Uniform Generation of Graphical Realizations of Joint Degree Matrices

by

Qianye Zhou

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2023

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

In this thesis, my primary contributions involve the design of the main algorithms and the derivation of the main theorem. I'd like to emphasize the significant support I received from my supervisor, Jane Gao, whose insights greatly elevated the quality of my work. She not only provided valuable suggestions for improving the algorithms but also directly refined my paper's writing, including restructuring the overall composition. These revisions played a crucial role in enhancing the quality of this thesis.

Abstract

In this thesis, we introduce JDM_GEN, an algorithm designed to uniformly generate graphical realizations of a given joint degree matrix. Amanatidis and Kleer [2] previously employed an MCMC-based method to address this problem. Their method fully resolved the case of two degree classes, and showed that their switch Markov chain is rapidly mixing. While our algorithm imposes certain restrictions on the maximum degrees, it is applicable to any bounded number of degree classes and has a runtime complexity linear in the number of edges.

Acknowledgements

I would like to thank my supervisor, Jane Gao. Her invaluable guidance, consistent patience, and dedicated assistance were crucial in the completion of this thesis.

I would also like to thank Penny Haxell and Joseph Cheriyan for taking the time to read the thesis and providing me with their insightful comments.

Dedication

This thesis is dedicated to my parents, Hongxing Zhou and Min Zhao, for their love and support throughout my journey. Also, to my girlfriend, Qiwen Wang, for standing by me during times of struggle. Your presence has meant a lot.

Table of Contents

Chapter 1

Introduction

Research into the generation of random graphs has a rich and varied history with numerous distinct branches. The most fundamental problem is the uniform generation of a simple graph with a given degree sequence. Tinhofer presented an approach to address this problem in [17]. However, the exact runtime of Tinhofer's algorithm is still undetermined.

Based on the enumeration works of Bender and Canfield [5], Békéssy et al.[4], and Bollobás[6], some simple algorithms can be designed to uniformly generate random graphs. These algorithms have a runtime complexity that is linear in the number of vertices, denoted by n, but exponential in the average degree. Therefore, these algorithms are only efficient when the average degree is small.

Using the configuration model presented in [6], one can easily generate multi-graphs with a given degree sequence. With this, McKay and Wormald [16] introduced switchings, a novel technique for removing multi-edges and loops from generated multi-graphs. They also developed a carefully crafted rejection scheme to maintain uniformity. As a result, the algorithm they designed could efficiently and uniformly generate random graphs even with a larger average degree. Specifically, when tasked with generating d-regular graphs, the algorithm can achieve a running time of $O(nd^3)$, provided that $d = O(n^{1/3})$. While this was a major breakthrough, there was still room for further improvements. Expanding upon this, Gao and Wormald [10] introduced different classes of switchings and improved the degree requirements to $d = o(n^{1/2})$. Subsequently, Arman, Gao, and Wormald [3] introduced the method of incremental relaxation, which enhanced the rejection scheme used during the algorithm's execution. This further improved the running time to $O(dn + d^4)$, provided $d = o(n^{1/2})$.

Another significant approach to generating random graphs is the utilization of Markov

Chain Monte Carlo (MCMC) methodologies. The core idea behind MCMC-based methods involves constructing a Markov chain where the states of this chain represent graphs that satisfy the given degree sequence, along with some auxiliary structures. Transitions between these states are achieved through random operations, such as the addition or deletion of edges. Over time, these transitions can lead the system towards a distribution that is approximately uniform across all states. The time taken to approach this result is known as the 'mixing time'. Remarkably, for specific degree sequences, the system may converge to the uniform distribution very quickly. When it only takes polynomially many steps to achieve this, the phenomenon is referred to as 'rapid mixing'. Jerrum and Sinclair [13] introduced a fully polynomial almost uniform generator that achieves 'rapid mixing' for a particular class of degree sequences, known as the P-stable class. Notably, this class includes all degree sequences for regular graphs.

Another distinct Markov chain, known as the switch chain, was first introduced by Kannan, Vempala, and Tetali [14]. In the switch chain, transitions occur by switching two edges in a specific manner. This method achieved rapid mixing for the generation of random bipartite regular graphs. Since its introduction, the switch chain has been used to handle various problems in the field of random graph generation. In [7], Cooper, Dyer, and Greenhill utilized the switch chain to generate d-regular graphs, achieving a mixing time of approximately $d^{24}n^9 \log n$. Subsequently, Greenhill adapted this approach to non-regular graphs [11], with the limit of mixing time calculated as $\Delta^{14}M^{10}\log M$, where Δ and M represent the maximum degree and the degree sum, respectively. As per the recent results in [12], the current restrictions on Δ and M have been set as $3 \leq \Delta \leq \frac{1}{3}\sqrt{M}$.

The two methods for generating random graphs discussed above each have their own strengths. The switching-based algorithms typically offer a more efficient runtime and can generate results that follow an exact uniform distribution. On the other hand, MCMC-based algorithms generate results with an approximately uniform distribution, but they generally have more lenient requirements on the degree sequence.

In this thesis, our research is centered around the generation of graphical realizations of a given joint degree matrix (JDM). The motivation for studying this particular type of random graph generation was originally proposed by Mahadevan et al. in [15]. They argued that, unlike the traditional problem which aims to generate graphs based on a given degree sequence, generating graphs that satisfy a certain joint degree matrix allows for the creation of synthetic graphs that more closely resemble real-world network topologies.

The problem we address involves a family of q pairwise disjoint vertex sets denoted by $\boldsymbol{V} = (V_1, ..., V_q)$, where $q \geq 1$ is a positive integer, and a sequence of q nonnegative integers, denoted by $\boldsymbol{d} = (d_1, ..., d_q)$. Additionally, a joint $q \times q$ "degree matrix" M is given. A graphical realization of the triple $(\boldsymbol{V}, M, \boldsymbol{d})$ is a simple graph G with vertex set $V = \cup_{i=1}^{q} V_i$ such that every vertex $u \in V_i$ has degree d_i for each $1 \le i \le q$, and the number of edges with one end in V_i and the other end in V_j is precisely M_{ij} for every $1 \le i \le j \le q$. Without loss of generality, we may assume that $d_i \ge 1$ for every $1 \le i \le q$. We may also assume that

(a) M is symmetric, and

(b) $2M_{ii} + \sum_{j:j\neq i} M_{ij} = d_i|V_i|$ for every $1 \le i \le q$, and

(c) $\sum_{i=1}^{n} d_i|V_i|$ is even,

as otherwise, there is no graphical realization for $(\boldsymbol{V}, M, \boldsymbol{d})$ (indeed, (a) and (b) above imply (c)). The goal of this research is to uniformly generate a graphical realization conforming to a given $(\boldsymbol{V}, M, \boldsymbol{d})$.

Following the proposal of the problem, Amanatidis, Green, and Mihail [1] presented a simple polynomial algorithm that produces a graphical realization of a given joint degree matrix, although it is not uniformly generated. They also considered the switch chain as a potential method to solve the problem, but they were uncertain about how to bound the mixing time. In [9], Erdös, Miklós, and Toroczkai prove a suitable MCMC algorithm is rapid mixing over a subspace of realizations of a given JDM (namely the space of balanced realizations). Later, Amanatidis and Kleer [2] demonstrated rapid mixing for strongly stable degree sequences, which include the cases of 2-degree classes (i.e., $q = 2$) in the context of JDM. However, they noted that their proof might not be applicable to cases with $q > 2$.

In this thesis, we concentrate on a different approach to graph generation problems, as discussed earlier, using switching-based methods. Compared to prior work [2], our method provides better running time and works for any bounded number of degree classes. However, we do require some additional restrictions on the input parameters $(\boldsymbol{V}, M, \boldsymbol{d})$.

In the subsequent chapters, we'll first provide a high-level overview of the switching-based algorithm developed by McKay and Wormald in [16], while also introducing the new ideas used in this thesis in Chapter 2. Then we present the essential aspects of the configuration model and discuss its adaptation for the graphical realizations of a joint degree matrix in Chapter 3. Following this, we demonstrate the generation of an initial multi-graph for our problem in Chapter 4. For a comprehensive understanding of our problem-solving process, we depict the base case with 2-degree classes in Chapter 5. We introduce new types of switching operations and show their use in removing multi-edges and loops in the initial graph. In Chapter 6, we extend our problem to a general case, which mostly aligns with the approach detailed in Chapter 5. Finally, we explore potential enhancements to our method in Chapter 7.

Chapter 2

Overview of switching-based algorithm

In this chapter, we provide a general description of the switching-based algorithm designed by McKay and Wormald in [16], along with some improvements made to the algorithm. Subsequently, we will demonstrate how to adapt their algorithm to address our problem and clarify the new ideas presented in this thesis.

2.1 Overview

The algorithm starts by generating a multigraph G with the given degree sequence $\boldsymbol{d} = \{d_1, d_2, \ldots, d_n\}$ by using the configuration model introduced by Bollobás [6]. Here, we aim for the generated graph G to be in a specific set $\mathcal{G}_0$, where $\mathcal{G}_0$ is a predefined set based on the degree sequence $\boldsymbol{d}$ such that graphs in $\mathcal{G}_0$ share the same degree sequence $\boldsymbol{d}$ and satisfy the following conditions: they do not contain multi-edges of multiplicity greater than two or loops of multiplicity greater than one. Additionally, they have at most d double edges and l loops, where the values of d and l are determined by the sequence $\boldsymbol{d}$. If G fails to meet these conditions, the algorithm will terminate and reject G, which we refer to as an initial rejection. In Section 3.1, we will provide a detailed introduction to the configuration model and demonstrate how it can be employed to generate multi-graphs that satisfy any valid degree sequence.

Next, two phases are employed to process the graph G with the goal of removing loops and double edges from it. The first phase is dedicated to eliminating all loops from G, while the second phase focuses on the removal of all double edges within G. In these phases, two operations designed by McKay and Wormald are utilized. The first operation, called l-switching, is used to remove loops, while the second, called d-switching, is used to remove double edges. The general idea behind these switching operations is to eliminate non-simple structures from the graph by adding and removing edges while preserving the degree of each vertex. Detailed explanations of these two operations will be provided in Section 5.1.

Throughout these two phases, specific rejection schemes are designed to ensure that uniformity is maintained throughout the process. To illustrate the general idea behind the design of these rejection schemes, we will use the second phase, which is dedicated to removing double edges from the graph, as an example. Suppose we obtain a graph G_1 after the first phase. In this graph, there are no loops, and it contains at most d double edges. We define $S_k \subseteq \mathcal{G}_0$ to be the set of multigraphs containing no loops and exactly k double edges, where $0 \leq k \leq d$. Hence, $G_1 \in S_k$ for some k. Subsequently, we proceed to perform d-switching on G_1 sequentially. Each d-switching reduces the number of double edges in G_1 by one. In other words, after applying a single d-switching operation to G_1, we obtain a graph in S_{k-1}. After two d-switching operations, we end up with a graph in S_{k-2}, and so on. This process continues until we have reduced the number of double edges in the graph to zero, which means we will end up with a graph in S_0.

If G_1 is uniformly chosen from S_k for some k, then applying a d-switching uniformly at random to G may result in a graph that is not uniformly distributed in S_{k-1}. This is because the number of ways to apply a d-switching to each graph in S_k is different, and the number of ways to generate each graph in S_{k-1} by a d-switching is also different.

To resolve this issue, two rejection steps are designed. We now briefly introduce these two steps and show how they preserve uniformity. Suppose G_1 is uniformly chosen from S_k for some k, then we uniformly at random choose some d-switching that can be applied on G_1 and suppose it produces $G_1' \in S_{k-1}$. Let's define the following parameters:

(a) Let $f(G)$ be the number of ways to perform a d-switching on graph G.

(b) Let $b(G)$ be the number of ways to generate G through a d-switching.

Additionally, two parameters $\overline{f}_k$ and $\underline{b}_k$ are chosen such that for each $1 \leq k \leq d$,

$$\overline{f}_k \geq \max_{G \in S_k} f(G), \quad \underline{b}_k \leq \min_{G \in S_k} b(G).$$

A simplest choice would be $\overline{f}_k = \max_{G\in S_k} f(G)$ and $\underline{b}_k = \min_{G\in S_k} b(G)$. However, computation of $\max_{G\in S_k} f(G)$ and $\min_{G\in S_k} b(G)$ can take a long time. Instead, we can specify $\overline{f}_k$ and $\underline{b}_k$ as some function of $\boldsymbol{d}$ and k and prove that they satisfy the conditions above.

Recall that we perform some d-switching on G_1 and produce G_1'. Next, the f-rejection rejects with probability $1 - \frac{f(G)}{\overline{f}_k}$ and the b-rejection rejects with probability $1 - \frac{\underline{b}_{k-1}}{b(G')}$. The probability that $G_1 = G$ is the same for every $G \in S_k$. For any $G' \in S_{k-1}$, the event $G_1' = G$ occurs, if $G_1 = G$ for some $G \in S_k$, and a d-switching S that coverts G to G' is selected by the algorithm. Hence,

$$\mathbb{P}(G_1' = G') = \sum_{(G,S)} \mathbb{P}(G_1 = G)\frac{1}{f(G)} \cdot \frac{f(G)}{\overline{f}_k}\frac{\underline{b}_{k-1}}{b(G')},$$

where the summation is over all (G, S) such that S converts G to G'. Note that $\frac{1}{f(G)}$ is the probability that S was selected, $\frac{f(G)}{\overline{f}_k}\frac{\underline{b}_{k-1}}{b(G')}$ is the probability that S is neither f-rejected, nor b-rejected. Since the number of (G, S) such that S converts G to G' is equal to $b(G')$, and $\mathbb{P}(G_1 = G)$ is a constant, $\mathbb{P}(G_1' = G)$ does not depend on G'. Hence, G_1' is uniform in S_{k-1}.

By applying this process inductively, the resulting graph after removing all double edges from G_1 is uniformly distributed in S_0.

In the process described above, the only term that cannot be computed in constant time is the calculation of the values of $f(G)$ and $b(G')$. However, based on the switching selection strategy, it is shown that there is no need to compute the value of $f(G)$. Consequently, the only computation left to perform is to determine the value of $b(G')$. This computation involves counting specific local structures within G', which can be efficiently executed.

To improve the algorithm's running time, Arman, Gao, and Wormald [3] introduced the technique of incremental relaxation by modifying the b-rejection steps during the phase of removing double edges and loops from the graph. To briefly explain the concept of incremental relaxation, let's consider the example above. As we said, the majority of running time is spent on computing the value of $b(G')$ in each iteration. Upon examining the structure of d-switching, $b(G')$ is the number of two vertex disjoint simple ordered two-paths $u_1v_1w_1, u_2v_2w_2$ in G' such that there are no edges between u_1 and u_2, v_1 and v_2, as well as w_1 and w_2. The computation of $b(G')$ involved enumerating all pairs of simple ordered 2-paths and verifying their availability, which is a time-consuming process. However, incremental relaxation breaks down the b-rejection into multiple rejection steps, with each step focusing on counting simpler structures than those considered in the calculation of $b(G')$. As a result, this significantly reduces the computation time in each iteration. For a more comprehensive explanation, readers can refer to Section 3 of [3] for more details.

2.2 Adaptation

The main difference in our problem is that we have a joint degree matrix M, which regulates the number of edges within and between vertex sets. As a result, we've made specific adjustments to both the initial graph generation and the removal of loops and double edges for our problem.

For the initial generation, we made slight modifications to the procedure for generating multigraphs using the configuration model. These modifications ensure that the number of edges within and between vertex sets satisfies the requirements imposed by the joint degree matrix M. Similarly, the set $\mathcal{G}_0$ is predefined based on the input parameters $(\boldsymbol{V}, M, \boldsymbol{d})$. Within $\mathcal{G}_0$, graphs must satisfy all the requirements for being a graphical realization of $(\boldsymbol{V}, M, \boldsymbol{d})$, except they may contain loops and multi-edges. Furthermore, the graphs in $\mathcal{G}_0$ must meet the following requirements:

(a) They should not contain multi-edges of multiplicity greater than two or loops of multiplicity greater than one.

(b) They should have at most l_i loops within V_i for $i \in [q]$.

(c) They should have at most d_{ij} double edges with one endpoint in V_i and the other in V_j for $1 \leq i \leq j \leq q$.

The values of l_is and d_{ij}s are determined by $(\boldsymbol{V}, M, \boldsymbol{d})$. The procedure for generating the initial graph for our problem will be provided in Section 4.1.

After obtaining an initial graph G from $\mathcal{G}_0$, we then remove double edges and loops from it. For our problem, we divide the process into several phases, each phase is dedicated to removing double edges or loops within or between particular vertex sets. For example, we address double edges with one endpoint in V_1 and the other in V_2, double edges with both endpoints in V_3, loops within V_4, etc.

In order to remove loops and double edges from G, the most intuitive idea is to apply the l-switching and d-switching designed by McKay and Wormald. However, directly applying these switchings may result in a relatively high rejection probability and lead to poor algorithm performance. We will provide a detailed explanation of why this can lead to a high rejection probability in Section 5.1, after introducing the l-switching and d-switching. To reduce the rejection probability, we draw on ideas from [10] by designing various types of switchings for each phase. Due to the requirements imposed by the joint degree matrix M, these switchings not only ensure that the degree of each vertex remains unchanged before and after the switching but also maintain the same number of edges within or between any vertex sets. We will provide examples of switchings used for two vertex sets (i.e. $q = 2$) in Section 5.2, with the general definitions of the switchings presented in Section 6.1. Furthermore, similar rejection schemes are designed to maintain uniformity in each phase.

Chapter 3

Preliminaries and main results

3.1 Configuration Model

The configuration model, also known as the pairing model in the literature, was originally introduced by Bollobás [6] in order to asymptotically enumerate graphs with a given degree sequence. Soon it became the indispensable tool for analysing random graphs with given degree sequences. Given a degree sequence $\boldsymbol{d} = \{d_1, d_2, \ldots, d_n\}$, the configuration model creates n cells, denoted by $C_1, \ldots, C_n$, where the i-th cell C_i contains exactly d_i points. Take a uniformly random matching P over the total $\sum_{i=1}^{n} d_i$ points. Note that for a graphical degree sequence $\boldsymbol{d}$, $\sum_{i=1}^{n} d_i$ must be even. The matching P is called a paring, and every pair of points matched by P is called a pair in P. Let $G(P)$ be the (multi)graph produced by P by representing each cell as a vertex, and each pair in P as an edge (See Figure 3.1 as an example). It is easy to show, by a simple counting argument, that $G(P)$ is a uniformly random simple graph, conditioned on that $G(P)$ is simple.

To facilitate the discussion, we introduce the following terminology for graphs and the configuration model:

1. **Cells** and **Vertices**: The term **cells** is used exclusively in the configuration model, while **vertices** is used in the graph. Each cell corresponds to a unique vertex where the number of points the cell contains is the same as the degree of its corresponding vertex.

2. **Pairs** and **Edges**: The term **pairs** is used exclusively in the configuration model, while **edges** is used in the graph. Each pair corresponds to a unique edge, or part of a multi-edge, in the graph.

