial (6) cannot be computed efficiently because the pfaffian polynomial cannot be computed efficiently. However, if the variables are substituted, the pfaffians can be evaluated efficiently [26]. Polynomial (6) can be written out as the sum of $\prod_{l=1}^{k}(m_l + 1)$ summands where each summand consists of a pfaffian and a coefficient.

The procedure to determine whether the polynomial is the zero polynomial is to randomly select values form a Galois field and substitute the variables. Then evaluate the polynomial (6). If the evaluated polynomial is not 0 then the polynomial in not the zero polynomial and we know with certainty that a dominating matching exists and the procedure returns TRUE. If the evaluated polynomial is 0 we repeat the random substitution. After a given number of repetitions the procedure returns FALSE.

If the result is 0 then either we evaluated the zero polynomial or the polynomial is not the zero polynomial and we the randomly have chosen a root. The probability of randomly choosing a root of a non 0 polynomial depends on the Galois field. The error probability can be controlled by the number of random trials and by choosing the Galois field sufficiently large using the Schwartz-Zippel Lemma [27].

Theorem 7. *There is a Monte-Carlo procedure for the Dominating Matching Problem, which runs in polynomial time under the assumption that $\prod_{l=1}^{k}(m_l + 1) = \mathcal{O}(polynomial(n))$, which certainly holds for a constant k. If the procedure returns TRUE, then the sought after graph exists, if the procedure returns FALSE, then with high probability such a graph does not exist.*

Proof. See [22]. □

4 A Statistical Motivation of the State Space

The goal of this section is to inquire what ensemble is a sensible comparison for the observed graph. For this purpose it is necessary to specify the probability space from which the observed graph is suspected to be. Based on the probability space the comparison is formulated as a test.

4.1 Probability Distributions on Graphs

Let $\mathcal{G}$ be the set of all graphs with a given number of nodes and $\mathcal{A}$ its power set $\mathbf{P}(\mathcal{G})$. $\mathcal{A}$ is a sigma algebra with the basic set $\mathcal{G}$. We start with a few statements for discrete probability spaces.

Definition 15. *For any function $p : \mathcal{G} \to \mathbb{R}$ we call $\mathbb{P}_p : \mathcal{A} \to \mathbb{R}$ with*

$$\mathbb{P}_p(A) := \sum_{G \in A} p(G)$$

the from p induced set function*.*

Proposition 14. *For any probability measure $\mathbb{P}$ on $(\mathcal{G}, \mathcal{A})$ there is a function $p : \mathcal{G} \to \mathbb{R}$ such that $\mathbb{P} = \mathbb{P}_p$.*

Proof. Let p be defined as $p : \mathcal{G} \to \mathbb{R}$ with $p(G) := \mathbb{P}(\{G\})$. Then for any $A \in \mathcal{A}$

$$\begin{aligned} \mathbb{P}(A) &= \mathbb{P}(\cup_{G \in A}\{G\}) \\ &= \sum_{G \in A} \mathbb{P}(\{G\}) \\ &= \sum_{G \in A} p(G) \\ &= \mathbb{P}_p(A) \end{aligned}$$

□

Proposition 15. *For any function $p : \mathcal{G} \to [0,1]$ with $\sum_{G \in \mathcal{G}} p(G) = 1$, the from p induced set function $\mathbb{P}_p$ is a probability measure.*

Proof. The sigma additivity follows directly from the definition 15. The positivity follows from the positivity of p and $\mathbb{P}_p(\mathcal{G}) = \sum_{G \in \mathcal{G}} p(G) = 1$. □

We consider probability distributions of the exponential family. Models of the exponential family are among the most studied and used.

Definition 16. *We call a probability measure $\mathbb{P}$ of the* exponential family *if there is a function $T : \mathcal{G} \to \mathbb{R}^k$, with the components $T_1, ..., T_k$, coefficients $\beta \in \mathbb{R}^k$ and a constant ϕ such that $\mathbb{P}$ is induced by $p : \mathcal{G} \to [0,1]$ with*

$$p(G) := exp(\sum_{i=1}^{k} \beta_i T_i(G) - \phi). \tag{7}$$

$T_1, ..., T_k$ are usually functions such as number of edges or number of incident edges to a particular node[3]. The triple $\mathcal{S}_\mathcal{G} = (\mathcal{G}, \mathcal{A}, \mathbb{P})$, where $\mathbb{P}$ is a probability

[3] We use the notation of [28], for a general introduction to statistical network models, see [29].

measure of the *exponential family*, is a probability space.