Here is an example of the configuration model for the degree sequence $\{1, 2, 3, 2\}$ with a pairing, along with its corresponding graph:

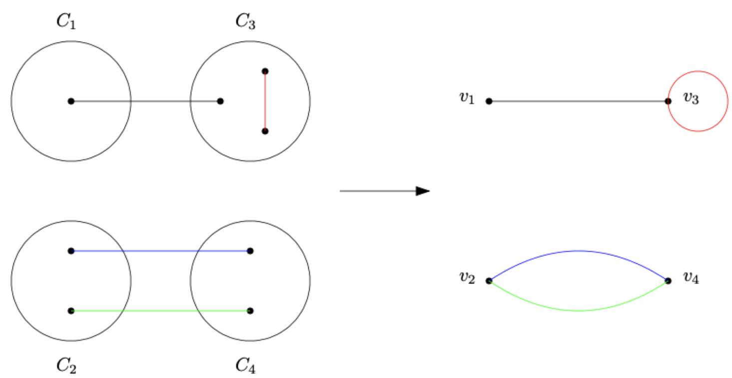


Figure 3.1: Example of the configuration model

As our algorithm will operate on multigraphs, we give a formal definition of it.

Definition 3.1. A multigraph G is defined by a triple-element set (V, E, M) where V is the set of vertices, $E \subseteq \binom{V}{2} \cup V$ is the set of multi-edges, and $M : E \to \mathbb{N} \setminus \{0\}$ denotes the multiplicities of the multi-edges in E. A multi-edge uv contributes $M(uv)$ to the degrees of u and v respectively. A loop at u contributes $2M(u)$ to the degree of u.

This definition aligns with that of Diestel [8], if one considers the isomorphism class of different edge labelings. In Diestel's definition, each multi-edge $uv \in E$ is replaced by $M(uv)$ distinct edges, all connecting vertices u and v. However, for the purposes of this thesis, it's more straightforward to consider a multi-edge as a single entity. Specifically, we refer to a multi-edge uv with a multiplicity of 2 as a *double* edge.

Bollobás' configuration model naturally extends to generate graphs realising $(\boldsymbol{V}, M, \boldsymbol{d})$ as follows. Let $\mathcal{C}_i$, where $1 \leq i \leq q$, be a set of $|V_i|$ cells representing vertices in V_i. Each cell in $\mathcal{C}_i$ contains exactly d_i points. Let $\Phi(\boldsymbol{V}, M, \boldsymbol{d})$ be the set of all pairings P on the set of $\sum_{i=1}^{q} d_i|V_i|$ points such that there are exactly M_{ij} pairs with one end contained in a cell in $\mathcal{C}_i$ and the other end contained in a cell in $\mathcal{C}_j$, for all $1 \leq i \leq j \leq q$. As before, let $G(P)$ denote the multigraph produced by P for any $P \in \Phi(\boldsymbol{V}, M, \boldsymbol{d})$. Again, by a simple counting argument (see the Appendix for a proof), $G(P)$ is a uniformly random graphical realization for $(\boldsymbol{V}, M, \boldsymbol{d})$, if P is a uniformly random pairing in $\Phi(\boldsymbol{V}, M, \boldsymbol{d})$ conditional on

$G(P)$ being simple. Figure 3.2 illustrates an example of the graphical realization (and its corresponding pairing) of $((V_1, V_2), \begin{pmatrix} 3 & 3 \\ 3 & 0 \end{pmatrix}, (3,1))$, where $|V_1| = |V_2| = 3$.

In the pairing on the left-hand side of the figure, cells C_1 through C_6 correspond to vertices v_1 through v_6 in the graph on the right-hand side of the figure. Specifically, we have $\mathcal{C}_1 = \{C_1, C_2, C_3\}$ and $\mathcal{C}_2 = \{C_4, C_5, C_6\}$. Additionally, U_i in the figure denotes the set of points contained within the cells in $\mathcal{C}_i$, where U_1 contains nine points and U_2 contains three points.

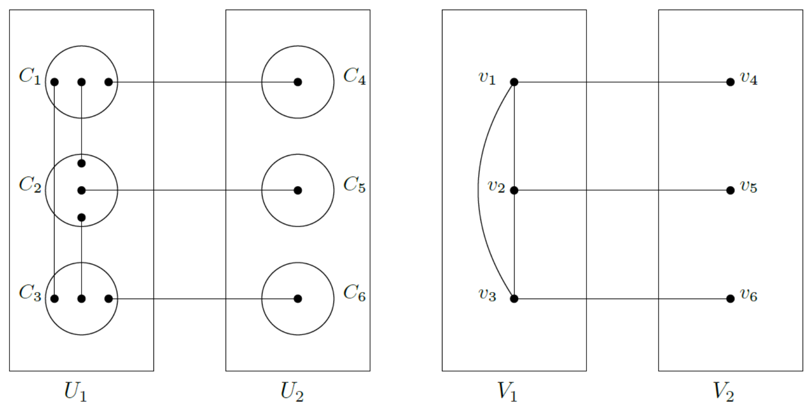


Figure 3.2: graphical realization and its corresponding pairing

Let P be a uniformly random pairing in $\Phi(\boldsymbol{V}, M, \boldsymbol{d})$. If $G(P)$ is simple then we are done. However, unless each of the d_i is relatively small, the probability that $G(P)$ is simple is very close to zero, and thus, we almost always end up with a multigraph $G(P)$. The first step of our algorithm is to generate P from a reasonably large subset $\Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$ of $\Phi(\boldsymbol{V}, M, \boldsymbol{d})$. To define $\Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$ we need to define a few parameters.

Let $\mathcal{G}(\boldsymbol{V}, M, \boldsymbol{d}) = \{G(P) : P \in \Phi(\boldsymbol{V}, M, \boldsymbol{d})\}$. For any $G \in \mathcal{G}(\boldsymbol{V}, M, \boldsymbol{d})$, we define $L(G) = (L_i(G))_{i \in [q]}$, where $L_i(G)$ is the number of simple loops in G whose ends are in V_i for $i \in [q]$. Additionally, define $D(G) = (D_{ij}(G))_{1 \le i \le j \le q}$, where $D_{ij}(G)$ is the number of double edges in G that join a vertex in V_i and a vertex in V_j.

Let $n_i = |V_i|$ for each $i \in [q]$ and set $\overline{L} = (\overline{L}_i)_{i \in [q]}$ and $\overline{D} = (\overline{D}_{ij})_{1 \le i \le j \le q}$ where

$$\overline{L}_i = \frac{2q^2 M_{ii}}{n_i} \quad \text{for } i \in [q]; \quad \overline{D}_{ij} = \frac{2q^2 M_{ij}^2}{n_i n_j} \quad \text{for } 1 \le i \le j \le q. \tag{3.1}$$

Let $\Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$ be the set of pairings P in $\Phi(\boldsymbol{V}, M, \boldsymbol{d})$ such that $L(G(P)) \le \overline{L}$, $D(G(P)) \le \overline{D}$, and $G(P)$ has no loops of multiplicities greater than one, or multiple edges of multiplicity greater than two. The following lemma ensures that $\Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$ is reasonably large compared to $\Phi(\boldsymbol{V}, M, \boldsymbol{d})$.

Lemma 3.2. *Let P be a uniformly random pairing in $\Phi(\boldsymbol{V}, M, \boldsymbol{d})$. Provided that*

$$\frac{M_{ij}^3 \Delta^2}{n_i^2 n_j^2} = o(1) \text{ for all } 1 \le i \le j \le q, \tag{3.2}$$

then,

$$\Pr(P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})) \ge \frac{3}{8} + o(1).$$

The proof of the lemma is a simple first moment argument and is deferred to Section 4.2.

In this thesis, we introduce a new algorithm called JDM_GEN that generates a uniformly random graphical realization of $(\boldsymbol{V}, M, \boldsymbol{d})$. The algorithm starts by repeatedly generating a uniformly P in $\Phi(\boldsymbol{V}, M, \boldsymbol{d})$ until P is in $\Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$. Lemma 3.2 above shows that $O(1)$ rounds is necessary in expectation. Let $G_0 = G(P)$ where P is the pairing obtained. We know that P is a uniformly random pairing in $\Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$. Next, our algorithm starts from G_0, and produces a sequence of multigraphs $G_0, G_1, G_2, \ldots$ using some switching operations defined in Section 5.1. These switching operations repeatedly replace multiple edges and loops with simple edges while maintaining the resulting multigraph in $\mathcal{G}(\boldsymbol{V}, M, \boldsymbol{d})$. Once all the multiple edges are removed, the resulting simple graph is output. However, each application of the switching operations changes the distribution of G_t slightly from uniform. The central work is to design and combine the use of a set of switching operations, together with a carefully crafted rejection scheme, to maintain the uniformity in each step of the algorithm. Whenever a rejection occurs, the algorithm restarts from the beginning. Thus, in order to control the running time of the algorithm, it is necessary that the overall rejection probability is not too big. This further imposes certain constraints on $(\boldsymbol{V}, M, \boldsymbol{d})$ in our main theorem.

We will discuss the uniformity of G_0 in Section 4.1, and examine the running time of generating G_0 in Section 4.2. In Section 5.3.3, we will explore how uniformity is maintained during the process of removing multiple edges and loops where we will use a specific phase that focuses on removing crossing double edges between two vertex sets as an example, and the time required for this process will be discussed in Section 5.4.1.

The main theorem of this thesis is stated as follows:

Theorem 3.3. JDM_GEN *generates a uniformly random graphical realization of* $(\boldsymbol{V}, M, \boldsymbol{d})$, *where* $\boldsymbol{V} = (V_1, V_2, \dots, V_q)$, $M = (M_{ij})_{i,j\in[q]}$ *and* $\boldsymbol{d} = (d_1, d_2, \dots, d_q)$ *are given and* $q \geq 2$ *is a fixed constant. Provided that*

$$\frac{M_{ij}^3 \Delta^2}{n_i^2 n_j^2} = o(1) \text{ for all } 1 \leq i \leq j \leq q,$$

where $\Delta = \max\{d_i : i \in [q]\}$, *and* $n_i = |V_i|$ *for* $i \in [q]$, *the expected running time of* JDM_GEN *is*

$$O\left(\sum_{i\in[q]} n_i d_i\right).$$

Corollary 3.4. JDM_GEN *generates a uniformly random graphical realization of* $(\boldsymbol{V}, M, \boldsymbol{d})$, *where* $\boldsymbol{V} = (V_1, V_2, \dots, V_q)$, $M = (M_{ij})_{i,j\in[q]}$ *and* $\boldsymbol{d} = (d_1, d_2, \dots, d_q)$ *are given and* $q \geq 2$ *is a fixed constant. Provided that all* $|V_i|$ *for* $i \in [q]$ *are of the same asymptotic order* n, *all* d_i *for* $i \in [q]$ *are of the same asymptotic order* d, *and*

$$d^5 = o(n),$$

the expected running time of JDM_GEN *is*

$$O(nd).$$

Furthermore, when all entries of M are of the same asymptotic order, we can improve the condition stated in the Theorem 3.3, yielding the following result.

Theorem 3.5. JDM_GEN *generates a uniformly random graphical realization of* $(\boldsymbol{V}, M, \boldsymbol{d})$, *where* $\boldsymbol{V} = (V_1, V_2, \dots, V_q)$, $M = (M_{ij})_{i,j\in[q]}$ *and* $\boldsymbol{d} = (d_1, d_2, \dots, d_q)$ *are given and* $q \geq 2$ *is a fixed constant. Provided that* M_{ij} *are of the same asymptotic order* m *for all* $1 \leq i \leq j \leq q$ *and*

$$\frac{m\Delta^2}{n_i n_j} = o(1) \text{ for all } 1 \leq i \leq j \leq q,$$

where $\Delta = \max\{d_i : i \in [q]\}$, *and* $n_i = |V_i|$ *for* $i \in [q]$, *the expected running time of* JDM_GEN *is*

$$O\left(\sum_{i\in[q]} n_i d_i\right).$$

Corollary 3.6. JDM_GEN *generates a uniformly random graphical realization of* $(\boldsymbol{V}, M, \boldsymbol{d})$, *where* $\boldsymbol{V} = (V_1, V_2, \dots, V_q)$, $M = (M_{ij})_{i,j \in [q]}$ *and* $\boldsymbol{d} = (d_1, d_2, \dots, d_q)$ *are given and* $q \geq 2$ *is a fixed constant. Provided that* M_{ij} *are of the same asymptotic order for all* $1 \leq i \leq j \leq q$, *all* $|V_i|$ *for* $i \in [q]$ *are of the same asymptotic order* n, *all* d_i *for* $i \in [q]$ *are of the same asymptotic order* d, *and*

$$d^3 = o(n),$$

the expected running time of JDM_GEN *is*

$$O(nd).$$

Chapter 4

Initialization

In this chapter, we present the details in the procedure for the generation of the initial graph, denoted as G_0, and discuss the distribution of G_0.

4.1 Generation of G_0

As discussed earlier, the algorithm finds $G(P)$ where P is a uniformly random pairing in $\Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$. Recall that $\mathcal{C}_i$ denotes the set of cells representing the vertices in V_i, for each $i \in [q]$. Let U_i denote the set of points in the cells of $\mathcal{C}_i$. The procedure for generating a uniformly random pairing P in $\Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$ is given below. Recall that when a rejection occurs, the algorithm restarts from the beginning.

procedure PAIRINGGEN($\boldsymbol{V}, M, \boldsymbol{d}$)
  for each $i \in [q]$ **do**
    Uniformly at random partition U_i into q parts, denoted by $\{X_{ij}, j \in [q]\}$, subject to $|X_{ij}| = M_{ij}$ for each $j \neq i$
  end for
  for each $1 \leq i < j \leq q$ **do**
    Match the points in X_{ij} to the points in X_{ji} uniformly at random.
  end for
  for each $i \in [q]$ **do**
    Take a uniform random perfect matching over the points in X_{ii}
  end for
  Let P be the pairing generated by the above steps; reject P if $P \notin \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$.
  Return $(P, G(P))$
end procedure

We first verify that the pairing output by PAIRINGGEN is uniformly distributed in $\Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$.

Lemma 4.1. *Each $P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$ is output by* PAIRINGGEN *with equal probability.*

Proof. For each $i \in [q]$, let

$$N_i = |U_i| = n_1 d_1.$$

Claim 4.2. Each pairing $P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$ is output by PAIRINGGEN with the probability:

$$\left(\prod_{i=1}^{q} \frac{1}{\binom{N_i}{M_{i1},\dots,M_{ii-1},2M_{ii},M_{ii+1},\dots,M_{iq}}}\right) \cdot \left(\prod_{1\leq i<j\leq q} \frac{1}{(M_{ij})!}\right) \cdot \left(\prod_{i=1}^{q} \frac{(M_{ii})!2^{M_{ii}}}{(2M_{ii})!}\right). \tag{4.1}$$

Since 4.1 only depends on $(\boldsymbol{V}, M, \boldsymbol{d})$, this completes the proof. □

Proof of Claim 4.2. For each $i \in [q]$, there are $\binom{N_i}{M_{i1},\dots,M_{ii-1},2M_{ii},M_{ii+1},\dots,M_{iq}}$ different ways to partition N_i points into q subsets $X_{i1}, ..., X_{iq}$ with sizes $M_{i1}, \dots, M_{ii-1}, 2M_{ii}, M_{ii+1}, \dots, M_{iq}$, respectively.

For any $1 \leq i < j \leq q$, once the partitions X_{ij} and X_{ji} are determined, there are $(M_{ij})!$ different perfect matchings that can be created between X_{ij} and X_{ji}.

Furthermore, for each $i \in [q]$, when the partition X_{ii} is determined, there are $\frac{|X_{ii}|!}{\frac{|X_{ii}|}{2}!2^{\frac{|X_{ii}|}{2}}}$, which simplifies to $\frac{(2M_{ii})!}{(M_{ii})!2^{M_{ii}}}$, ways to create perfect matchings on X_{ii}.

Hence, by following the steps in PairingGen, each $P \in \Phi(\boldsymbol{V}, M, \boldsymbol{d})$ is generated with the probability demonstrated by 4.1. Since the procedure rejects P if $P \notin \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$, each $P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$ is output by PairingGen with the probability demonstrated by 4.1.

□

Let G_0 be the $G(P)$ output by PairingGen. Given that P is uniformly distributed in $\Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$, it is not true that $G(P)$ is uniformly distributed in $\{G(P) : P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})\}$. However, we show that $G(P)$ is uniformly distributed in a certain conditional probability space. We need a few definitions to formalize this. Recall that for any $G \in \mathcal{G}(\boldsymbol{V}, M, \boldsymbol{d})$, we denote $L(G) = (L_i(G))_{i\in[q]}$ as a q-tuple, where $L_i(G)$ represents the number of simple loops whose ends are in V_i. Similarly, $D(G) = (D_{ij}(G))_{1\le i\le j\le q}$ has been defined as a q^2-tuple, where $D_{ij}(G)$ denotes the number of double edges in G that have one end in V_i and the other end in V_j for $1 \le i \le j \le q$.

Given $\boldsymbol{L} = (L_1, L_2, ..., L_q)$ and $\boldsymbol{D} = (D_{ij})_{1\le i\le j\le q}$, let $\mathcal{G}_0(\boldsymbol{D}, \boldsymbol{L})$ denote the set of multigraphs in $\{G(P) : P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})\}$ such that $L(G) = \boldsymbol{L}, D(G) = \boldsymbol{D}$. We show that G_0 has uniform distribution after conditioning on $L(G_0)$ and $D(G_0)$.

Lemma 4.3. *Conditioning on $L(G_0) = \boldsymbol{L}, D(G_0) = \boldsymbol{D}, G_0$ is uniform in $\mathcal{G}_0(\boldsymbol{D}, \boldsymbol{L})$.*

Proof. Let $(P^*, G(P^*))$ denote the output of the PairingGen procedure.

For any $G \in \mathcal{G}_0(\boldsymbol{D}, \boldsymbol{L})$, the probability of obtaining $G_0 = G$ is given by:

$$\Pr(G_0 = G) = \sum_{P\in\Phi_0(\boldsymbol{V},M,\boldsymbol{d}),G(P)=G} \Pr(P^* = P).$$

Each $P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$ is generated with the same probability by Lemma 4.1. Therefore, it suffices to prove that for each $G \in \mathcal{G}_0(\boldsymbol{D}, \boldsymbol{L})$, the cardinality of the set $\{P : P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d}), G(P) = G\}$ is the same.