The exponential family includes many of the well known random graphs. Consider T_1 equal to the number of edges, define $q := \frac{exp(\beta)}{exp(\beta)+1}$ and set $\phi(\beta) = -log(1-q)\binom{n}{2}$.

$$\begin{aligned} P(G) &= exp(\beta_1 T_1(G) - \phi(\beta)) \qquad (8) \\ &= exp(log(\frac{q}{1-q})T_1(G) + log(1-q)\binom{n}{2}) \qquad (9) \\ &= \frac{q}{1-q}^{T_1(G)}(1-q)^{\binom{n}{2}} \qquad (10) \\ &= q^{T_1(G)}(1-q)^{\binom{n}{2}-T_1(G)} \qquad (11) \end{aligned}$$

which is the Erdős Rényi model, where each edge has independently a probability of q. Another common choice for $T_1, ..., T_k$ is that for each node i, T_i is the degree of node i. This model allows to capture degree heterogeneity in a network. The corresponding β has the interpretation of affinity of this node. A natural extension would be to further consider groups of nodes, which influence the edge formation between them. In the context of social networks an example would be the node groups induced by gender. This is captured by additional statistic T_j, $j > n$ which counts the edges between two groups, or within one group. The corresponding β_j has the interpretation of affinity between these groups.

The set of all graphs with a given number of nodes $\mathcal{G}$ is finite. Therefore the function T can only take on finitely many values $(a_1, ..., a_m)$.

Definition 17. *Let $\mathcal{G}_{a_h}$ be equal to $\{G \in \mathcal{G} | T(G) = a_h\}$, $\mathcal{A}_{a_h}$ be the power set of $\mathcal{G}_{a_h}$ and $\mathbb{P}_{a_h} : \mathcal{A}_{a_h} \to [0,1]$ be a measure with*

$$\mathbb{P}_{a_h}(A) := \frac{\mathbb{P}(A)}{\mathbb{P}(\mathcal{G}_{a_h})}.$$

We call the triple $\mathcal{S}_{a_h} = (\mathcal{G}_{a_h}, \mathcal{A}_{a_h}, \mathbb{P}_{a_h})$ the on a_h conditional probability space*.*

Note that $\mathcal{A} \supset \mathcal{A}_{a_h}$ and $\mathbb{P}$ is of the exponential family and thus positive for any except the empty set. Therefore the on a_h conditional probability space is well defined.

Proposition 16. *Let $(\mathcal{G}_{a_h}, \mathcal{A}_{a_h}, \mathbb{P}_{a_h})$ be the* on a_h conditional probability space*. $\mathbb{P}_{a_h}$ is the uniform distribution.*

Proof. We have to show that for any $G_1, G_2 \in \mathcal{G}_{a_h}$ is $\mathbb{P}_{a_h}(\{G_1\}) = \mathbb{P}_{a_h}(\{G_2\})$.

Let G_1, G_2 be two graphs from $\mathcal{G}_{a_h}$, then

$$\begin{aligned}
\mathbb{P}_{a_h}(\{G_1\}) &= \frac{p(G_1)}{\mathbb{P}(\mathcal{G}_{a_h})} \\
&= \frac{exp(\sum_{i=1}^{k} \beta_i T_i(G_1) - \phi)}{\mathbb{P}(\mathcal{G}_{a_h})} \\
&= \frac{exp(\sum_{i=1}^{k} \beta_i (a_h)_i - \phi)}{\mathbb{P}(\mathcal{G}_{a_h})} \\
&= \frac{exp(\sum_{i=1}^{k} \beta_i T_i(G_2) - \phi)}{\mathbb{P}(\mathcal{G}_{a_h})} \\
&= \frac{p(G_2)}{\mathbb{P}(\mathcal{G}_{a_h})} \\
&= \mathbb{P}_{a_h}(\{G_2\})
\end{aligned}$$

□

4.2 Test

We are interested in comparing the observed graph to an ensemble. Usually the graphs are not compared directly but a set of characteristics is used for comparison. Based on the characteristics it is decided, whether the observed graph differs substantially form the ensemble. This can be formalized using test theory. The 0-hypothesis is: the observed graph does not differ substantially from the ensemble.