Given $\boldsymbol{D} = (D_{ij}(G))1 \le i, j \le q$ and $\boldsymbol{L} = (L_i(G))i \in [q]$, it is easy to see that for any $G \in \mathcal{G}_0(\boldsymbol{D}, \boldsymbol{L})$:

$$|\{P : P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d}), G(P) = G\}| = \prod_{i=1}^{q} (d_i!)^{|V_i|} \cdot \left(\frac{1}{2}\right)^{\sum_{i=1}^{q} L_i+\sum_{1\le i\le j\le q} D_{ij}} . \tag{4.2}$$

(We include a proof for (4.2) in Appendix A.) Hence, the size of the set only depends on $\boldsymbol{L}$ and $\boldsymbol{D}$, which completes the proof. $\square$

4.2 Running time of PairingGen and Proof of Lemma 3.2

Recall $\overline{L}$ and $\overline{D}$ in (3.1). Given a nonnegative integer t, let $[x]_t$ denote the t-th falling factorial of x; i.e. $[x]_t = x(x-1)...(x-t+1)$. Moreover, let

$$N_i = |U_i| = n_i d_i \quad \text{for } i \in [q],$$
$$N_i^* = N_i - \sum_{j\in[q], j\neq i} M_{ij} = 2M_{ii} \quad \text{for } i \in [q],$$

and recall that $n_i = |V_i|$ and $\Delta = \max\{d_i : i \in [q]\}$.

By analogous proofs as [16, **Lemma 1**] and [16, **Lemma 2**] we obtain the following two simple lemmas. We include their proofs for completeness.

Lemma 4.4. *Let P be a uniform random pairing in $\Phi(\boldsymbol{V}, M, \boldsymbol{d})$.*

(a) The probability of t_{ij} given pairs which have one endpoint in U_i and another in U_j occurring in P is

$$\frac{[M_{ij}]_{t_{ij}}}{[N_i]_{t_{ij}}[N_j]_{t_{ij}}} = (1+o(1))\left(\frac{M_{ij}}{N_i N_j}\right)^{t_{ij}}$$

if $t_{ij} = o(\sqrt{M_{ij}})$.

(b) The probability of t_i given pairs whose both endpoints are in U_i occurring in P is

$$\frac{[M_{ii}]_{t_i} 2^{t_i}}{[N_i]_{2t_i}} = (1+o(1))\left(\frac{2M_{ii}}{N_i^2}\right)^{t_i}$$

if $t_i = o(\sqrt{M_{ii}})$.

Proof. The precise probability of (a) is

$$\frac{\binom{N_i - t_{ij}}{M_{ij}-t_{ij}}\binom{N_j - t_{ij}}{M_{ij}-t_{ij}} \cdot (M_{ij}-t_{ij})!}{\binom{N_i}{M_{ij}}\binom{N_j}{M_{ij}} \cdot M_{ij}!} = \frac{[M_{ij}]_{t_{ij}}}{[N_i]_{t_{ij}}[N_j]_{t_{ij}}}.$$

For the denominator, $\frac{N_i}{M_{ij}}$ indicates the number of ways to choose M_{ij} points from U_i, which is the number of possible outcomes of X_{ij} in PairingGen. Similarly, $\frac{N_j}{M_{ij}}$ indicates the number of possible outcomes of X_{ij}. And $M_{ij}!$ is the number of possible pairings that can be created between X_{ij} and X_{ji}.

For the numerator, since t_{ij} pairs are already given, it accounts for the number of ways to choose $M_{ij} - t_{ij}$ points from U_i, U_j and pair them up.

The precise probability of (b) when $i = 1$ is

$$\frac{\binom{N_i - 2t_i}{2M_{ii} - 2t_i}}{\binom{N_i}{2M_{ii}}} \cdot \frac{\frac{\binom{N_1^* - 2t_1}{2}\binom{N_1^* - 2t_1 - 2}{2}\cdots\binom{2}{2}}{(\frac{N_1^* - 2t_1}{2})!}}{\frac{\binom{N_1^*}{2}\binom{N_1^* - 2}{2}\cdots\binom{2}{2}}{(\frac{N_1^*}{2})!}} = \frac{[N_1^*]_{2t_1}}{[N_1]_{2t_1}} \cdot \frac{[N_1^*/2]_{t_1} 2^{t_1}}{[N_1^*]_{2t_1}} = \frac{[N_1^*/2]_{t_1} 2^{t_1}}{[N_1]_{2t_1}} = \frac{[M_{11}]_{t_1} 2^{t_1}}{[N_1]_{2t_1}}.$$

For the first term, it represents the probability that the endpoints of given t_i pairs are contained in X_{ii}. The denominator of the second term is the number of all perfect matching that can be taken over the points in X_{ii}. The numerator of the second term indicates the number of all perfect matching, which contains t_i given pairs, that can be taken over the points in X_{ii}. □

Lemma 4.5. *Let $P \in \Phi(\boldsymbol{V}, M, \boldsymbol{d})$ be a uniformly random pairing*

(a) The probability that $G(P)$ contains at least one triple edge between V_i and V_j is at most $O\left(\frac{M_{ij}^3}{n_i^2 n_j^2}\right)$ for all $1 \le i \le j \le q$;

(b) The probability that $G(P)$ contains at least one triple edge in V_i is at most $O\left(\frac{M_{ii}^3}{n_i^4}\right)$ for $i \in [q]$;

(c) The probability that $G(P)$ contains at least one loop of multiplicity at least two in V_i is at most $O\left(\frac{M_{ii}^2}{n_i^3}\right)$ for $i \in [q]$.

Provided that

$$\frac{M_{ij}^3 \Delta^2}{n_i^2 n_j^2} = o(1) \text{ for all } \ 1 \le i \le j \le q, \tag{4.3}$$

all the above probabilities are $o(1)$.

Proof. By Lemma 4.4 we can compute the expected number of triple edges in $G(P)$ between V_i, V_j as following:

$$(1+o(1))(\frac{M_{ij}}{N_iN_j})^3 \cdot 6\binom{d_i}{3}\binom{d_j}{3}n_1n_2 < (1+o(1))(\frac{M_{ij}}{N_iN_j})^3 \cdot 6\frac{d_i^3d_j^3}{36}n_in_j = (1+o(1))\frac{M_{ij}^3}{6n_i^2n_j^2}.$$

Similarly, the expected number of triple edges in $G(P)$ in V_i is:

$$(1+o(1))\left(\frac{2M_{ii}}{N_i^2}\right)^3 \cdot 6\binom{d_i}{3}^2\binom{n_i}{2} = (1+o(1))\left(\frac{2M_{ii}}{N_i^2}\right)^3 \cdot 6\binom{d_i}{3}^2\binom{n_i}{2} < (1+o(1))\frac{2M_{ii}^3}{3n_i^4}.$$

Similarly, the expected number of double loops in $G(P)$ in V_i is:

$$(1+o(1))\left(\frac{2M_{ii}}{N_i^2}\right)^2 \cdot 3\binom{d_i}{4}n_i < (1+o(1))\frac{M_{ii}^2}{2n_i^3}.$$

Then, by employing Markov's Inequality, we can determine three probabilities $(a), (b), (c)$ as stated previously. It's evident that $O\left(\frac{M_{ij}^3}{n_i^2n_j^2}\right) = o(1)$ and $O\left(\frac{M_{ii}^3}{n_i^4}\right) = o(1)$, considering the given assumption. To verify that $O\left(\frac{M_{ii}^2}{n_i^3}\right) = o(1)$, we have:

$$\frac{M_{ii}^2}{n_i^3} \cdot \frac{M_{ii}}{n_i} \cdot \Delta^2 = \frac{M_{ii}^3\Delta^2}{n_i^4}.$$

Let's consider the case $\frac{M_{ii}}{n_i} \neq o(1)$, then we obtain $\frac{M_{ii}^2}{n_i^3} = O\left(\frac{M_{ii}^3\Delta^2}{n_i^4}\right) = o(1)$. However, if $\frac{M_{ii}}{n_i} = o(1)$, then $\frac{M_{ii}^2}{n_i^3} = \left(\frac{M_{ii}}{n_i}\right)^2 \cdot \frac{1}{n_i} = o(1)$. Therefore, the probability that $G(P)$ contains at least one double loop in V_i is also $o(1)$.

$\square$

Proof of Lemma 3.2. Let P be a uniformly random pairing in $\Phi(\boldsymbol{V}, \boldsymbol{M}, \boldsymbol{d})$, we want to show that

$$\Pr(P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})) \geq \frac{3}{8} + o(1),$$

which is equivalent to showing that

$$\Pr(P \notin \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})) \leq \frac{5}{8} + o(1).$$

Let l_i be the number of loops in V_i in $G(P)$ for $i \in [q]$, d_{ij} be the number of double edges in $G(P)$ with one endpoint in V_i and the other endpoint in V_j for $1 \leq i \leq j \leq q$. Then if $P \notin \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$, it might be one of the following cases:

(a) $G(P)$ contains triple edges or double loops.

(b) $l_i > \overline{L}_i$ for some $i \in [q]$.

(c) $d_{ij} > \overline{D}_{ij}$ for some $1 \leq i \leq j \leq q$.

By Lemma 4.5, we have the probability of case (a) is $o(1)$. Next, we compute the probability of case (b) and (c) by employing the similar idea of the proof for Lemma 4.5.

First, we compute the expected value for each l_i and d_{ij}, where we have

$$\begin{aligned}
E(l_i) &< (1+o(1))\left(\frac{2M_{ii}}{N_i^2}\right)\cdot\binom{d_i}{2}n_i < (1+o(1))\frac{M_{ii}}{n_i} \text{ for } i \in [q];\\
E(d_{ii}) &< (1+o(1))\left(\frac{2M_{ii}}{N_i^2}\right)^2\cdot 2\binom{d_i}{2}^2\binom{n_i}{2} < (1+o(1))\frac{M_{ii}^2}{n_i^2} \text{ for } i \in [q];\\
E(d_{ij}) &< (1+o(1))\left(\frac{M_{12}}{N_1N_2}\right)^2\cdot 2\binom{d_1}{2}\binom{d_2}{2}n_1n_2 < (1+o(1))\frac{M_{12}^2}{2n_1n_2} \text{ for all } 1 \leq i < j \leq q.
\end{aligned}$$

Hence we have

$$\begin{aligned}
&\Pr(l_i > \overline{L}_i) < \frac{1}{2q^2} + o(1) \text{ for } i \in [q];\\
&\Pr(d_{ii} > \overline{D}_{ii}) < \frac{1}{2q^2} + o(1) \text{ for } i \in [q];\\
&\Pr(d_{ij} > \overline{D}_{ij}) < \frac{1}{4q^2} + o(1) \text{ for all } 1 \leq i < j \leq q.
\end{aligned}$$

Thus, the probability of case (b) or case (c) occurs can be bounded as following:

$$\begin{aligned}
&\Pr(l_i > \overline{L}_i \text{ for some } i \in [q] \text{ or } d_{ij} > \overline{D}_{ij} \text{ for some } 1 \le i \le j \le q) \\
&< q \cdot \frac{1}{2q^2} + q \cdot \frac{1}{2q^2} + \frac{q(q-1)}{2} \cdot \frac{1}{4q^2} + o(1) \\
&= \frac{1}{q} + \frac{(q-1)}{8q} + o(1) \\
&< \frac{1}{q} + \frac{1}{8} + o(1) \\
&\le \frac{5}{8} + o(1) \text{ for } q \ge 2.
\end{aligned}$$

Together with the probability of case (a) being $o(1)$, we may conclude that

$$\Pr(P \notin \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})) \le \frac{5}{8} + o(1),$$

and it completes the proof. □

Finally, generating P in PairingGen is equivalent to generating uniform random permutations of points in U_i, which can be done in time linear in $|U_i|$. This immediately implies the following.

Lemma 4.6. *Provide that no rejection occurs, the running time of* PairingGen *is* $O\left(\sum_{i=1}^{q} n_i d_i\right)$.

During the execution of PairingGen, we use data structures to store the positions of multi-edges and loops of each time. This data will be useful in the subsequent phases of JDM_GEN.

Chapter 5

Special case when $q = 2$

The case $q = 1$ reduces to the problem of uniform generation of random regular graphs, which has been studied in [3], [10], and [16]. Thus we may assume that $q \geq 2$. For an easier exposition, we first describe our algorithm JDM_GEN for the special case where $q = 2$, i.e. $\boldsymbol{V} = (V_1, V_2)$. In this case, V_1 consists of vertices with degree d_1, V_2 consists of vertices with degree d_2, and the matrix M is a 2×2 matrix that specifies the number of edges between V_1 and V_2 as well as the number of edges within V_1 and V_2. The algorithm extends naturally to the general case where $q \geq 2$ is a fixed integer, which will be discussed in Chapter 6.

Recall that G_0 is the multigraph output by PairingGen. By the definition of $\Phi_0(\boldsymbol{V}, M, \boldsymbol{d})$ and recalling (3.1), G_0 does not contain loops of multiplicity greater than one, or multiple edges with multiplicity greater than two. Moreover, the numbers of loops and double edges in G_0 are bounded from above by $\overline{L}$ and $\overline{D}$, respectively.

As mentioned earlier, our algorithm JDM_GEN generates a sequence of multigraphs $G_0, G_1, \ldots,$ by repeatedly removing loops and double edges in G_0, using a set of "switching operations". These switchings will be formally defined in Section 5.1. In Section 5.3, we will define JDM_GEN for the case $q = 2$ and prove the uniformity of G_t for every $t \geq 1$. In Section 5.4, we will analyze the running time of JDM_GEN. Before proceeding to the definition of the switchings JDM_GEN uses, it is insightful to see how switchings were used in the literature for the case $q = 1$. Hence, we start our discussions by reviewing these switchings.

5.1 Switchings: the old

The switching shown in Figure 5.1 demonstrates how a loop can be removed. The operation shown in the figure replaces the loop on v_1 and two other simple edges v_2v_3 and v_4v_5, by three simple edges, v_1v_2, v_1v_5, and v_3v_4. This switching operation preserves the degree sequence of the graph while reducing the number of loops by 1. Similarly, Figure 5.2 illustrates the elimination of a double edge.

These switchings illustrated in Figure 5.1 and Figure 5.2 were introduced by McKay and Wormald [16]. As mentioned earlier, each application of the switchings distorts the distribution of the resulting multigraph slightly from the uniform. This is due to the fact that some multigraphs permit more number of switchings that can be applied to them than the others, as well as that some multigraphs can be created in more ways via switchings than the others. McKay and Wormald [16] designed reject schemes to correct the distortion of the distribution caused by the application of the switchings, maintaining uniformity of G_t in each step. More recently, Gao and Wormald [10] introduced the "boosting" technique, which complements the switchings in Figure 5.1 and Figure 5.2 by a new set of switchings called "boosters" that are performed only "occasionally". These boosters boost the probability of the multigraphs that are otherwise generated less frequently compared to the others, which would consequently result in a high chance of rejection without the help of boosters. The introduction of the boosting technique significantly broadened the family of degree sequences for which the switching-based generation algorithms can work efficiently.

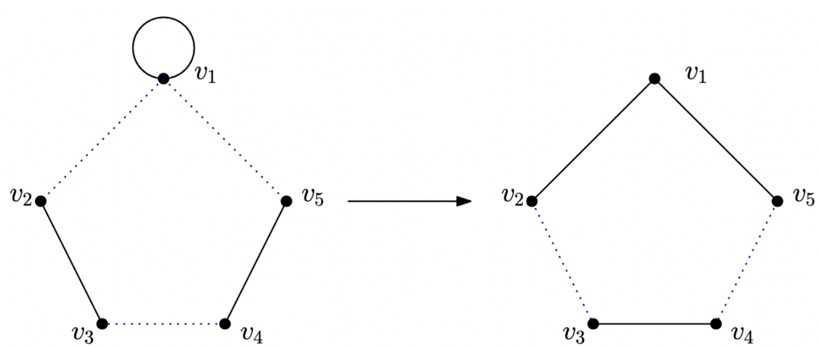


Figure 5.1: l-switching

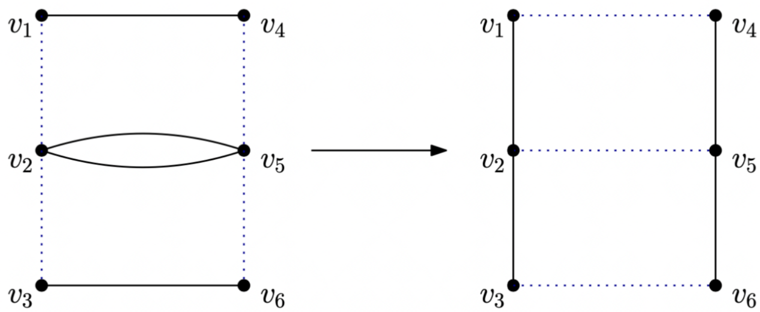


Figure 5.2: d-switching

For the problem we study in this thesis, the most intuitive idea would be to use the two switchings above, with the natural restrictions to maintain the number of edges restricted by M, to remove the loops and double edges in G_0. For instance, if v_2v_5 in Figure 5.2 is a double edge with both ends in V_1, then one would choose the other two edges both lying entirely in V_1. However, a naive application of switchings like this would result in a high overall rejection probability, even for rather small d_1 and d_2.

We give an intuitive explanation of the cause of frequent rejections. Suppose G_0 has d double edges with both ends in V_1. For simplicity and for exposition purposes we assume that G_0 does not have other types of multiple edges or loops. Suppose that G_0 is uniformly distributed in the set S_d of multigraphs with exactly d double edges lying in V_1 (and no other types of multiple edges). Let us consider the multigraphs in S_{d-1}. If G is a multigraph in S_{d-1}, the number of ways that G can be created is equal to the number of pairs of vertex-disjoint 2-paths $v_1v_2v_3$ and $v_4v_5v_6$ lying in V_1 such that v_1v_4, v_2v_5 and v_3v_6 are not edges of G. Let d'_v denote the number of neighbours of v lying in V_1 in G for every $v \in V_1$. Thus, the number of pairs of 2-paths described above is approximately $\rho(G) := (\sum_{v\in V_1} d'_v(d'_v-1))^2$. This graph parameter is larger for some members in S_{d-1}, and smaller for some other members. We have to reject more often for the graphs G with larger $\rho(G)$ in order to maintain uniformity. It is not hard to work out the distribution of $\rho(G)$ for a uniformly random G in S_{d-1}, and we find that $\rho(G)$ is not sufficiently concentrated to guarantee a small rejection probability.

To resolve this problem, our algorithm instead uses a set of switchings including the one in Figure 5.2, allowing G in S_{d-1} to be created by different types of switchings. Although the number of ways that G can be created by the particular type of switching in Figure 5.2 varies significantly for different G in S_{d-1}, the total number of ways that G can be created by any type of switchings the algorithm is permitted to apply will be sufficiently concentrated.

This idea is similar to the use of boosters. However, in all previous work, the boosters are applied very rarely, and they complement a “main type” of switching that is applied almost always in each step.

In our case, all types of switchings are applied with similar frequencies. This signifies a notable distinction from the role of boosters from the previous research.

5.2 Switchings: the new

We formally define the set of switchings used in JDM_GEN. For all the switchings described below, we refer to the operation of transforming a multigraph from the left-hand side to the right-hand side as a forward switching, while the reverse operation is referred to as a backward switching. Additionally, given a graph G and some switching S that can be applied on G, we let $S(G)$ represent the graph that emerges after applying the switching S to G.

5.2.1 Switchings to remove crossing double edges between V_1 and V_2

After the generation of G_0, our algorithm will repeatedly remove all double edges with one end in V_1 and the other end in V_2. For convenience, we call these double edges the *crossing double edges*. After the removal of all crossing double edges, the algorithm proceeds to remove the other types of multiple edges, which are treated in a similar manner. Thus, we start by defining the set of switchings used by the algorithm to remove the crossing double edges.

There are in total 16 different types of switchings that the algorithm uses to remove the crossing double edges. We use D_{12}^i, $1 \le i \le 16$, to denote the names of the types. The letter ”D” indicates that these switchings are designed to eliminate double edges. The subscript ”12” signifies that the crossing double edges have one end in V_1 and the other end in V_2, and the superscripts 1 to 16 represent the 16 different types.

We define the D_{12}^5 type switching in detail, and explain how it is illustrated in Figure 5.3. All the remaining switching types are visually depicted through figures; see Figure 5.5 and Figure 5.6. We omit the lengthy explanations, as their absence will not lead to confusion, given the evident visual illustrations in the figures.

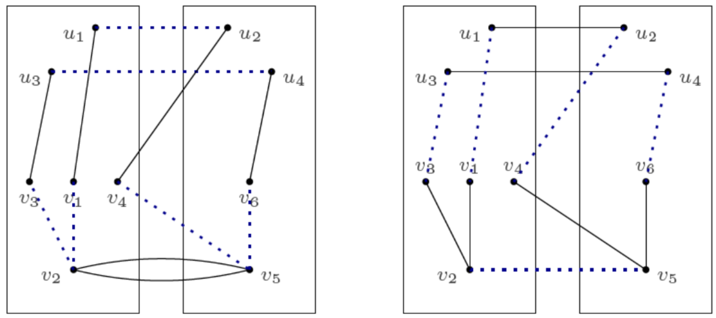


Figure 5.3: D^5_{12} switching

To perform a D^5_{12} type switching, we choose an ordered set of ten vertices

$$(v_1, v_2, v_3, v_4, v_5, v_6, u_1, u_2, u_3, u_4)$$

subject to the following constraints:

(a) $v_2 \in V_1, v_5 \in V_2$ and v_2v_5 induces a crossing double edge;

(b) $v_1, v_3, v_4, u_1, u_3 \in V_1, v_6, u_2, u_4 \in V_2$ where $v_1u_1, v_3u_3, v_4u_2, v_6u_4$ are all simple edges;

(c) $v_1v_2, v_2v_3, v_4v_5, v_5v_6, u_1u_2, u_3u_4$ are all non-edges.

Then, we remove the edges v_1u_1, v_3u_3, v_4u_2, v_6u_4 and the double edge v_2v_5, and add the edges v_1v_2, v_2v_3, v_4v_5, v_5v_6, u_1u_2 and u_3u_4. This operation is illustrated in Figure 5.3. All solid arcs (except for the two arcs between v_2 and v_5) denote simple edges, and all the dashed arcs indicate non-edges. Vertices belonging to different vertex sets of $\boldsymbol{V}$ are put into different square boxes.

It is obvious that the degree sequence and the number of edges between any V_i and V_j, $i, j \in [2]$, do not change after the application of the switchings.

All the 16 switchings are similar, with the main distinction being the placement of v_1, v_3, v_4, and v_6, each of which can be either in V_1 or V_2. This results in a total of $2^4 = 16$ different switchings.

5.2.2 Switchings to remove the other types of multiple edges and loops

Figure 5.4 illustrates the 4 types of switchings that are used to eliminate loops within V_i, $i \in [2]$.

Figure 5.7 and Figure 5.8 illustrate the 16 types of switchings that are used to eliminate the double edges within V_i, $i \in [2]$.

In the figures, the square box on the left-hand side is the vertex set V_i.

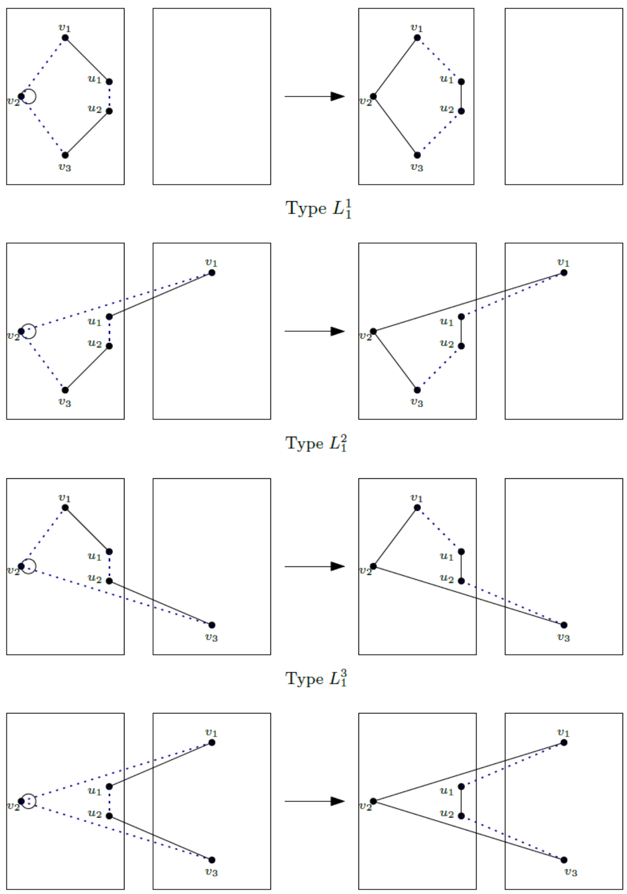
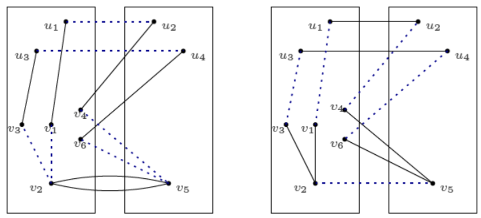
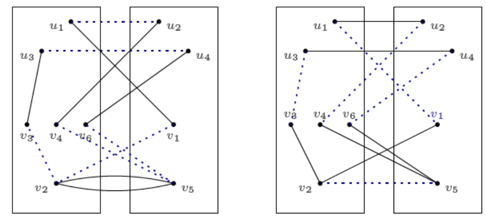


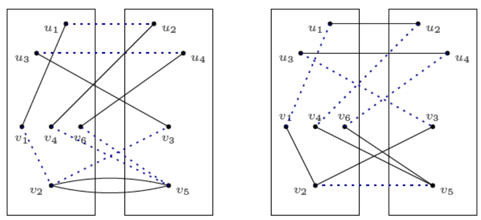
Figure 5.4: L_1 switching 1-4



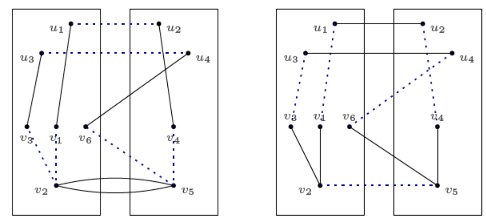
Type D_{12}^1



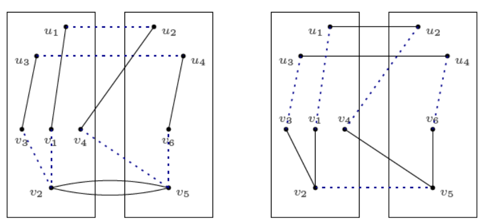
Type D_{12}^2



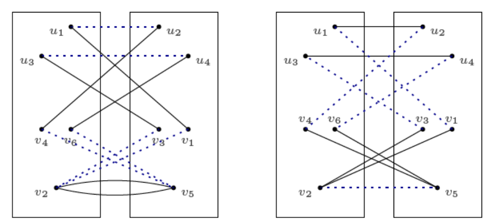
Type D_{12}^3



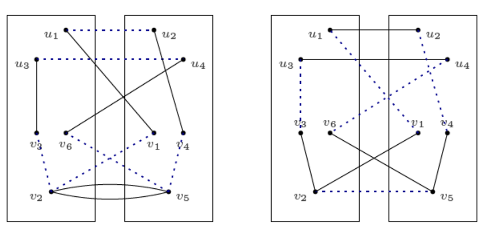
Type D_{12}^4



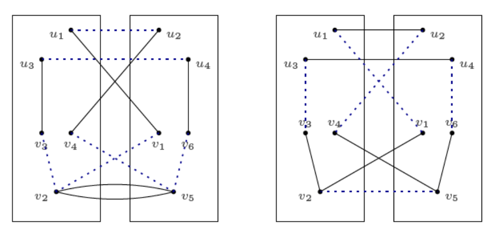
Type D_{12}^5



Type D_{12}^6

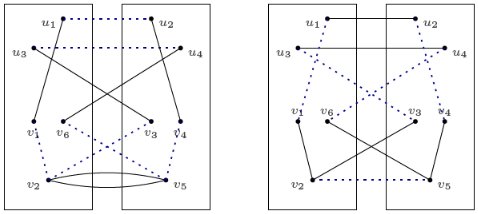


Type D_{12}^7

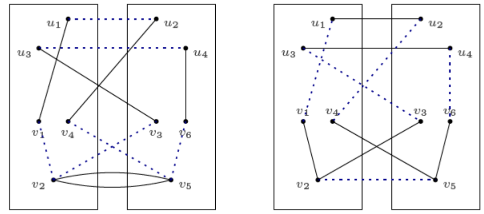


Type D_{12}^8

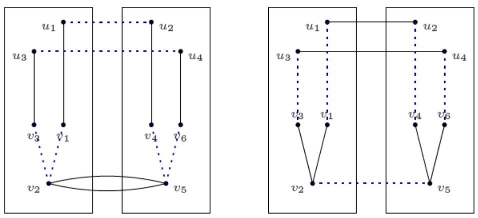
Figure 5.5: D_{12} switching 1-8



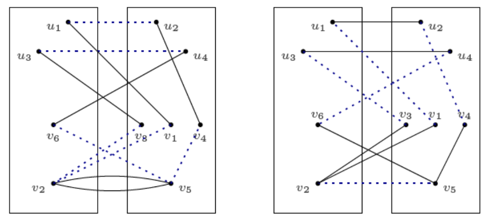
Type D_{12}^{9}



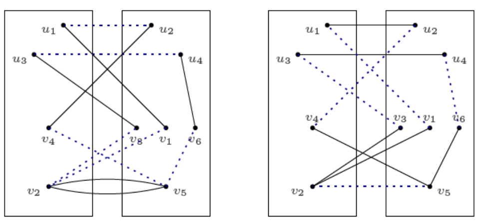
Type D_{12}^{10}



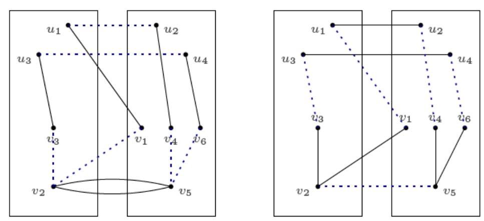
Type D_{12}^{11}



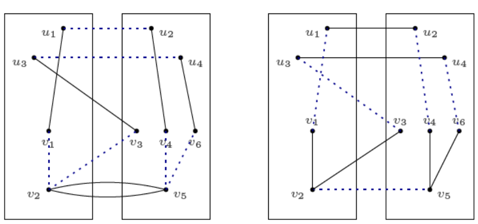
Type D_{12}^{12}



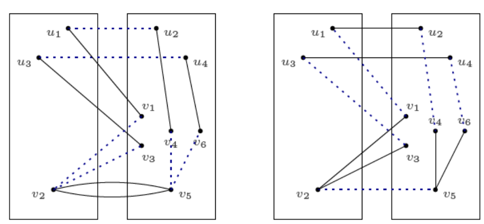
Type D_{12}^{13}



Type D_{12}^{14}

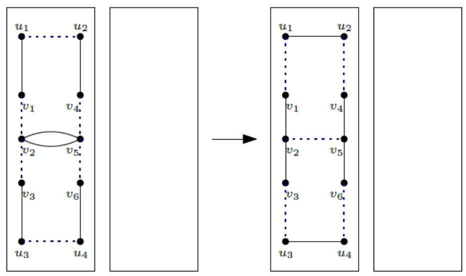


Type D_{12}^{15}

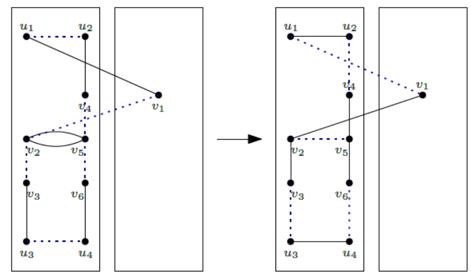


Type D_{12}^{16}

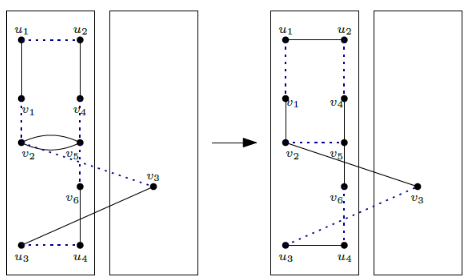
Figure 5.6: D_{12} switching 9-16



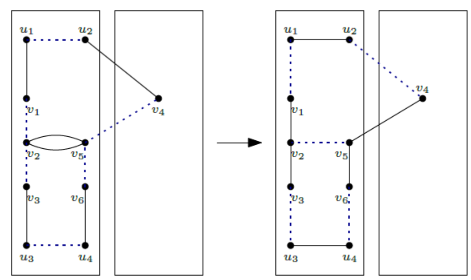
Type D_1^1



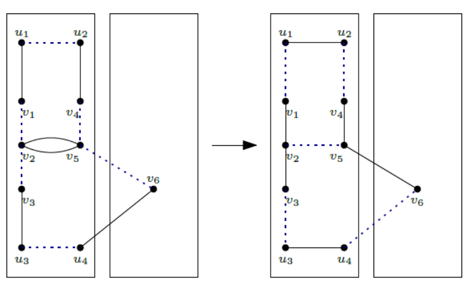
Type D_1^2



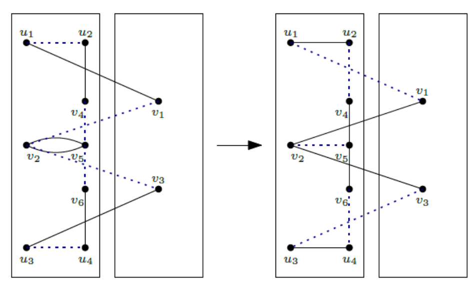
Type D_1^3



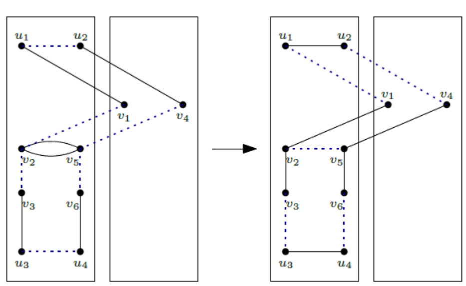
Type D_1^4



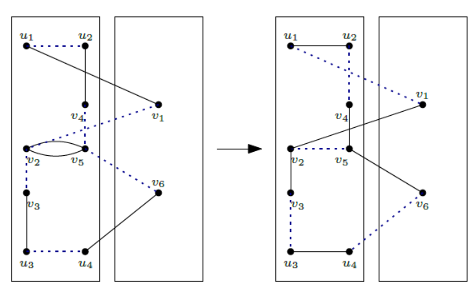
Type D_1^5



Type D_1^6



Type D_1^7



Type D_1^8

Figure 5.7: D_1 switching 1-8

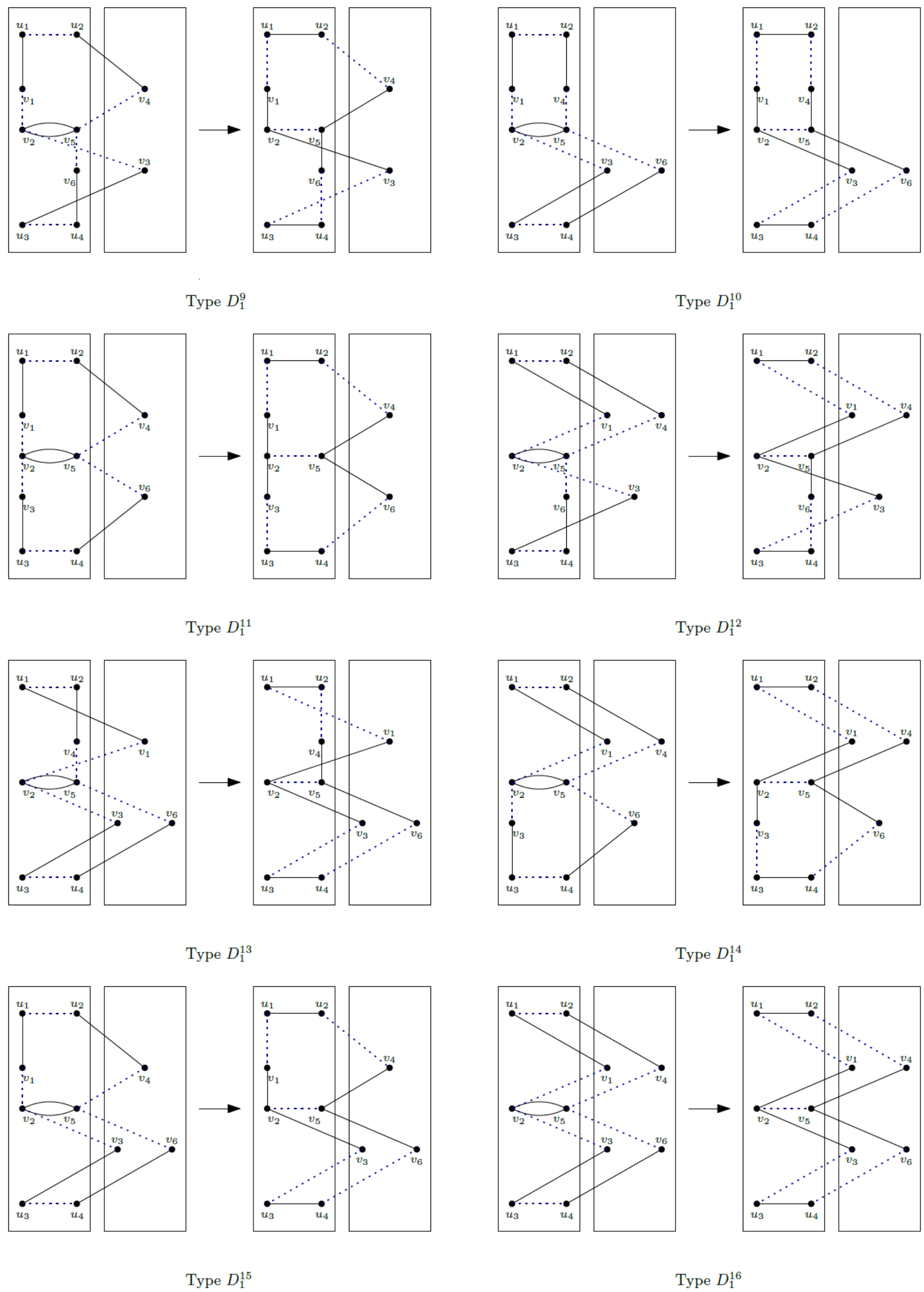


Figure 5.8: D_1 switching 9-16

5.3 JDM_GEN for $q = 2$

Recall again that G_0 is the multigraph output by PairingGen. We are ready to define JDM_GEN. The algorithm consists of three phases. In the first phase, all the crossing double edges are sequentially removed. Then, the double edges within each V_i, and the loops within each V_i, are sequentially removed in the second and the third phase respectively. We start with the discussions of the first phase. The general idea of the other two phases is similar.