Definition 18. *Let $\mathcal{S}_\Omega = (\Omega, \mathcal{A}_\Omega, \mathbb{P}_\Omega)$ be a probability space, with $\mathcal{A}_\Omega$ being the power set of Ω. Let $X : \Omega \to \mathbb{R}^l$ be a function of characteristics, $\mathcal{O} = (O^1, O^2, O^3)$ where O^1, O^2, O^3 are disjoint subsets of $\mathbb{R}^l$ with $\mathbb{P}_\Omega(X^{-1}(O^1 \cup O^2 \cup O^3)) = 1$ and $r, \alpha \in [0, 1]$.*
We call the quintuple $(\mathcal{S}_\Omega, X, \mathcal{O}, r, \alpha)$ test of X to significance level α *if*

$$\alpha = \mathbb{P}_\Omega(X^{-1}(O^3)) + r\mathbb{P}_\Omega(X^{-1}(O^2))$$

The interpretation is that O^3 is the area in which the 0-hypothesis is rejected. O^2 is the area in which the 0-hypothesis rejected with probability r. O^1 is the area in which the 0-hypothesis is not rejected.
We call a test $(\mathcal{S}_\Omega, X, \mathcal{O}, r, \alpha)$, where $\mathcal{S}_\Omega$ is a conditional probability space, a conditional test.

Theorem 8. *Let $(\mathcal{S}_{a_h}, X_{a_h}, \mathcal{O}_{a_h}, r_{a_h}, \alpha)$, $h \in \{1, ..., m\}$ be conditional tests to significance level α for all possible values a_h, which T can take on. Then there is a test $(\mathcal{S}_\mathcal{G}, X, \mathcal{O}, r, \alpha)$ to the significance level α, such that* *for any graph G for which the 0-hypothesis is with certainty rejected by $(\mathcal{S}_{T(G)}, X_{T(G)}, \mathcal{O}_{T(G)}, r_{T(G)}, \alpha)$, the 0-hypothesis is with certainty rejected by $(\mathcal{S}_{\mathcal{G}}, X, \mathcal{O}, r, \alpha)$.*

Proof. For every $G \in \mathcal{G}$ there is a $h \in \{1, ..., m\}$ such that $T(G) = a_h$. Recall $\mathcal{O}_{a_h} = (O^1_{a_h}, O^2_{a_h}, O^3_{a_h})$. Since $\mathbb{P}(\{G\}) > 0$, G is either in $X^{-1}_{a_h}(O^1_{a_h})$ or in $X^{-1}_{a_h}(O^2_{a_h})$ or in $X^{-1}_{a_h}(O^3_{a_h})$. Therefore $\mathcal{G}^i = \cup^m_{h=1} X^{-1}_{a_h}(O^i_{a_h})$ with $i = 1, 2, 3$ is a disjoint partition of $\mathcal{G}$.

$$\begin{aligned}
\mathbb{P}(\mathcal{G}^1) &= \sum_{h \in \{1,...,m\}} \mathbb{P}(X^{-1}_{a_h}(O^1_{a_h})) \\
&= \sum_{h \in \{1,...,m\}} \mathbb{P}(\mathcal{G}_{a_h}) \mathbb{P}_{a_h}(X^{-1}_{a_h}(O^1_{a_h})) \\
&\leq \sum_{h \in \{1,...,m\}} \mathbb{P}(\mathcal{G}_{a_h})(1 - \alpha) \\
&= (1 - \alpha)
\end{aligned}$$

Analogously follows $\mathbb{P}(\mathcal{G}^3) \leq \alpha$. If $\mathcal{G}^2 \neq \emptyset$ then $\mathbb{P}(\mathcal{G}^2) > 0$ and the inequalities are strict. If $\mathcal{G}^2 \neq \emptyset$ set $r := \frac{\alpha - \mathbb{P}(\mathcal{G}^3)}{\mathbb{P}(\mathcal{G}^2)} = 1 - \frac{(1-\alpha) - \mathbb{P}(\mathcal{G}^1)}{\mathbb{P}(\mathcal{G}^2)}$ otherwise set $r := 0.5$. r is strictly between 0 and 1. We now define $X : \mathcal{G} \to \mathbb{R}$ with

$$X(G) := \begin{cases} 0 & \text{if } G \in \mathcal{G}^1 \\ r & \text{if } G \in \mathcal{G}^2 \\ 1 & \text{if } G \in \mathcal{G}^3. \end{cases}$$