5.3.1 Definition

Recall that for any $\boldsymbol{L} = (L_1, L_2) \in \mathbb{N}^2$ and $\boldsymbol{D} = (D_{11}, D_{22}, D_{12}) \in \mathbb{N}^3$, the set $\mathcal{G}_0(\boldsymbol{L}, \boldsymbol{D})$ is defined as follows. For any $G \in \mathcal{G}_0(\boldsymbol{L}, \boldsymbol{D})$, the following conditions hold:

(a) G has no loops of multiplicity greater than one or multiple edges of multiplicity greater than two.

(b) The number of simple loops in G in V_1 and V_2 is L_1 and L_2 respectively.

(c) The number of double edges in G in V_1 and V_2 is D_{11} and D_{22} respectively.

(d) The number of double edges in G between V_1 and V_2 is D_{12}.

Suppose we have $G_0 \in \mathcal{G}_0(\boldsymbol{L}, \boldsymbol{D})$ for some $\boldsymbol{L} = (L_1, L_2)$ and $\boldsymbol{D} = (D_{11}, D_{22}, D_{12})$. Recall the 16 types of switchings demonstrated in Figure 5.5 and Figure 5.6 for the removal of the crossing double edges. Note that these switchings do not alter the number of double edges or loops within V_1 or V_2. Thus, during the whole execusion of the first phase of JDM_GEN, $L_1(G_t), L_2(G_t), D_{11}(G_t)$ and $D_{22}(G_t)$ will not change. Hence, for convenience and without the danger of confusion, we suppress the lengthy notation $(\boldsymbol{L}, \boldsymbol{D})$, and instead use the notation $\mathcal{H}_m$ to represent the set $\mathcal{G}_0((L_1, L_2), (D_{11}, D_{22}, m))$ for $0 \le m \le D_{12}$ throughout the discussions of the first phase of JDM_GEN.

For each $\tau \in \{D_{12}^i : 1 \le i \le 16\}$, let $f_\tau(G)$ denote the number of switchings of type τ that can be applied to G. Recall that M is the joint degree matrix. The following are parameters set by JDM_GEN. Given $m \in \mathbb{N}$, set $\overline{f}^m_{D^i_{12}}$ as follows:

$$
\begin{aligned}
\overline{f}^{m}_{D^1_{12}} &= 4mM_{12}^2M_{11}^2; && (5.1a)\\
\overline{f}^{m}_{D^2_{12}} = \overline{f}^{m}_{D^3_{12}} &= 2mM_{12}^3M_{11}; && (5.1b)\\
\overline{f}^{m}_{D^4_{12}} = \overline{f}^{m}_{D^5_{12}} &= 8mM_{12}M_{11}^2M_{22}; && (5.1c)\\
\overline{f}^{m}_{D^6_{12}} &= mM_{12}^4; && (5.1d)\\
\overline{f}^{m}_{D^7_{12}} = \overline{f}^{m}_{D^8_{12}} = \overline{f}^{m}_{D^9_{12}} = \overline{f}^{m}_{D^{10}_{12}} &= 4mM_{12}^2M_{11}M_{22}; && (5.1e)\\
\overline{f}^{m}_{D^{11}_{12}} &= 16mM_{11}^2M_{22}^2; && (5.1f)\\
\overline{f}^{m}_{D^{12}_{12}} = \overline{f}^{m}_{D^{13}_{12}} &= 2mM_{12}^3M_{22}; && (5.1g)\\
\overline{f}^{m}_{D^{14}_{12}} = \overline{f}^{m}_{D^{15}_{12}} &= 8mM_{12}M_{11}M_{22}^2; && (5.1h)\\
\overline{f}^{m}_{D^{16}_{12}} &= 4mM_{12}^2M_{22}^2. && (5.1i)
\end{aligned}
$$

The parameters $\overline{f}^{m}_{\tau}$ here serve a similar role as $\overline{f}_d(m)$ in [3]. We prove that parameters $\overline{f}^{m}_{\tau}$ are upper bounds for $f_\tau(G)$ for $G \in \mathcal{H}_m$.

Lemma 5.1. *Given $m \in \mathbb{N}$, for any graph $G \in \mathcal{H}_m$ and for each $\tau \in \{D^i_{12} : 1 \le i \le 16\}$, $f_\tau(G) \le \overline{f}^{m}_{\tau}$.*

Proof. We will use the D^5_{12} switching as an example to demonstrate that $f_{D^5_{12}}(G) \le \overline{f}^{m}_{D^5_{12}} = 8mM_{12}M_{11}^2M_{22}$.

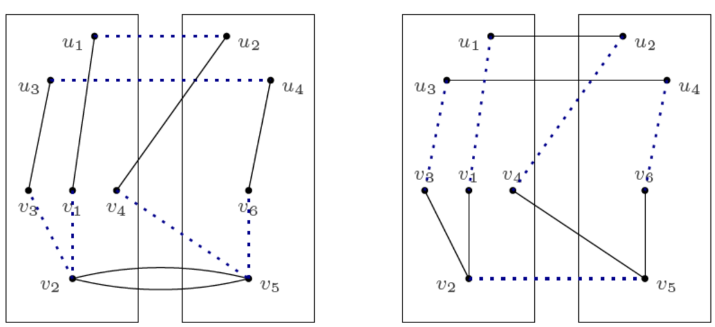


Figure 5.9: D^5_{12} switching

In the D^5_{12} switching, we have m choices for the double edge v_2v_5, at most M_{12} choices for the edge v_4u_2, at most $2M_{11}$ choices for each of the edges v_1u_1 and v_3u_3(order matters), and at most $2M_{22}$ choices for the edge v_6u_4. Therefore, without considering the forbidden cases such as vertex coincidences or the presence of non-edges, we have at most $\overline{f}^m_{D^5_{12}} = 8mM_{12}M^2_{11}M_{22}$ possible choices for $f_{D^5_{12}}(G)$. Hence, we conclude that $f_{D^5_{12}}(G) \leq \overline{f}^m_{D^5_{12}}$.

By following a similar reasoning, it can be verified that $f_\tau(G) \leq \overline{f}^m_\tau$ for all $\tau \in \{D^i_{12} : 1 \leq i \leq 16\}$. □

For each $\tau \in \{D^i_{12} : 1 \leq i \leq 16\}$, set

$$p^m_\tau = \frac{\overline{f}^m_\tau}{\sum_{i\in[16]} \overline{f}^m_{D^i_{12}}}.$$

This set of parameters will be used by JDM_GEN to probabilistically choose a particular type of switchings to perform in each step. Note that by definition, $\sum_{\tau\in\{D^i_{12}:1\leq i\leq 16\}} p^m_\tau = 1$, and thus $(p^m_\tau)_{\tau\in\{D^i_{12}:1\leq i\leq 16\}}$ defines a probability distribution over $\{D^i_{12} : 1 \leq i \leq 16\}$.

Now we define NoDoubles$_{12}$, which is a procedure called by JDM_GEN to eliminate all the crossing double edges. NoDoubles$_{12}$ calls a subprocedure RelaxGraph, which computes the probability of rejecting the switching S selected by NoDoubles$_{12}$, so that the resulting multigraph maintains uniformity after performing S. RelaxGraph is rather complicated and we will explain the details in the next section.

---

procedure NoDoubles12(G)
  while G has double edges between V_1, V_2 **do**
    Suppose $G \in \mathcal{H}_m$
    Choose switching type τ with probability p^m_τ, for $\tau \in \{D^1_{12}, \ldots, D^{16}_{12}\}$
    Choose a uniform random type τ switching S that can be applied to G
    f-rejection: reject with probability $1 - \frac{f_\tau(G)}{\overline{f}^m_\tau}$
    b-rejection: RelaxGraph(G, S)
    $G \leftarrow S(G)$
  end while
end procedure

5.3.2 Incremental Relaxation

Now we define RELAXGRAPH, which uses a technique called *incremental relaxation*, developed by Arman, Gao and Wormald [3]. Before introducing *incremental relaxation*, it is helpful to understand how b-rejection would work without incremental relaxation, and the advantages of using this technique. Recall NoDoubles$_{12}$. Suppose at step t we have G_t uniformly distributed in $\mathcal{H}_m$ for some m. By the definition of NoDoubles$_{12}$, the probability that a particular type τ switching S is selected, and is not f-rejected is

$$p_\tau^m \cdot \frac{1}{f_\tau(G)} \cdot \frac{f_\tau(G)}{\overline{f}_\tau^m} = \frac{\overline{f}_\tau^m}{\sum_{i\in[16]} \overline{f}_{D_{12}^i}^m} \cdot \frac{1}{\overline{f}_\tau^m} = \frac{1}{\sum_{i\in[16]} \overline{f}_{D_{12}^i}^m}. \tag{5.2}$$

Note that the right hand side above is constant for all (G, S) where $G \in \mathcal{H}_m$ and S is any switching that NoDoubles$_{12}$ may select. In particular, the quantity does not depend on τ, the type of the switching selected by NoDoubles$_{12}$. Let $b(G')$ denote the number of all possible switchings (of any of the 16 types) that can produce G' for $G' \in \mathcal{H}_{m-1}$. Given (G, S) in NoDoubles$_{12}$, let G' denote the multigraph that S transforms G into. Suppose we b-reject (G, S) with probability $C/b(G')$ for some constant C that does not depend on G'. Then, the probability that any graph $G' \in \mathcal{H}_{m-1}$ is created after a switching step and is not rejected (either f-rejected or b-rejected) is equal to

$$\sum_{(G,S):S \text{ produces } G'} \frac{1}{|\mathcal{H}_m|} \frac{1}{\sum_{i\in[16]} \overline{f}_{D_{12}^i}^m} \frac{C}{b(G')},$$

where $1/|\mathcal{H}_m|$ is the probability that the multigraph obtained before the switching step is G, $1/\sum_{i\in[16]} \overline{f}_{D_{12}^i}^m$ is the probability that S is selected and is not f-rejected, and $C/b(G')$ is the probability that (G, S) is not b-rejected. Note that the right hand side above is constant since C is constant and

$$|\{(G, S) : S \text{ produces } G'\}| = b(G'),$$

by definition of $b(G')$. Inductively, the resulting multigraph obtained after the switching step is uniform in $\mathcal{H}_{m-1}$.

This is essentially the idea of rejection schemes used in the literature [16] before the technique of incremental relaxation. The drawback is that the computation of $b(G')$ is time costly. With a brute-force search, it takes $O(\Delta^4 n^2)$ time to compute $b(G')$ by navigating through each pair of 2-paths in G' and checking if they can be created by a d-switching as in Figure 5.2. The time complexity can be reduced to $O(d^3 n)$ in the case of a d-regular degree sequence, by sophisticated counting schemes using inclusion and exclusion, and by using proper data structure. The time cost will be even higher in our case, as the structures created by a switching (recalling Figure 5.3) is more complicated than just a pair of 2-paths.

Incremental relaxation is a new rejection scheme that performs b-rejection without computing $b(G')$. We give an intuitive explanation of how it works in NoDoubles$_{12}$.

After the procedure chooses S, the resulting multigraph G' produced by S is determined. Instead of viewing G' as the output of S, we view the 5-tuple $(G', v_1v_2v_3, v_4v_5v_6, u_1u_2, u_3u_4)$ as the output of S, where each coordinate in the tuple except for G' signifies a certain structure in G' created by S (e.g. $v_1v_2v_3$ is a 2-path in G'); moreover, these structures satisfy certain constraints (e.g. all vertices involved are disjoint, u_1v_1 is a non-edge, etc.). By (5.2), every possible such 5-tuple is generated with equal probability. However, some G' produces more such 5-tuple than others. Instead of computing how many 5-tuples there are with the first coordinate being G', which is equivalent to computing $b(G')$, incremental relaxation computes $b(G', v_1v_2v_3)$, which is the number of simple 2-paths in G', and then $b(G', v_1v_2v_3, v_4v_5v_6)$, which is the number of choices for $v_4v_5v_6$, given a particular choice of $(G', v_1v_2v_3)$, etc. Finally, using these $b()$ values that are computed, incremental relaxation performs a sequence of b-rejections and sequentially produces $(G', v_1v_2v_3, v_4v_5v_6, u_1u_2)$, and then $(G', v_1v_2v_3, v_4v_5v_6)$, and then $(G', v_1v_2v_3)$ and finally G', each time uniformly distributed in their corresponding probability spaces. The essence of incremental relaxation is that computing the new $b()$ functions costs much less time than computing $b(G')$.

Here, we adopt the notation in [3]. For a more detailed explanation of incremental relaxation, readers are referred to Section 3 of [3]. In the following discussion, we will explain how incremental relaxation is applied to our problem. We always start with a general setting as in [3], followed by how it is applied in our particular problem.

Given a finite set $\mathcal{F}$ and a positive integer k, we are also provided with multisets S_i for $1 \leq i \leq k$, where each S_i consists of subsets of $\mathcal{F}$.

Let $\mathcal{F}_k$ be any subset of $\mathcal{F} \times S_1 \times \ldots \times S_k$ such that for $F = (G, C_1, \ldots, C_k) \in \mathcal{F}_k$, F satisfies $G \in C_k \subseteq C_{k-1} \subseteq \ldots \subseteq C_1$. For any $F = (G, C_1, \ldots, C_k) \in \mathcal{F}_k$, we define $P_i(F) = (G, C_1, \ldots, C_i)$ for $1 \leq i \leq k$.

For $1 \leq i \leq k-1$, we define $\mathcal{F}_i = \{P_i(F) : F \in \mathcal{F}_k\}$, and we set $\mathcal{F}_0 = \mathcal{F}$.

For $1 \leq i \leq k$, let $F = (G, C_1, \ldots, C_i) \in \mathcal{F}_i$. We define $P(F) = (G, C_1, \ldots C_{i-1})$ be the prefix of F.

For our problem, suppose RelaxGraph is called with parameters (S, G) such that $S(G) \in \mathcal{H}_m$. Then $\mathcal{F}$ will be $\mathcal{H}_m$ and $k = 4$. We define S_i in the following way:

Let $v_1, \ldots, v_6, u_1, \ldots, u_4$ be distinct vertices. We use $E_S(G)$ to denote the set of simple edges in graph G.

We define the following sets:

$$
\begin{aligned}
C_1^{(v_1,v_2,v_3)} &= \{G \in \mathcal{F} : v_1v_2, v_2v_3 \in E_S(G)\};\\
C_2^{(v_1,\ldots,v_6)} &= \{G \in C_1^{(v_1,v_2,v_3)} : v_4v_5, v_5v_6 \in E_S(G), v_2v_5 \notin E(G)\};\\
C_3^{(v_1,\ldots,v_6,u_1,u_2)} &= \{G \in C_1^{(v_1,\ldots,v_6)} : u_1u_2 \in E_S(G), v_1u_1, v_4u_2 \notin E(G)\};\\
C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)} &= \{G \in C_1^{(v_1,\ldots,v_6,u_1,u_2)} : u_3u_4 \in E_S(G), v_3u_3, v_6u_4 \notin E(G)\}.
\end{aligned}
$$

These sets can be considered as subsets of $\mathcal{F}$.

Now, we define the following sets:

$$
\begin{aligned}
&S_1 = \{C_1^{(v_1,v_2,v_3)} : v_1, v_2, v_3 \text{ are distinct}, v_2 \in V_1\};\\
&S_2 = \{C_2^{(v_1,\ldots,v_6)} : v_1, \ldots, v_6 \text{ are distinct}, v_2 \in V_1, v_5 \in V_2\};\\
&S_3 = \{C_3^{(v_1,\ldots,v_6,u_1,u_2)} : v_1, \ldots, v_6, u_1, u_2 \text{ are distinct}, v_2, u_1 \in V_1, v_5, u_2 \in V_2\};\\
&S_4 = \{C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)} : v_1, \ldots, v_6, u_1, \ldots, u_4 \text{ are distinct}, v_2, u_1, u_3 \in V_1, v_5, u_2, u_4 \in V_2\}.
\end{aligned}
$$

We let $\mathcal{F}_k = \mathcal{F}_4$ be

$$
\begin{aligned}
\mathcal{F}_4 = \{(G, &C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}, C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)}) : G \in C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)}\\
&v_1, \ldots, v_6, u_1, \ldots, u_4 \text{ are distinct and } v_2, u_1, u_3 \in V_1, v_5, u_2, u_4 \in V_2\}.
\end{aligned}
$$

Let $v_1, \ldots, v_6, u_1, \ldots, u_4$ be distinct vertices where $v_2, u_1, u_3 \in V_1, v_5, u_2, u_4 \in V_2$ such that

$$F = (G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}, C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)}) \in \mathcal{F}_4.$$

Then we have

$$
\begin{aligned}
&P_1(F) = (G, C_1^{(v_1,v_2,v_3)}) \in \mathcal{F}_1;\\
&P_2(F) = (G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}) \in \mathcal{F}_2;\\
&P_3(F) = (G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}) \in \mathcal{F}_3;
\end{aligned}
$$

$$
\begin{aligned}
&P(G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}) = (G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)});\\
&P(G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}) = (G, C_1^{(v_1,v_2,v_3)});\\
&P(G, C_1^{(v_1,v_2,v_3)}) = G.
\end{aligned}
$$

Get back to the general setting, for $0 \le i \le k-1$ and $F = (G, C_1, \ldots, C_i) \in \mathcal{F}_i$, we define $b(F)$ to be the number of $F' \in \mathcal{F}_{i+1}$ such that $P(F') = F$.

Suppose B_i $(0 \le i \le k-1)$ are numbers specified such that $b(F) \ge B_i$ for all $F \in \mathcal{F}_i$ and we assume that $B_i > 0$ for all $0 \le i \le k-1$.

For our problem, for $0 \le i \le 3$ and $F = (G, C_1, \ldots, C_i) \in \mathcal{F}_i$, $b(F)$ will be the number of $F' \in \mathcal{F}_{i+1}$ such that $P(F') = F$. For example, let $F = (G, C_1^{(v_1,v_2,v_3)}) \in \mathcal{F}_1$, then $b(F)$ is the number of $F' \in \mathcal{F}_2$ such that $P(F') = (G, C_1^{(v_1,v_2,v_3})$. In other words, $b(F)$ is the number of three tuples (v_4, v_5, v_6) such that $(G, C_1^{(v_1,v_2,v_3}, C_2^{(v_1,\ldots,v_6)}) \in \mathcal{F}_2$, we will have a more detailed discussion on $b(F)$ in Section 5.3.4 to give a more specific combinatorial interpretation of $b(F)$.

Recall that Δ is the maximum degree, that is $\Delta = max\{d_1, d_2\}$. We specify parameters below that correspond to B_i mentioned above in the general setup:

$$B_{12}^{(0)} = n_1 d_1(d_1 - 1) - \frac{32 M_{12}^2 d_1}{n_1 n_2} - \frac{64 M_{11}^2 d_1}{n_1^2} - \frac{16 M_{11} d_1}{n_1}; \tag{5.3a}$$

$$B_{12}^{(1)} = n_2 d_2(d_2 - 1) - \frac{32 M_{12}^2 d_2}{n_1 n_2} - \frac{64 M_{22}^2 d_2}{n_2^2} - \frac{16 M_{22} d_2}{n_2} - 8 d_2 \Delta - d_2^2 \Delta; \tag{5.3b}$$

$$B_{12}^{(2)} = M_{12} - \frac{16 M_{12}^2}{n_1 n_2} - 10\Delta - 2\Delta^2; \tag{5.3c}$$

$$B_{12}^{(3)} = M_{12} - \frac{16 M_{12}^2}{n_1 n_2} - 12\Delta - 2\Delta^2. \tag{5.3d}$$

The parameters B_{12} here serve a similar role as $\underline{b}_d$ in [3]. The following lemma verifies that the above $B_{12}^{(i)}$s are indeed lower bounds for $b(F)$ for $F \in \mathcal{F}_i$. Its proof is deferred to Section 5.4.2.

Lemma 5.2. *Provided that*

$$\frac{M_{ij}^3 \Delta^2}{n_i^2 n_j^2} = o(1) \text{ for all } 1 \le i \le j \le 2,$$

$b(F) \ge B_{12}^{(i)} > 0$ *for all* $F \in \mathcal{F}_i$ *for* $0 \le i \le 3$.

Incremental consists of two main procedures: Loosen and Relax.

The LOOSEN procedure takes an input $F \in \mathcal{F}_i$. It may either reject with a certain probability or output the prefix $P(F)$ of F.

The RELAX procedure takes an element F from $\mathcal{F}_k$. It repeatedly calls the LOOSEN procedure until the value returned is an element from $\mathcal{F}_0$.

```
procedure LOOSEN(F)
    Suppose F ∈ F_i
    Reject with probability 1 − B_{i−1}/b(P(F))
    Return P(F)
end procedure
```

```
procedure RELAX(F)
    Suppose F ∈ F_k
    i := k
    while i ≥ 1 do
        F = LOOSEN(F)
        i = i − 1
    end while
end procedure
```

The following three lemmas(5.3, 5.4 and 5.5) are proved in [3], which states that if F is chosen uniformly from $\mathcal{F}_k$, then the output of the RELAX procedure, denoted as RELAX(F), is uniform in $\mathcal{F}$ if no rejection occurs.

Lemma 5.3. *[3, **Lemma 5**] Assume that $i \in [k]$ and $B_i > 0$. Provided that $F \in \mathcal{F}_i$ is chosen uniformly at random, the output of* LOOSEN(F) *is uniformly in $\mathcal{F}_{i-1}$ assuming no rejection occurs.*

Lemma 5.4. *[3, **Corollary 7**] When applied* RELAX *to $(G, C_1, \ldots, C_k) \in \mathcal{F}_k$, the algorithm outputs G with probability $\prod_{i=0}^{k-1} \frac{B_i}{b(G,C_1,\ldots,C_i)}$, and ends in rejection otherwise.*

Lemma 5.5. *[3, **Corollary 6**] Assume that for all $i \in [k]$, $B_{i-1} > 0$, and assume $F \in \mathcal{F}_k$ is chosen uniformly at random. Then the output of* RELAX(F) *is uniform in $\mathcal{F}$ if there is no rejection.*

5.3.3 Uniformity of NoDoubles$_{12}$

In this section we show that NoDoubles$_{12}$ preserves the uniformity under the assumption of $B_{12}^{(i)} > 0$ for $0 \le i \le 3$. The idea is to apply Relax to our problem. First, we define the procedure RelaxGraph which is called within NoDoubles$_{12}$. Suppose RelaxGraph(G, S) is called in NoDoubles$_{12}$ for some G and S where S involves an ordered set of ten vertices $(v_1, v_2, v_3, v_4, v_5, v_6, u_1, u_2, u_3, u_4)$. We define the following notation for simplicity:

(a) $V_1(S) = (v_1, v_2, v_3)$;

(b) $V_2(S) = (v_1, v_2, v_3, v_4, v_5, v_6)$;

(c) $V_3(S) = (v_1, v_2, v_3, v_4, v_5, v_6, u_1, u_2)$;

(d) $V_4(S) = (v_1, v_2, v_3, v_4, v_5, v_6, u_1, u_2, u_3, u_4)$.

Then, the procedure RelaxGraph is defined as follows:

```
procedure RelaxGraph(G, S)
    Let F = (S(G), C_1^{V_1(S)}, C_2^{V_2(S)}, C_3^{V_3(S)}, C_4^{V_4(S)})
    Relax(F)
end procedure
```

By employing a similar proof strategy as presented in [3, **Corollary 15**], we derive the following lemma.

Lemma 5.6. *Provided that $B_{12}^{(i)} > 0$ for $0 \le i \le 3$, if no rejection occurs during the execution of* NoDoubles$_{12}(G_0)$*, then the resulting G_0 is uniformly distributed in $\mathcal{H}_0$.*

Proof. We prove it by induction. Let $G'_0 = G_0$ and G'_i be the graph obtained after the i-th iteration for $1 \le i \le m$ where m is the number of crossing double edges in G_0. If no rejection occurs in the first i iterations, we have $G'_i \in \mathcal{H}_{m-i}$. We will now demonstrate that G'_i is uniformly distributed in $\mathcal{H}_{m-i}$.

When $i = 0$, we have $G'_0 = G_0$ chosen uniformly at random from $\mathcal{H}_m$ according to Lemma 4.3. Suppose no rejection occurs in the first k iterations and G'_i is uniformly distributed in $\mathcal{H}_{m-i}$ for all $i \le k$. Now, suppose no rejection occurs in the $(k+1)$-th iteration. We will prove that G'_{k+1} is uniformly distributed in $\mathcal{H}_{m-(k+1)}$.

Since G'_k is uniformly distributed in $\mathcal{H}_{m-k}$, the probability that $G'_k = G$ is the same for all $G \in \mathcal{H}_{m-k}$ where we denote this probability as δ_k. Now, consider $G \in \mathcal{H}_{m-k}$ and a type τ switching S for some τ. The probability that (G, S) is called by RelaxGraph can be expressed as:

$$\delta_k p_\tau^{(m-k)} \frac{1}{f_\tau(G')} \frac{f_\tau(G')}{\overline{f}_\tau^{(m-k)}} = \delta_k \frac{p_\tau^{(m-k)}}{\overline{f}_\tau^{(m-k)}},$$

where this probability only depends on m and k.

Hence for each possible (G, S) pair, the corresponding term $F = (S(G), C_4^{V_4(S)}, C_3^{V_3(S)}, C_2^{V_2(S)}, C_1^{V_1(S)}) \in \mathcal{F}_4$ is called by Relax with the same probability.

The remaining task is to show that for every $F \in \mathcal{F}_4$, there exists $G \in \mathcal{H}_{m-k}$ and S such that the corresponding term of (G, S) is precise F. By establishing this, we can conclude that each $F \in \mathcal{F}_4$ is called by Relax with the same probability. This allows us to apply Lemma 5.5.

Suppose $F = (G', C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}, C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)}) \in \mathcal{F}_4$ for some graph $G' \in \mathcal{F} = \mathcal{H}_{m-k+1}$ and vertices $v_1, \ldots, v_6, u_1, \ldots, u_4$. We can get a graph $G \in \mathcal{H}_{m-k}$ by performing a backward switching on G' with the vertices $v_1, \ldots, v_6, u_1, \ldots, u_4$, which involves the following steps:

(a) Remove the edges $v_1v_2, v_2v_3, v_4v_5, v_5v_6, u_1u_2, u_3u_4$ from G'.

(b) Add the simple edges $v_1u_1, v_3u_3, v_4u_2, v_6u_4$ to G'.

(c) Add the double edge v_2v_5 to G'.

Let S be the switching such that $S(G) = G'$. It can be observed that the corresponding term of (G, S) is F.

Hence, each $F \in \mathcal{F}_4$ is called by Relax with the same probability. Thus, if no rejection occurs during the $(k+1)$-th iteration, G'_{k+1} is uniformly distributed in $\mathcal{H}_{m-(k+1)}$ by Lemma 5.5.

By induction, if no rejection occurs in the first m iterations, we can conclude that the final graph $G = G'_m$ is uniformly distributed in $\mathcal{H}_{m-m} = \mathcal{H}_0$. □

5.3.4 Combinatorial interpretation of $b(F)$

Recall that in the b-rejection scheme, we need to compute the value of $b(P(F))$ when LOOSEN is called. In this section, we will give a combinatorial description of $b(P(F))$ in our problem by interpreting $\mathcal{F}_i$ from a different perspective. This will be used to verify that parameters in (5.3) are lower bounds for the $b(F)$s, yielding a proof for Lemma 5.2. It will also be used to design a counting scheme for efficient computation of $b(F)$. These will be further discussed in Section 5.4.1.

Suppose RELAXGRAPH is called with parameters (G, S), where $G' = S(G)$. In RELAXGRAPH, RELAX is called with parameters $(G', C_4^{V_4(S)}, C_3^{V_3(S)}, C_2^{V_2(S)}, C_1^{V_1(S)})$. If no rejection occurs in RELAX, the following values need to be computed when LOOSEN is processed:

$$b(P(G', C_1^{V_1(S)}, C_2^{V_2(S)}, C_3^{V_3(S)}, C_4^{V_4(S)})) = b(G', C_1^{V_1(S)}, C_2^{V_2(S)}, C_3^{V_3(S)}); \tag{5.4a}$$

$$b(P(G', C_1^{V_1(S)}, C_2^{V_2(S)}, C_3^{V_3(S)})) = b(G', C_1^{V_1(S)}, C_2^{V_2(S)}); \tag{5.4b}$$

$$b(P(G', C_1^{V_1(S)}, C_2^{V_2(S)})) = b(G', C_1^{V_1(S)}); \tag{5.4c}$$

$$b(P(G', C_1^{V_1(S)})) = b(G'). \tag{5.4d}$$

Suppose $F \in \mathcal{F}_i$ for some i. Recall that $b(F)$ represents the number of $F' \in \mathcal{F}_{i+1}$ such that $P(F') = F$, and $\mathcal{F}_i = \{P_i(F) : F \in \mathcal{F}_4\}$ for $1 \le i \le 3$. Currently, we lack an efficient method to compute the value of $b(F)$. Hence, we introduce a different perspective on $\mathcal{F}_i$, which brings a combinatorial meaning of $b(F)$ and hence leads to an efficient approach for computing $b(F)$.

Define

$$\begin{aligned}
\mathcal{F}'_1 &= \{(G, C_1^{(v_1,v_2,v_3)}) : G \in C_1^{(v_1,v_2,v_3)}, v_1, \ldots, v_3 \text{ be distinct vertices where } v_2 \in V_1\};\\
\mathcal{F}'_2 &= \{(G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}) : G \in C_2^{(v_1,\ldots,v_6)}\\
&\qquad v_1, \ldots, v_6 \text{ be distinct vertices where } v_2 \in V_1, v_5 \in V_2\};\\
\mathcal{F}'_3 &= \{(G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}) : G \in C_3^{(v_1,\ldots,v_6,u_1,u_2)}\\
&\qquad v_1, \ldots, v_6, u_1, u_2 \text{ be distinct vertices where } v_2, u_1 \in V_1, v_5, u_2 \in V_2\}.
\end{aligned}$$

Claim 5.7. Provided that $B_{12}^{(i)} > 0$ for $0 \le i \le 3$, $\mathcal{F}_i = \mathcal{F}'_i$ for $1 \le i \le 3$.

Proof. The proof can be seen as an inductive argument, starting with the case for $i = 3$ and using the results of $\mathcal{F}_3 = \mathcal{F}'_3$ to establish the proof for $i = 2$. By extending this approach, we can then prove the cases for $i = 1$ in a similar manner. We start by showing that $\mathcal{F}_3 = \mathcal{F}_3'$.

Suppose $F \in \mathcal{F}_3$. Then, there exists $F^* \in \mathcal{F}_4$ such that $F = P_3(F^*)$. Let us assume that

$$F^* = (G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}, C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)})$$

for some $v_1, \ldots, v_6, u_1, \ldots, u_4$.

By definition of $\mathcal{F}_4$, we have that $v_1, \ldots, v_6, u_1, u_2$ are distinct, $v_2, u_1 \in V_1$, $v_5, u_2 \in V_2$, and $G \in C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)} \subseteq C_3^{(v_1,\ldots,v_6,u_1,u_2)}$. Therefore, we can conclude that

$$F = P_3(F^*) = (G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}) \in \mathcal{F}_3'.$$

Next, let's suppose $F \in \mathcal{F}_3'$ where we assume that

$$F = (G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)})$$

for some distinct vertices $v_1, \ldots, v_6, u_1, u_2$.

To show that $F \in \mathcal{F}_3$, we need to find (u_3, u_4) such that $G \in C_4^{(v_1,\ldots,v_6,u_1,u_2,u_3,u_4)}$ and $C_4^{(v_1,\ldots,v_6,u_1,u_2,u_3,u_4)} \in S_4$. Then we have

$$(G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}, C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)}) \in \mathcal{F}_4,$$

and hence we have

$$P_3(G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}, C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)}) = F,$$

which shows that $F \in \mathcal{F}_3$.

It is equivalent to find (u_3, u_4) that satisfy the following conditions:

(a) u_3 and u_4 are distinct vertices, different from $v_1, \ldots, v_6, u_1, u_2$,

(b) $u_3 \in V_1$ and $u_4 \in V_2$,

(c) $u_3u_4 \in E_S(G)$, and $v_3u_3, v_6u_4 \notin E(G)$.

Conditions $(a), (b)$ ensure that $C_4^{(v_1,\ldots,v_6,u_1,u_2,u_3,u_4)} \in S_4$, and condition (c) ensures that $G \in C_4^{(v_1,\ldots,v_6,u_1,u_2,u_3,u_4)}$.

The existence of such (u_3, u_4) will be demonstrated later in Section 5.4.2. This completes the proof that $\mathcal{F}_3 = \mathcal{F}'_3$.

Next, let's consider the case for $i = 2$. The direction of showing $\mathcal{F}_2 \subseteq \mathcal{F}'_2$ is almost the same as the case for $i = 3$ and can be considered trivial.

So let's show the direction of $\mathcal{F}'_2 \subseteq \mathcal{F}_2$. Suppose $F \in \mathcal{F}'_2$ such that

$$F = (G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)})$$

for some vertices $v_1, \ldots, v_6$. Similarly, if we want to show that $F \in \mathcal{F}_2$, we need to find (u_1, u_2, u_3, u_4) such that $G \in C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)} \subseteq C_3^{(v_1,\ldots,v_6,u_1,u_2)}$, $C_4^{(v_1,\ldots,v_6,u_1,\ldots,u_4)} \in S_4$ and $C_3^{(v_1,\ldots,v_6,u_1,u_2)} \in S_3$.

Instead of finding the four-vertex tuple, we first try to find (u_1, u_2) such that $G \in C_3^{(v_1,\ldots,v_6,u_1,u_2)}$ and $C_3^{(v_1,\ldots,v_6,u_1,u_2)} \in S_3$. This is equivalent to finding (u_1, u_2) that satisfy the following conditions:

(a) u_1 and u_2 are distinct vertices, different from $v_1, \ldots, v_6$.

(b) $u_1 \in V_1$ and $u_2 \in V_2$.

(c) $u_1u_2 \in E_S(G)$, and $v_1u_1, v_4u_2 \notin E(G)$.

Similarly, the existence of such (u_1, u_2) will be demonstrated later in Section 5.4.2.

Once we have such (u_1, u_2), we have

$$(G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}) \in \mathcal{F}'_3 = \mathcal{F}_3.$$

Hence, there exists $F^* \in \mathcal{F}_4$ such that

$$P_3(F^*) = (G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}, C_3^{(v_1,\ldots,v_6,u_1,u_2)}).$$

Thus, we have

$$P_2(F^*) = (G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\ldots,v_6)}) = F,$$

which shows $F \in \mathcal{F}_2$, and it completes the proof of $\mathcal{F}_2 = \mathcal{F}'_2$.

For the case $i = 1$, the approach remains the same. Following a similar idea, we need to solve the following problem when proving for $\mathcal{F}'_1 \subseteq \mathcal{F}_1$:

We are given vertices (v_1, v_2, v_3) and we need to find (v_4, v_5, v_6) that satisfy the following conditions:

(a) v_4, v_5, v_6 are distinct vertices, different from v_1, v_2, v_3.

(b) $v_5 \in V_2$.

(c) $v_4v_5, v_5v_6 \in E_S(G), v_2v_5 \notin E(G)$.

Similarly, the existence of (v_4, v_5, v_6) will be demonstrated in Section 5.4.2. $\square$

Now with the new perspective on $\mathcal{F}_i$, let's explore the combinatorial meaning of the value (5.4) we need to compute. For example, one of the values we need to compute in LOOSEN is $b(G', C_1^{V_1(S)})$, where $b(G', C_1^{V_1(S)})$ represents the number of $F' \in \mathcal{F}_2$ such that $P(F') = (G', C_1^{V_1(S)})$. Suppose $V_1(S) = (v_1, v_2, v_3)$, given that

$$\begin{aligned}\mathcal{F}_2 = \mathcal{F}_2' = \{(G, C_1^{(v_1,v_2,v_3)}, C_2^{(v_1,\dots,v_6)}) : G \in C_2^{(v_1,\dots,v_6)} \\ \text{and } v_1, \dots, v_6 \text{ are distinct vertices, where } v_2 \in V_1 \text{ and } v_5 \in V_2\},\end{aligned}$$

we can express $b(G', C_1^{V_1(S)})$ as the number of tuples (w_1, w_2, w_3) such that w_1, w_2, w_3 are distinct vertices and different from v_1, v_2, v_3, and $w_2 \in V_2, w_1w_2, w_2w_3 \in E_S(G'), w_2v_2 \notin E(G')$.

Next, following a similar approach, we can determine the meaning of $b(F)$ for each term in (5.4). To do so, we introduce the following definitions:

Let $G' \in \mathcal{H}_m$ for some $0 \le m \le \overline{D}_{12}$. Let S be a switching that is used to remove double edges with one endpoint in V_1 and the other endpoint in V_2 and S produce G'. Suppose S involves an ordered set of ten vertices $(v_1, v_2, v_3, v_4, v_5, v_6, u_1, u_2, u_3, u_4)$. We define $b_{12}^{(i)}(G', S)$ for $1 \le i \le 4$ as follows:

(a) Let $b_{12}^{(1)}(G', S)$ be the number of tuples (w_1, w_2, w_3) in G' such that w_1, w_2, w_3 are distinct vertices, and $w_2 \in V_1, w_1w_2, w_2w_3 \in E_S(G')$.