Set $\mathcal{O} = (\{0\}, \{r\}, \{1\})$. $(\mathcal{S}_{\mathcal{G}}, X, \mathcal{O}, r, \alpha)$ is a test with the desired properties. □

The challenge of setting up a test for a characteristic of interest X is to determine the rejection sets $\mathcal{O}$. To do so, the probability distribution of X has to be known. This is especially problematic because the parameters β as well as the normalizing constant are usually unknown. In order to circumvent this problem we propose to evaluate T at the observed graph G_{obs} and conduct a conditional test $(\mathcal{S}_{T(G_{obs})}, X_{T(G_{obs})}, \mathcal{O}_{T(G_{obs})}, r_{T(G_{obs})}, \alpha)$, where $X_{T(G_{obs})}$ is the restriction of X on $\mathcal{G}_{T(G_{obs})}$. According to proposition 16 the graphs in $\mathcal{G}_{T(G_{obs})}$ are uniformly distributed independent of β. Theorem 8 states that the conditional test to significance level α corresponds to a test to significance level α on the whole graph space $\mathcal{G}$.

The image space of $X_{T(G_{obs})}$ is finite $\{x_1, ..., x_o\}$. The probability distribution of $X_{T(G_{obs})}$ is therefore fully described by the probability of the elements in the image space $(p_{x_1}, ..., p_{x_o})$. The set $\mathcal{G}_{T(G_{obs})}$ is usually too big as to be computed explicitly. The distribution of $X_{T(G_{obs})}$ can be estimated by uniformly drawing G out of $\mathcal{G}_{T(G_{obs})}$ and calculate $X_{T(G_{obs})}(G)$. For n draws $G_1, ..., G_n$ let n_{x_i} be $\sum^n_{k=1} \mathbb{1}(X_{T(G_{obs})}(G_k) = x_i)$, where $\mathbb{1}$ is an indicator function. Set as the estimated distribution $\tilde{p}_{x_i} := \frac{n_{x_i}}{n}$ for $i \in \{1, ..., o\}$.

Proposition 17. *The distrubtion estimate $\tilde{p}_{x_i}$ for $i \in \{1, ..., o\}$ is consistent.*

Proof. Let $Y_k^i : \mathcal{G}_{T(G_{obs})} \to \mathbb{R}$ with

$$Y_k^i(G) := \begin{cases} 1 & \text{if } X_{T(G_{obs})}(G) = x_i \\ 0 & \text{else} \end{cases}$$

for $k \in \mathbb{N}$ be a series of i.i.d. random variables. $\tilde{p}_{x_i}$ can be written as

$$\tilde{p}_{x_i} = \frac{1}{n} \sum_{k=1}^{n} Y_k^i.$$

The expectation of $\tilde{p}_{x_i}$ is

$$E(\tilde{p}_{x_i}) = E\left(\frac{1}{n}\sum_{k=1}^{n} Y_k^i\right) = \frac{1}{n}\sum_{k=1}^{n} E(Y_k^i) = p_{x_i}.$$

The Y_k^i are uncorrelated because the draws are independent, therefore the variance is

$$Var(\tilde{p}_{x_i}) = Var\left(\frac{1}{n}\sum_{k=1}^{n} Y_k^i\right) = \frac{1}{n^2}\sum_{k=1}^{n} Var(Y_k^i) = \frac{Var(Y_k^i)}{n}$$

which converges to 0 for $n \to \infty$. The point estimates $\tilde{p}_{x_i}$ are therefore consistent. The distribution estimate is composed of finite many point estimates and therefore is the estimated distribution is consistent. □

4.3 Procedure

Let G_{obs} be the observed graph. First consider the characteristics, which are suspected to influence the graph-probability. Common characteristics are:

1. Degree-sequence in order to capture the node heterogeneity.

2. Number of crossing edges between node groups in order to capture affinity between groups.

3. Number of edges within a particular group in order to capture affinity within the group (or maybe impossibility of edge-formation within the group).

4. Triangle count, in order to capture the transitivity index of the graph.

These characteristics correspond to the statistic T. Choose a graph characteristic, which influence has to be tested. This characteristic corresponds to the statistic X. Estimate the distribution of X by uniform sampling from $\mathcal{G}_{T(G_{obs})}$. Define the rejection areas $\mathcal{O}$, using the estimated distribution. Evaluate X at G_{obs} and decide on whether to reject the 0-hypothesis based on the area, which

$X(G_{obs})$ falls in.

The algorithm proposed in Section 2 allows to draw uniformly out of the set $\mathcal{G}_{T(G_{obs})}$, where T corresponds to characteristics 1-3. The algorithms discussed in the introduction account exclusively for characteristic 1. As discussed in Section 2.4, there are swap algorithms which try to account for characteristic 4 [19]. However, depending on the configurations, this could potentially cause the state graph to be disconnected, even for simple graphs [30]. Note that the proposed algorithm can be extended in order to account for more complex constraints, such as number of triangles. This is done by modifying the constraint check. However, apart from the constraint concerning the partition adjacency matrix restriction, the constraints may be computational expensive to check.

5 Application

The theory developed in the previous section helps to understand the Nyakatoke risk sharing network. Nyakatoke is a small Haya village in the Kagera Region of Tanzania. In 2001, Joachim De Weerdt conducted a census of all 119 households and documented the links between them [31]. The nodes of the network are the households. The edges are risk sharing relationships. In order to determine the links, the following questions have been asked to the households:

> *"Can you give a list of people from inside or outside of Nyakatoke, who you can personally rely on for help and/or that can rely on you for help in cash, kind or labour."[2]*

If two households refer reciprocally to each other, a link is formed. There are 738 links in the network. Apart from the relationships between them, also a variety of household characteristics were collected. The most important ones are:

- head age
- head sex
- education
- wealth
- religion
- tribe
- occupation (with multiple occupation allowed)

Figure 6 depicts the Nyakatoke network. Despite its modest size of only 119 nodes, it is impossible to gain some information on the link formation properties by visual analysis.

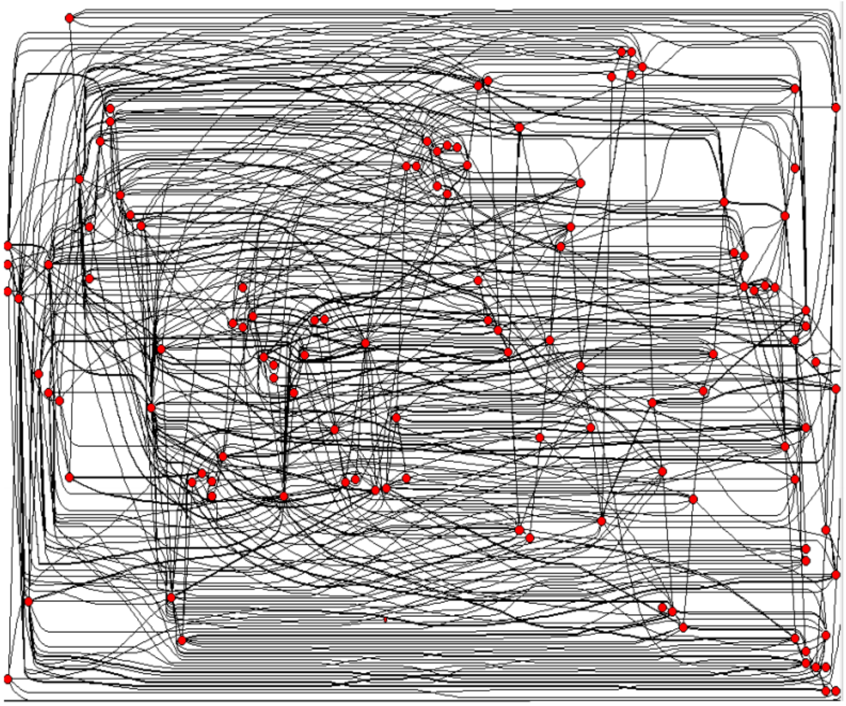


Figure 6: The Nyakatoke network: A plot illustrating the Nyakatoke network. The red dots are the households, the edges are the risk sharing links. [4]