(b) Let $b_{12}^{(2)}(G', S)$ be the number of tuples (w_1, w_2, w_3) in G' such that w_1, w_2, w_3 are distinct vertices and different from v_1, v_2, v_3, and $w_2 \in V_2, w_1w_2, w_2w_3 \in E_S(G'), w_2v_2 \notin E(G')$.

(c) Let $b_{12}^{(3)}(G', S)$ be the number of tuples (w_1, w_2) in G' such that w_1, w_2 are distinct vertices and different from $v_1, \dots, v_6$, and $w_1 \in V_1$, $w_2 \in V_2, w_1w_2 \in E_S(G'), w_1v_1, w_2v_4 \notin E(G')$.

(d) Let $b_{12}^{(4)}(G', S)$ be the number of tuples (w_1, w_2) in G' such that w_1, w_2 are distinct vertices and different from $v_1, \ldots, v_6, u_1, u_2$, and $w_1 \in V_1$, $w_2 \in V_2, w_1w_2 \in E_S(G'), w_1v_3, w_2v_6 \notin E(G')$

Claim 5.8. For every $0 \leq m \leq \overline{D}_{12}$ and every $G' \in \mathcal{H}_m$. Let S be a switching produce G', then we have

$$\begin{aligned}
&b(G', C_1^{V_1(S)}, C_2^{V_2(S)}, C_3^{V_3(S)}) = b_{12}^{(4)}(G', S);\\
&b(G', C_1^{V_1(S)}, C_2^{V_2(S)}) = b_{12}^{(3)}(G', S);\\
&b(G', C_1^{V_1(S)}) = b_{12}^{(2)}(G', S);\\
&b(G') = b_{12}^{(1)}(G', S).
\end{aligned}$$

Proof. We have already shown that $b(G', C_1^{V_1(S)}) = b_{12}^{(2)}(G', S)$ in the above discussion. The other three equations can be shown in a similar way. □

Corollary 5.9. *For every $0 \leq m \leq \overline{D}_{12}$ and every $G \in \mathcal{H}_m$. Let S be a switching that can be applied on G where $S(G) = G'$, then* RelaxGraph(G, S) *rejects with probability*

$$1 - \frac{B_{12}^{(0)}}{b_{12}^{(1)}(G', S)} \frac{B_{12}^{(1)}}{b_{12}^{(2)}(G', S)} \frac{B_{12}^{(2)}}{b_{12}^{(3)}(G', S)} \frac{B_{12}^{(3)}}{b_{12}^{(4)}(G', S)}.$$

Proof. This result follows directly from the definition of Loosen and Claim 5.8. □

5.3.5 Definiton of JDM_GEN for $q = 2$

We are ready to define JDM_GEN for $q = 2$. Recall that NoDoubles$_{12}$ is a procedure that repeatedly removes crossing double edges joining vertices in V_1 and vertices in V_2. The procedures NoDoubles$_1$, NoDoubles$_2$, NoLoops$_1$, and NoLoops$_2$ will be defined in Chapter 6 to remove the other types of double edges and loops. Our algorithm JDM_GEN simply calls these procedures in sequence and returns the graph obtained after the last iteration of NoLoops$_2$.

Additionally, we include a step for parameter validation. This step is processed after generating the initial graph using PairingGen. Its purpose is to ensure that JDM_GEN can handle all possible inputs $(\boldsymbol{V}, M, \boldsymbol{d})$. Specifically, if any of the parameters used in RelaxGraph are not greater than zero, we reject the initial graph if it's not a simple graph. The parameters for NoDoubles$_1$, NoDoubles$_2$, NoLoops$_1$, and NoLoops$_2$ will be declared in Section 6.3, which can be found in (6.4a)–(6.4d) and (6.6a)–(6.6b).

Algorithm JDM_GEN($\boldsymbol{V}, M, \boldsymbol{d}$) for $q = 2$

```
(P, G) = PairingGen(V, M, d)
if not (B_ij^(k) > 0 for 0 ≤ k ≤ 3 for all 1 ≤ i ≤ j ≤ 2 and
        B_i^(k) > 0 for 0 ≤ k ≤ 1 for i ∈ [2]) then
    Reject if G is not a simple graph
end if
NoDoubles_12(G)
NoDoubles_1(G)
NoDoubles_2(G)
NoLoops_1(G)
NoLoops_2(G)
Return G
```

5.4 Running time and rejection probabilities

In this section, we analyze the overall running time of JDM_GEN. As part of the analysis, we bound the overall probability that any rejection occurs.

5.4.1 Running time of NoDoubles$_{12}$

In this section, we first analyze the running time of NoDoubles$_{12}$. We then provide proof of Lemma 5.2 and complete the proof for Claim 5.7. Finally, we bound the probability that any rejection occurs during the execution of NoDoubles$_{12}$.

Lemma 5.10. *If no rejection occurs during the execution of* NoDoubles$_{12}$*, then the running time of* NoDoubles$_{12}$ *is* $O(M_{12}^2 \Delta^2 / n_1 n_2)$.

Proof. Recall the procedure NoDoubles$_{12}$ defined in Section 5.3.1.

The time cost of each step of NoDoubles$_{12}$ comes from computing the values of $f_\tau(G)$, $b_{12}^{(1)}(G', S)$, $b_{12}^{(2)}(G', S)$,$b_{12}^{(3)}(G', S)$, and $b_{12}^{(4)}(G', S)$.

Computation of $f_\tau(G)$ We show that we can perform f-rejections without computing $f_\tau(G)$. Consider the case of type D_{12}^5 as an example, and all the other cases of types are similar. Assume that G has m crossing double edges. Instead of choosing a uniform random type $\tau = D_{12}^5$ switching that can be applied to G and then computing $f_{D_{12}^5}(G)$, we randomly select a double edge v_2v_5 between V_1 and V_2, a simple edge v_3v_6 between V_1 and V_2, two simple edges within V_1 (namely, v_1v_4 and u_1u_2), and a single simple edge u_3u_4 within V_2. Then f-reject if the choice does not produce a valid type τ-switching, i.e. one that can be applied to G. The total number of choices is exactly $\overline{f}^m_{D_{12}^5}$, recalling that $\overline{f}^m_{D_{12}^5} = 8mM_{12}M_{11}^2M_{22}$. The total number of valid choices is $f_{D_{12}^5}(G)$. Thus, the rejection probability is exactly $1 - f_{D_{12}^5}(G)/\overline{f}^m_{D_{12}^5}$, as desired.

Computation of $b_{12}^{(i)}(G', S)$ We will prove the following claim.

Claim 5.11. For each $1 \le i \le 4$, $b_{12}^{(i)}(G', S)$ can be computed in $O(\Delta^2)$ time.

Since G_0 has at most $\overline{D}_{12} = O(M_{12}^2/n_1n_2)$ crossing double edges by (3.1), NoDoubles$_{12}$ lasts $O(M_{12}^2/n_1n_2)$ steps. Multiplying it by $O(\Delta^2)$ gives the desired bound for the running time of NoDoubles$_{12}$ by Claim 5.11. □

Proof of Claim 5.11. To compute $b_{12}^{(i)}(G', S)$ for $1 \le i \le 4$, we first define some parameters. For each vertex v in G, for $j \in \{1, 2\}$, define $k_v^{(j)}$ to be the number of neighbours of v in V_j such that the edge between these neighbours and v is a simple edge. Additionally, let $k_v = k_v^{(1)} + k_v^{(2)}$ for each vertex v. The values of $k_v^{(1)}, k_v^{(2)}$, and k_v for all vertices can be determined in $O(n_1d_1 + n_2d_2)$ time once the initial graph G_0 is generated. This step is prepared before we run NoDoubles$_{12}$, so the running time is not included here. We update the value of $k_v^{(1)}, k_v^{(2)}$, and k_v after each switching is applied during the algorithm. Since each switching involves at most 10 vertices, and we only need to update $k_v^{(1)}, k_v^{(2)}$,and k_v for these 10 vertices, it takes $O(1)$ time for updating after each switching is applied.

Computation of $b_{12}^{(1)}(G', S)$ By Claim 5.8, $b_{12}^{(1)}(G', S) = \sum_{v\in V_1} k_v(k_v - 1)$. We can store the value for $\sum_{v\in V_1} k_v(k_v - 1)$ simultaneously when computing the value for $k_v^{(1)}, k_v^{(2)}$,and k_v during the preparation. And the update for $\sum_{v\in V_1} k_v(k_v - 1)$ can be done in $O(1)$ time after each switching is applied. Hence it takes $O(1)$ time to compute $b_{12}^{(1)}(G', S)$ in each iteration.

Computation of $b_{12}^{(2)}(G', S)$ By Claim 5.8, $b_{12}^{(1)}(G', S) \le \sum_{v\in V_2} k_v(k_v - 1)$ since there are forbidden cases that need to be considered. The value of $\sum_{v\in V_2} k_v(k_v - 1)$ can be computed and updated similar to $\sum_{v\in V_1} k_v(k_v - 1)$. Hence, the value of $b_{12}^{(2)}(G', S)$ can be computed by subtracting the number of forbidden cases from $\sum_{v\in V_2} k_v(k_v - 1)$ where the number of forbidden cases can be computed as follows:

Suppose $V_1(S) = (v_1, v_2, v_3)$, set $p = 0$ to store the data for the number of forbidden cases.

(a) For each neighbour v of v_2 in G', if $v \in V_2$, then we add $k_v(k_v - 1)$ to p.

(b) For each neighbour v of v_1 in G', if $v \in V_2$ then:

  (i) If v is adjacent to v_2, this situation has been accounted for in case (a).

  (ii) If v is not adjacent to v_2 but is adjacent to v_3, then we add $4k_v - 6$ to p.

  (iii) If v is not adjacent to both v_2 and v_3, then we add $2k_v - 2$ to p.

(c) For each neighbour v of v_3 in G', if $v \in V_2$ then:

  (i) If v is adjacent to v_2 or v_1, these situations have been accounted for in cases (a) and (b).

  (ii) If v is not adjacent to both v_2 and v_1, then we add $2k_v - 2$ to p.

Once we finish the above process, we have $b_{12}^{(2)}(G', S) = \sum_{v \in V_2} k_v(k_v - 1) - p$. It takes $O(\Delta)$ time to go through the above process and thus, to compute $b_{12}^{(2)}(G', S)$.

Computation of $b_{12}^{(3)}(G', S)$ By Claim 5.8, $b_{12}^{(3)}(G', S) \leq M_{12}$. Similarly, we need to compute the number of forbidden cases, and the process is as follows:

Suppose $V_2(S) = (v_1, v_2, v_3, v_4, v_5, v_6)$ and there are m crossing double edges in G'. Set $p = 0$ to store the data for the number of forbidden cases.

(a) Add $2m$ to p.

(b) Add $k_{v_2}^{(2)} + k_{v_5}^{(1)}$ to p.

(c) For $i \in \{1, 3, 4, 6\}$, if $v_i \in V_1$, add $k_{v_i}^{(2)}$ to p; if $v_i \in V_2$, add $k_{v_i}^{(1)}$ to p.

(d) For each edge with both endpoints in $\{v_1, \ldots, v_6\}$ and being a crossing edge in G', subtract 1 from p.

(e) For each neighbour v of v_1, if $v \in V_1$, and v does not coincide with $\{v_1, \ldots, v_6\}$ then:

  (i) For each neighbour u of v, if u does not coincide with $\{v_1, \ldots, v_6\}$, then add 1 to p.

(f) For each neighbour v of v_4, if $v \in V_2$, and v does not coincide with $\{v_1, \ldots, v_6\}$ then:

   (i) For each neighbour u of v, if u does not coincide with $\{v_1, \ldots, v_6\}$ and $v_1 u \notin E(G')$, then add 1 to p.

Case (a) includes forbidden cases of double edges, cases $(b), (c), (d)$ include the forbidden cases of vertex coincidence, and cases $(e), (f)$ include forbidden cases of presence non-edges.

Once we finish the above process, we have $b_{12}^{(3)}(G', S) = M_{12} - p$. It takes $O(\Delta^2)$ time to go through the above process and thus, to compute $b_{12}^{(3)}(G', S)$.

Computation of $b_{12}^{(4)}(G', S)$ The computation for $b_{12}^{(4)}(G', S)$ is very similar to $b_{12}^{(3)}(G', S)$ and also takes $O(\Delta^2)$ time.

Hence, for each $1 \le i \le 4, b_{12}^{(i)}(G', S)$ can be computed in $O(\Delta^2)$ time. □

5.4.2 Proof of Lemma 5.2 and Completion of the Proof for Claim 5.7.

Next, we proceed with the proof of Lemma 5.2 and fulfill the proof for Claim 5.7. Initially, we establish the following claim, which is essential to our proof of Claim 5.7.

Recall that we have

$$
\begin{aligned}
B_{12}^{(0)} &= n_1 d_1(d_1 - 1) - \frac{32M_{12}^2 d_1}{n_1 n_2} - \frac{64M_{11}^2 d_1}{n_1^2} - \frac{16M_{11} d_1}{n_1};\\
B_{12}^{(1)} &= n_2 d_2(d_2 - 1) - \frac{32M_{12}^2 d_2}{n_1 n_2} - \frac{64M_{22}^2 d_2}{n_2^2} - \frac{16M_{22} d_2}{n_2} - 8d_2\Delta - d_2^2\Delta;\\
B_{12}^{(2)} &= M_{12} - \frac{16M_{12}^2}{n_1 n_2} - 10\Delta - 2\Delta^2;\\
B_{12}^{(3)} &= M_{12} - \frac{16M_{12}^2}{n_1 n_2} - 12\Delta - 2\Delta^2.
\end{aligned}
$$

Claim 5.12. Suppose some graph G appears during the execution of $\textsc{NoDoubles}_{12}$. Let $(v_1, \ldots, v_6, u_1, \ldots, u_4)$ be a set of ten vertices such that

(a) $v_1, \ldots, v_6, u_1, \ldots, u_4$ are distinct,

(b) $v_2, u_1, u_3 \in V_1, v_5, u_2, u_4 \in V_2$, Then:

(a) Let b_1 be the number of distinct 3-tuple (w_1, w_2, w_3) such that $w_2 \in V_1$, $w_1w_2, w_2w_3 \in E_S(G)$.

(b) Let b_2 be the number of distinct 3-tuple (w_1, w_2, w_3) such that w_1, w_2, w_3 distinct from (v_1, v_2, v_3), $w_2 \in V_2$, $w_1w_2, w_2w_3 \in E_S(G)$, $v_2w_2 \notin E(G)$.

(c) Let b_3 be the number of distinct 2-tuple (w_1, w_2) such that w_1, w_2 distinct from $v_1, \dots, v_6$, $w_1 \in V_1, w_2 \in V_2$, $w_1w_2 \in E_S(G)$, $w_1v_1, w_2v_4 \notin E(G)$.

(d) Let b_4 be the number of distinct 2-tuple (w_1, w_2) such that w_1, w_2 distinct from $v_1, \dots, v_6, u_1, u_2$, $w_1 \in V_1, w_2 \in V_2$, $w_1w_2 \in E_S(G)$, $w_1v_3, w_2v_6 \notin E(G)$.

We claim

(a) $B_{12}^{(0)} \le b_1 \le n_1d_1(d_1 - 1)$;

(b) $B_{12}^{(1)} \le b_2 \le n_2d_2(d_2 - 1)$;

(c) $B_{12}^{(2)} \le b_3 \le M_{12}$;

(d) $B_{12}^{(3)} \le b_4 \le M_{12}$.

We give detailed proof for b_2 and b_4 where the other two can be proved with similar ideas.

proof for (b). The value of b_2 is clearly bounded by $n_2d_2(d_2 - 1)$, where n_2 represents the number of choices for w_2, and $d_2(d_2 - 1)$ represents the number of choices for the two ordered neighbors w_1 and w_3 of w_2. To compute the lower bound, we need to subtract the number of following forbidden cases from $n_2d_2(d_2 - 1)$:

1. w_1w_2 or w_2w_3 is a double edge;
2. $v_1, v_2, v_3, w_1, w_2, w_3$ are not distinct vertices;
3. v_2w_2 is present.

For the second case, two subcases could be

2.i $\{w_1, w_2, w_3\} \cap \{v_1, v_2, v_3\} \neq \emptyset$;

2.ii w_1, w_2, w_3 are not distinct.

For case (1), it has a maximum of $4\overline{D}_{12}d_2 + 8\overline{D}_{22}d_2$ possible cases.

For case (2.i), it has a maximum of $8d_2\Delta$ possible cases.

For case (2.ii), the equation $n_2d_2(d_2-1)$ ensures that w_1 and w_3 do not coincide. If w_1 or w_3 coincides with w_2, it implies the existence of a loop at w_2, resulting in a maximum of $2\overline{L}_2d_2$ possible cases.

For case (3), it has a maximum of $d_2^2\Delta$ possible cases.

Hence we have

$$b_2 \geq n_2d_2(d_2-1) - \frac{32M_{12}^2d_2}{n_1n_2} - \frac{64M_{22}^2d_2}{n_2^2} - \frac{16M_{22}d_2}{n_2} - 8d_2\Delta - d_2^2\Delta = B_{12}^{(1)}$$

□

proof for (d). The value of b_4 is clearly bounded by M_{12}. To compute the lower bound, we need to subtract the number of following forbidden cases from M_{12}

1. w_1w_2 is a double edge
2. $v_1, \ldots, v_6, u_1, u_2, w_1, w_2$ are not distinct vertices
3. w_1v_3 or w_2v_6 is present

Since $w_1 \in V_1, w_2 \in V_2$, we don't need to consider the case of $w_1 = w_2$ in the second case.

For case (1), it has a maximum of $2\overline{D}_{12}$ possible cases.

For case (2), it has a maximum of 12Δ possible cases.

For case (3), it has a maximum of $2\Delta^2$ possible cases.

Hence we have:

$$b_4 \geq M_{12} - \frac{16M_{12}^2}{n_1n_2} - 12\Delta - 2\Delta^2 = B_{12}^{(3)}$$

□

Claim 5.12 fulfill the proof for Claim 5.7. Next, we complete the proof for Lemma 5.2, where the following claim shows that $B_{12}^{(i)} > 0$ for $0 \leq i \leq 3$ under the assumption of Lemma 5.2.