We were interested in the link formation between households of different wealth. A link is formed if two households claim to seek help from each other in case of financial distress. A link can impact a household in two ways. Either it has to help or it can seek help from another household. While one way is positive for the household, the other way is negative. The overall benefit, however, can be positive because households valuate income more in distressed situation than in a situation of abundance [5]. If, however, there are asymmetries in wealth between the households, then the richer household has less chance to obtain meaningful help in case of distress. Therefore, a positive net benefit of the richer household is less likely and the probability of link-formation is decreased. This is the reasoning we try to falsify by setting up the

> *0-hypothesis:* There are equally many (or more) risk-sharing links between poor and rich households than in a random network.

Apart from wealth we also suspect that education has an influence on the link

[4] Source: plot taken from the survey-paper [2]

[5] This statement is based on decreasing marginal utility or equivalently concavity of the utility function, which is a standard assumption.

of network formation. There may be a connection based on friendship obtained during school time. Additionally, for a connection involving two households with the same educational background, there is less asymmetry in knowledge.
Education often goes with wealth. Less links between the rich and poor households may be due to different education levels predominant in groups of different wealth. Comparing the observed network to networks with the same degree sequence, could, therefore, lead to a distorted estimate of the distribution. It is important to consider education when determining which ensemble of graphs the observation is compared to. For the test distribution, the comparison ensemble was chosen as all the graphs with the same degree-sequence and the same number of links between the different educational groups.
Since the alternative hypothesis is a reduction in the number of links, the test is one-sided. The level of significance is set at 5%.

The estimated distribution under the null-model is shown in figure 7. The observed graph is marked by the red line. It lies in the 0.021 quantile of the test-distribution. We therefore reject the $0-$hypothesis, which can be interpreted as statistical evidence for our reasoning about link formation between households of different wealth.

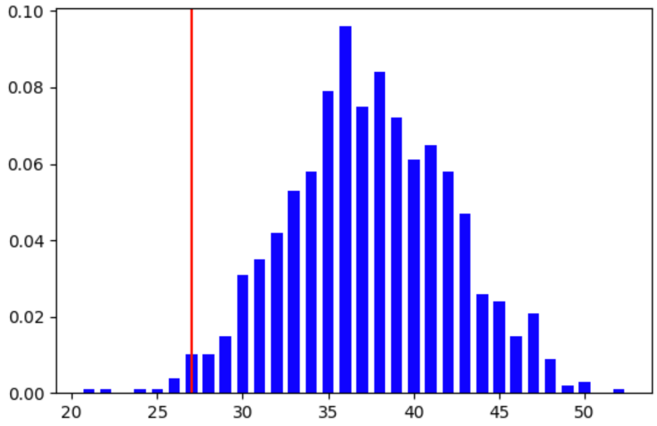
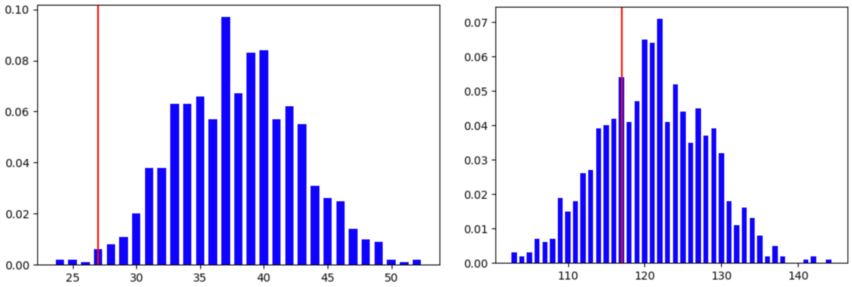


Figure 7: The distribution of risk sharing links between poor and rich households in the Nyakatoke network under the null-model. The red line marks the number of links in the observed network.

The estimated test distribution was estimated considering the effect of education. In order to see how education is influencing the results, the distribution is estimated without fixing the number of links across educational groups. The resulting distribution is shown in Figure 8 (a). The observed graph lies in the 0.011 quantile. Also in this case, the 0-hypothesis is rejected. The mean of the distribution is shifted down by 0.5 links if education is not controlled for. The

shift in the mean is significant[6], and goes in the expected direction. However, the magnitude of the change in distribution when controlling for the effect of education is very low.
In order to understand the minor impact of education on the test distribution, the influence of education on link formation is inquired. Figure 8 (b) shows the crossing of high (secondary) to low (primary) educated households (head of household is relevant) of the observed graph in comparison to graphs with the same degree sequence. The assumption that the level of education has an impact on link formation is not supported. More precisely, the hypothesis stating that education has no impact, cannot be rejected (at any reasonable level of significance). This is consistent with the small change in distribution, as when we considered education in the distribution estimate for rich-poor links.[7]



(a) Wealth

(b) Education

Figure 8: Inquiring the influence of education on the link distribution. (a) shows the distribution of risk sharing links between poor and rich households in the Nyakatoke network when education is not controlled for. (b) shows the number of risk-sharing links between high (secondary) and low (primary) educated households. The red lines mark the number of links in the observed network.

6 Conclusion

This thesis inquiries graphs with a fixed degree sequence under partition constraints.

The main result is an algorithm which generates for a given graph a uniform sample out of the same PAM-restriction class. This algorithm can be used for a Markov-Chain Monte Carlo estimation of a distribution of a graph characteristic. We prove the correctness of the algorithm using only elementary results. Improvements on the original algorithm are discussed. In particular, how hard constraints and isolated nodes can be considered during the

[6] to a significance level of 5 %

[7] The paragraph *Further Inquire of the Network* also uses the sample algorithm. However, its purpose is rather of the type *descriptive statistics* than *testing* because the procedure: *setting up model, deriving the 0-hypothesis, defining level of significance and then testing* is not followed. Nevertheless, it is helpful to reveal interesting network properties.

construction of the switch set. This leads to a lower rejection rate of switch sets.

We provide examples, showing that for existing edge swap algorithms, the state graph becomes unconnected when considering the PAM-restrictions. Alternative proposed procedures to randomly rewire the edges and reject the step if the new network violates any restriction. This class of procedures is not practical because the rejection probability approaches 1 rapidly with the increasing size of the network.

A central question in all Markov-Chain algorithms used to generate graphs is whether the Markov-Chain is strongly mixing. This question is still open for the case, with no partition constraints. It is also open for the PAM-restricted case.

A natural question which arises is whether there exists a feasible realization to a given degree sequence and partition restrictions. In the literature, this problem is referred to as Partition Adjacency Matrix realization problem and is conjectured to be *np*-complete. We discussed the solution approach. In the two group case, there is a polynomial decision algorithm. In the general case, there is an algebraic random decision algorithm. The error probability can be set minimal but positive.

A connection between the PAM-restricted network and the theory of random graphs is established. In particular, it is elaborated how the partition constraints relate to random graphs of the exponential family as sufficient statistic. The knowledge about the probability distribution of the graphs is used to construct a test framework for the random network. Using this test framework the choice of the comparison is statistically justified.

The developed theory is applied to analyze the risk sharing network of Nyakatoke. The procedure provides evidence that wealth does play a role when deciding which risk sharing links to form.

References

[1] Nicholas J. Gotelli. Null model analysis of species co-occurrence patterns. *Ecology*, 81:2606–2621, 09 2000.

[2] Joachim De Weerdt. Risk-sharing and endogenous network formation. WIDER Working Paper Series 057, World Institute for Development Economic Research (UNU-WIDER), 2002.

[3] Joseph Blitzstein and Persi Diaconis. A sequential importance sampling algorithm for generating random graphs with prescribed degrees. *Internet Math.*, 6(4):489–522, 2010.

[4] Charo I. Del Genio, Hyunju Kim, Zoltán Toroczkai, and Kevin E. Bassler. Efficient and exact sampling of simple graphs with given arbitrary degree sequence. *Plos One*, 5(4):1–7, 2010.

[5] H Kim, C I Del Genio, K E Bassler, and Z Toroczkai. Constructing and sampling directed graphs with given degree sequences. *New Journal of Physics*, 14(2):023012, 2012.

[6] László Lovász. Random walks on graphs: A survey, combinatorics, paul erdos is eighty. *Bolyai Society, Mathematical Studies*, 2, 01 1993.

[7] Alistair Sinclair. *Algorithms for Random Generation and Counting: A Markov Chain Approach.* Birkhauser Verlag, Basel, Switzerland, 1993.

[8] Ravi Kannan, Prasad Tetali, and Santosh Vempala. Simple markov-chain algorithms for generating bipartite graphs and tournaments. *Random Structures & Algorithms*, 14(4):293–308, 1999.

[9] I. Miklós and J. Podani. Randomization of presence–absence matrices: Comments and new algorithms. *Ecology*, 85(1):86–92, 2004.

[10] Annabell Berger and Matthias Müller-Hannemann. Uniform sampling of undirected and directed graphs with a fixed degree sequence. *CoRR*, abs/0912.0685, 2009.

[11] Giovanni Strona, Domenico Nappo, Francesco Boccacci, Simone Fattorini, and J San-Miguel-Ayanz. A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. *Nature communications*, 5:4114, 06 2014.

[12] C. J. Carstens. Proof of uniform sampling of binary matrices with fixed row sums and column sums for the fast curveball algorithm. *Physical Review E*, 91:042812, Apr 2015.

[13] Corrie Jacobien Carstens, Annabell Berger, and Giovanni Strona. Curveball: a new generation of sampling algorithms for graphs with fixed degree sequence. *CoRR*, 2016. arxiv:1609.05137.

[14] B. Fosdick, D. Larremore, J. Nishimura, and J. Ugander. Configuring random graph models with fixed degree sequences. *SIAM Review*, 60(2):315–355, 2018.

[15] Peter Mahlmann and Christian Schindelhauer. Peer-to-peer networks based on random transformations of connected regular undirected graphs. In *Proceedings of the Seventeenth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 155–164, New York, NY, USA, 2005. ACM.

[16] T. Feder, A. Guetz, M. Mihail, and A. Saberi. A local switch markov chain on given degree graphs with application in connectivity of peer-to-peer networks. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 69–76, Oct 2006.

[17] Sven Oliver Krumke and Hartmut Noltemeier. *Graphentheoretische Konzepte und Algorithmen*. Springer DE, 2009.

[18] Peter Erdős, Stephen G. Hartke, Leo van Iersel, and István Miklós. Graph realizations constrained by skeleton graphs. *Electronic Journal of Combinatorics*, 24, 08 2015.

[19] Lionel Tabourier, Camille Roth, and Jean-Philippe Cointet. Generating constrained random graphs using multiple edge switches. *J. Exp. Algorithmics*, 16:1.7:1.1–1.7:1.15, December 2011.

[20] Colin Cooper, Martin Dyer, and Catherine Greenhill. Sampling regular graphs and a peer-to-peer network. *Comb. Probab. Comput.*, 16(4):557–593, July 2007.

[21] Pèter L. Erdős, István Miklós, and ZOLTÁN TOROCZKAI. New classes of degree sequences with fast mixing swap markov chain sampling. *Combinatorics, Probability and Computing*, 27(2):186–207, 2018.

[22] Eva Czabarka, Laszlo A Szekely, Zoltan Toroczkai, and Shanise Walker. An algebraic monte-carlo algorithm for the bipartite partition adjacency matrix realization problem. *arXiv preprint arXiv:1708.08242*, 2017.

[23] W. T. Tutte. A short proof of the factor theorem for finite graphs. *Canadian Journal of Mathematics*, 6:347–352, 1954.

[24] Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards*, 69:125–130, 1965.

[25] W. T. Tutte. The Factorization of Linear Graphs. *Journal of the London Mathematical Society*, s1-22(2):107–111, 1947.

[26] G. Galbiati and F. Maffioli. On the computation of pfaffians. *Discrete Applied Mathematics*, 51(3):269 – 275, 1994.

[27] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.

[28] Sourav Chatterjee and Persi Diaconis. Estimating and understanding exponential random graph models. *The Annals of Statistics*, 41(5):2428–2461, 10 2013.

[29] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, February 2010.

[30] Joel Nishimura. Swap connectivity for two graph spaces between simple and pseudo graphs and disconnectivity for triangle constraints. *arXiv preprint arXiv:1704.01951*, 04 2017.

[31] Joachim De Weerdt. Community organisations in rural tanzania: A case study of the community of nyakatoke, bukoba rural district. *Economic Development Initiatives*, 2001.