Claim 5.13. Provided that

$$\frac{M_{ij}^3 \Delta^2}{n_i^2 n_j^2} = o(1) \text{ for } 1 \le i \le j \le 2, \tag{5.5}$$

$B_{12}^{(i)} > 0$ for $0 \le i \le 3$

Proof. Let's compute the case of $i = 1$:

$$\begin{aligned} B_{12}^{(1)} &= n_2 d_2(d_2 - 1) - \frac{8M_{12}^2 d_2}{n_1 n_2} - \frac{48 M_{22}^2 d_2}{n_2^2} - \frac{12 M_{22} d_2}{n_2} - 8d_2\Delta - d_2^2\Delta \\ &= n_2 d_2 (d_2 - 1)\left(1 - O\left(\frac{M_{12}^2}{n_1 n_2^2 d_2} + \frac{M_{22}^2}{n_2^3 d_2} + \frac{M_{22}}{n_2^2 d_2} + \frac{\Delta}{n_2}\right)\right) \end{aligned}$$

The claim will be verified if we can demonstrate that $O\left(\frac{M_{12}^2}{n_1 n_2^2 d_2} + \frac{M_{22}^2}{n_2^3 d_2} + \frac{M_{22}}{n_2^2 d_2} + \frac{\Delta}{n_2}\right) = o(1)$. In fact, a stronger form of this will be demonstrated in Lemma 5.16, where we are going to show that

$$O\left(\frac{M_{12}^2}{n_1 n_2^2 d_2} + \frac{M_{22}^2}{n_2^3 d_2} + \frac{M_{22}}{n_2^2 d_2} + \frac{\Delta}{n_2}\right) \cdot O\left(\frac{M_{12}^2}{n_1 n_2}\right) = o(1).$$

Without loss of generality, we may assume that

$$\frac{8M_{12}^2}{n_1 n_2} \ge 1,$$

as otherwise $\overline{D}_{12} < 1$ by (3.1), in which case, G_0 would have no crossing double edges and NoDoubles$_{12}$ would not be executed.

Hence we have $\frac{M_{12}^2}{n_1 n_2} = \Omega(1)$ and it can be concluded that $B_{12}^{(1)} > 0$. An analogous proof can be applied for $i \in \{0, 2, 3\}$. □

Proof for Lemma 4.2. Claim 5.13 addresses the positivity part of the proof, it remains to show that $b(F) \ge B_{12}^{(i)}$ for all $F \in \mathcal{F}_i$ for $0 \le i \le 3$. We prove for the case of $i = 1$, the idea for the remaining three cases is identical.

Suppose $F = (G, C_1^{(v_1,v_2,v_3)}) \in \mathcal{F}_1$ for some vertices v_1, v_2, v_3. $b(F)$ is the number of $F' \in \mathcal{F}_2 = \mathcal{F}_2'$ (by Claim 5.7) such that $P(F') = F$, which is equivalent to the number of (w_1, w_2, w_3) that satisfy the following conditions:

(a) w_1, w_2, w_3 are distinct vertices, different from v_1, v_2, v_3.

(b) $w_2 \in V_2$.

(c) $w_1w_2, w_2w_3 \in E_S(G)$ and $v_2w_2 \notin E(G)$.

By Claim 5.12, we may conclude that have $b(F) \geq B_{12}^{(1)}$. $\square$

5.4.3 Rejection probability of NoDoubles$_{12}$

In this section, we analyze the rejection probabilities of both f-rejections and b-rejections.

Corollary 5.14. *For every $0 \leq m \leq \overline{D}_{12}$ and every $G' \in \mathcal{H}_m$. Let S be a switching removes crossing double edges between V_1, V_2 and produce G', then we have*

$$
\begin{aligned}
\frac{B_{12}^{(0)}}{b_{12}^{(1)}(G',S)} &\geq 1 - O\left(\frac{M_{12}^2}{n_1^2 n_2 d_1} + \frac{M_{11}^2}{n_1^3 d_1} + \frac{M_{11}}{n_1^2 d_1}\right) = 1 - O\left(\frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{11}}\right)\\
\frac{B_{12}^{(1)}}{b_{12}^{(2)}(G',S)} &\geq 1 - O\left(\frac{M_{12}^2}{n_1 n_2^2 d_2} + \frac{M_{22}^2}{n_2^3 d_2} + \frac{M_{22}}{n_2^2 d_2} + \frac{\Delta}{n_2}\right) = 1 - O\left(\frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{22}}\right)\\
\frac{B_{12}^{(2)}}{b_{12}^{(3)}(G',S)} &\geq 1 - O\left(\frac{M_{12}}{n_1 n_2} + \frac{\Delta^2}{M_{12}}\right) = 1 - O\left(\frac{\Delta^2}{M_{12}}\right)\\
\frac{B_{12}^{(3)}}{b_{12}^{(4)}(G',S)} &\geq 1 - O\left(\frac{M_{12}}{n_1 n_2} + \frac{\Delta^2}{M_{12}}\right) = 1 - O\left(\frac{\Delta^2}{M_{12}}\right)
\end{aligned}
$$

Proof. The initial inequality is derived by Claim 5.12. The subsequent equality is obtained by using the fact that $M_{ii} \leq n_i d_i$ and $M_{ij} \leq n_i d_i$. $\square$

Lemma 5.15. *Let $G \in \mathcal{H}_m$ for some $m \leq \overline{D}_{12}$. For every $\tau \in \{D_{12}^i : 1 \leq i \leq 16\}$,*

$$\overline{f_\tau}(1 - O(\xi)) \leq f_\tau(G) \leq \overline{f_\tau},$$

where $\xi = \frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{11}} + \frac{\Delta^2}{M_{22}}$.

Proof. We prove the case that $\tau = D^5_{12}$ where the switching is shown in Figure 5.10. The proof for all the other cases is analogous.

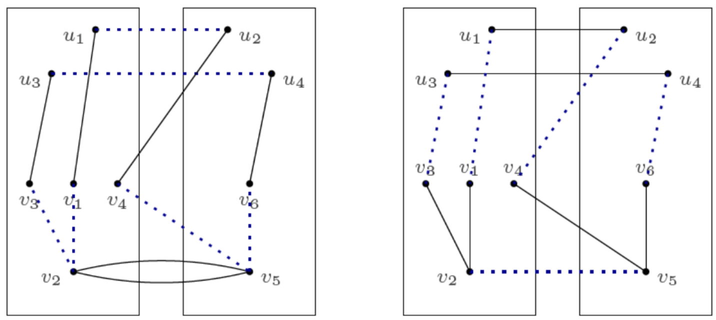


Figure 5.10: D^5_{12} switching

Suppose G has m double edges between V_1, V_2. We showed that the upper bound for $f_{D^5_{12}}(G)$ is $\overline{f}_{D^5_{12}} = 8mM_{12}M_{11}^2M_{22}$ by Lemma 5.1. To find the lower bound, we need to subtract the number of the following forbidden cases:

1. Edge other than v_2v_5 is a double edge.
2. $v_1, ..., v_6, u_1, ..., u_4$ are not distinct.
3. some of the non-edges present

For case (1),

1a) If v_4u_2 is a double edge, there are at most $O(m^2M_{11}^2M_{22})$ choices.

1b) If v_1u_1 or v_3u_3 is a double edge, there are at most $O(mM_{12}M_{11}M_{22}\overline{D}_1)$ choices.

1c) If v_6u_4 is a double edge there are at most $O(mM_{12}M_{11}^2\overline{D}_2)$ choices.

For case (2),

2a) If some of $\{v_1, v_3, u_1, u_3\}$ coincides with other vertices, there are at most $O(mM_{12}M_{11}M_{22}\Delta)$ choices. (Please note that the scenario where v_1 coincides with u_1, or in other words, u_1 and v_1 form a loop, is included in this calculation. This is because there are at most $O\left(mM_{12}M_{11}M_{22}\frac{M_{11}}{n_1}\right)$ such cases. And given that $M_{11} \leq n_1\Delta$, it implies $\frac{M_{11}}{n_1} \leq \Delta$.).

2b) If v_4 or u_2 coincide with other vertices, there are at most $O(mM_{11}^2 M_{22}\Delta)$ choices.

2c) If v_4 or u_6 coincides with other vertices, there are at most $O(mM_{12}M_{11}^2\Delta)$ choices.

For case (3), suppose there is some edge xy which should be a non-edge but it presents.

3a) If $\{x, y\} \cap \{v_1, u_1, v_3, u_3\} \neq \emptyset$, there are at most $O(mM_{12}M_{11}M_{22}\Delta^2)$ choices.

3b) If $\{x, y\} \cap \{u_2, v_4\} \neq \emptyset$, there are at most $O(mM_{11}^2 M_{22}\Delta^2)$ choices.

3c) If $\{x, y\} \cap \{v_6, u_4\} \neq \emptyset$, there are at most $O(mM_{12}M_{11}^2\Delta^2)$ choices.

Hence we have the lower bound for $f_{D_{12}^5}(G)$ is

$$\begin{aligned}\underline{f}_{D_{12}^5}(G) = 8mM_{12}M_{11}^2M_{22} - &O(m^2M_{11}^2M_{22}) - O(mM_{12}M_{11}M_{22}\overline{D_1}) - O(mM_{12}M_{11}^2\overline{D_2}) \\ &-O(mM_{11}^2M_{22}\Delta) - O(mM_{12}M_{11}M_{22}\Delta) - O(mM_{12}M_{11}^2\Delta) \\ &-O(mM_{11}^2M_{22}\Delta^2) - O(mM_{12}M_{11}M_{22}\Delta^2) - O(mM_{12}M_{11}^2\Delta^2)\end{aligned}$$

Recall that $\overline{D_i} = \frac{6M_{ii}^2}{n_i^2} = O(\frac{M_{ii}^2}{n_i^2}), m \leq \overline{D_{12}} = \frac{8M_{12}^2}{n_1n_2} = O(\frac{M_{12}^2}{n_1n_2})$

Hence we have

$$\begin{aligned}\underline{f}_{D_{12}^5}(G) =& 8mM_{12}M_{11}^2M_{22} - O(m^2M_{11}^2M_{22}) - O(mM_{12}M_{11}M_{22}\frac{M_{11}^2}{n_1^2}) - O(mM_{12}M_{11}^2\frac{M_{22}^2}{n_2^2}) \\ &- O(mM_{11}^2M_{22}\Delta^2) - O(mM_{12}M_{11}M_{22}\Delta^2) - O(mM_{12}M_{11}^2\Delta^2) \\ =& 8mM_{12}M_{11}^2M_{22}\left(1 - O\left(\frac{m}{M_{12}} + \frac{M_{11}}{n_1^2} + \frac{M_{22}}{n_2^2} + \frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{11}} + \frac{\Delta^2}{M_{22}}\right)\right) \\ =& \overline{f}_{D_{12}^5}\left(1 - O\left(\frac{M_{12}}{n_1n_2} + \frac{M_{11}}{n_1^2} + \frac{M_{22}}{n_2^2} + \frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{11}} + \frac{\Delta^2}{M_{22}}\right)\right)\end{aligned}$$

Using the fact that $M_{ii} < n_id_i$ and $M_{ij} \leq n_id_i$, we get

$$\overline{f}_{D_{12}^5}\left(1 - O\left(\frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{11}} + \frac{\Delta^2}{M_{22}}\right)\right) \leq f_{D_{12}^5}(G) \leq \overline{f}_{D_{12}^5}$$

Using a similar idea, we have the following:

$$\overline{f}_{D_{12}^i}\left(1 - O\left(\frac{\Delta^2}{M_{12}}\right)\right) \leq f_{D_{12}^i}(G) \leq \overline{f}_{D_{12}^i} \text{ for } i \in \{6\} \tag{5.6a}$$

$$\overline{f}_{D_{12}^i}\left(1 - O\left(\frac{\Delta^2}{M_{11}} + \frac{\Delta^2}{M_{12}}\right)\right) \leq f_{D_{12}^i}(G) \leq \overline{f}_{D_{12}^i} \text{ for } i \in \{1, 2, 3\} \tag{5.6b}$$

$$\overline{f}_{D_{12}^i}\left(1 - O\left(\frac{\Delta^2}{M_{22}} + \frac{\Delta^2}{M_{12}}\right)\right) \leq f_{D_{12}^i}(G) \leq \overline{f}_{D_{12}^i} \text{ for } i \in \{12, 13, 16\} \tag{5.6c}$$

$$\overline{f}_{D_{12}^i}\left(1 - O\left(\frac{\Delta^2}{M_{11}} + \frac{\Delta^2}{M_{22}}\right)\right) \leq f_{D_{12}^i}(G) \leq \overline{f}_{D_{12}^i} \text{ for } i \in \{11\} \tag{5.6d}$$

$$\overline{f}_{D_{12}^i}\left(1 - O\left(\frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{11}} + \frac{\Delta^2}{M_{22}}\right)\right) \leq f_{D_{12}^i}(G) \leq \overline{f}_{D_{12}^i} \text{ for } i \in \{4, 5, 7, 8, 9, 10, 14, 15\} \tag{5.6e}$$

Hence for $\tau \in \{D_{12}^i : 1 \leq i \leq 16\}$

$$\overline{f}_\tau(1 - O(\xi)) \leq f_\tau(G) \leq \overline{f}_\tau$$

for $\xi = \frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{11}} + \frac{\Delta^2}{M_{22}}$

□

By employing a similar proof strategy as presented in [3, **Lemma 16**], we derive the following lemma.

Lemma 5.16. *The probability of an f-rejection or b-rejection occurring in* NoDoubles$_{12}$ *is*

$$1 - O\left(\frac{M_{12}\Delta^2}{n_1 n_2} + \frac{M_{12}^2\Delta^2}{M_{11}n_1 n_2} + \frac{M_{12}^2\Delta^2}{M_{22}n_1 n_2}\right)$$

Proof. By Corollary 5.14 and Lemma 5.15. The probability that no rejection occurs in a single iteration of $\textsc{NoDoubles}_{12}$ is

$$\begin{aligned} p \geq & \left(1 - O\left(\frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{11}} + \frac{\Delta^2}{M_{22}}\right)\right) \\ & \cdot \left(1 - O\left(\frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{11}}\right)\right) \\ & \cdot \left(1 - O\left(\frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{22}}\right)\right) \\ & \cdot \left(1 - O\left(\frac{\Delta^2}{M_{12}}\right)\right)^2 \end{aligned}$$

Hence we have

$$p \geq \exp\left(-O\left(\frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{11}} + \frac{\Delta^2}{M_{22}}\right)\right)$$

Given that the maximum number of double edges between V_1, V_2 is $\overline{D}_{12} = \frac{8M_{12}^2}{n_1 n_2}$, the $\textsc{NoDoubles}_{12}$ algorithm iterates at most $\frac{8M_{12}^2}{n_1 n_2}$ times. Therefore, the probability that no rejection occurs during the execution of $\textsc{NoDoubles}_{12}$ is

$$\begin{aligned} p^{\frac{8M_{12}^2}{n_1 n_2}} \geq & \exp\left(-O\left(\frac{\Delta^2}{M_{12}} + \frac{\Delta^2}{M_{11}} + \frac{\Delta^2}{M_{22}}\right)\frac{8M_{12}^2}{n_1 n_2}\right) \\ = & \exp\left(-O\left(\frac{M_{12}\Delta^2}{n_1 n_2} + \frac{M_{12}^2\Delta^2}{M_{11} n_1 n_2} + \frac{M_{12}^2\Delta^2}{M_{22} n_1 n_2}\right)\right) \\ = & 1 - O\left(\frac{M_{12}\Delta^2}{n_1 n_2} + \frac{M_{12}^2\Delta^2}{M_{11} n_1 n_2} + \frac{M_{12}^2\Delta^2}{M_{22} n_1 n_2}\right) \end{aligned}$$

Hence, the probability that no rejection occurs during $\textsc{NoDoubles}_{12}$ can be expressed as

$$1 - O\left(\frac{M_{12}\Delta^2}{n_1 n_2} + \frac{M_{12}^2\Delta^2}{M_{11} n_1 n_2} + \frac{M_{12}^2\Delta^2}{M_{22} n_1 n_2}\right) \tag{5.7}$$

□

5.4.4 The remaining phases

$\textsc{NoDoubles}_i$ and $\textsc{NoLoops}_i$ for $i = 1, 2$ follow exactly the same frame as $\textsc{NoDoubles}_{12}$ defined in Section 5.3.1. To complete the definitions we only need to specify parameters p_τ^m,

$\overline{f}^m$ and all the $B_i^{(j)}$ analogous to the $B_{12}^{(j)}$ in (5.3a)–(5.3d). These parameters are specified in Section 6.3 for general q. See (6.4a)–(6.4d) and (6.6a)–(6.6b). The uniformity of the output after each phase is guaranteed by the call of the subprocedure RelaxGraph, and by Lemma 5.5. The running time in each phase for general q is discussed in Section 6.4. See Lemma 6.1. We state below the corresponding lemmas for the special case $q = 2$ so that we can complete the proof for Theorem 3.3 for the case $q = 2$.

Lemma 5.17. *Suppose that there is no rejection in any of the phases* NoDoubles$_i$ *and* NoLoops$_i$ *for $i = 1, 2$. Then, the total running time of these four phases is bounded by* $O(\frac{M_{11}^2\Delta^2}{n_1^2} + \frac{M_{22}^2\Delta^2}{n_2^2} + \frac{M_{11}\Delta^2}{n_1} + \frac{M_{22}\Delta^2}{n_2})$.

Proof. This follows as a corollary of Lemma 6.1.

Lemma 5.18. *The probability of any rejection occurring during* NoDoubles$_i$ *for $i \in \{1, 2\}$ is:*

$$O\left(\frac{M_{ii}^2}{n_i^2} \cdot \sum_{j\in\{1,2\}} \frac{\Delta^2}{M_{ij}}\right)$$

Proof. This follows as a corollary of Lemma 6.2.

Lemma 5.19. *The probability of any rejection occurring during* NoLoops$_i$ *for $i \in \{1, 2\}$ is:*

$$O\left(\frac{M_{ii}}{n_i} \cdot \sum_{j\in\{1,2\}} \frac{\Delta^2}{M_{ij}}\right)$$

Proof. This follows as a corollary of Lemma 6.2.

5.5 Proof of Theorem 3.3 for $q = 2$

Without loss of generality, we may assume that $\overline{D}_{12} > 1$ when discussing running time and rejection probability of NoDoubles$_{12}$, as otherwise, G_0 would have no crossing edges between V_1, V_2 and NoDoubles$_{12}$ would not be executed.

Similarly, we may assume that $\overline{D}_{ii} > 1$ when discussing NoDoubles$_i$ for $i \in \{1, 2\}$, and assume that $\overline{L}_i > 1$ when discussing NoLoops$_i$ for $i \in \{1, 2\}$.

Lemma 5.20. *Provided that*

$$\frac{M_{ij}^3 \Delta^2}{n_i^2 n_j^2} = o(1) \text{ for all } 1 \le i \le j \le 2, \tag{5.8}$$

The probability of an f-rejection or b-rejection occurring in NoDoubles$_{12}$*,* NoDoubles$_1$*,* NoDoubles$_2$*,* NoLoops$_1$*, and* NoLoops$_2$ *is* $o(1)$*.*

Proof. Let's start with NoDoubles$_{12}$, by Lemma 5.16, it remains to show that $\frac{M_{12}\Delta^2}{n_1 n_2} + \frac{M_{12}^2\Delta^2}{M_{11}n_1 n_2} + \frac{M_{12}^2\Delta^2}{M_{22}n_1 n_2} = o(1)$ under the assumption of $\frac{M_{12}^3\Delta^2}{n_1^2 n_2^2} = o(1)$.

First, we note that $\frac{M_{12}^3\Delta^2}{n_1^2 n_2^2} = \frac{M_{12}\Delta^2}{n_1 n_2} \cdot \frac{M_{12}^2}{n_1 n_2}$, where we may assume $\overline{D}_{12} = \frac{8M_{12}^2}{n_1 n_2} > 1$ by (3.1). Hence, $\frac{M_{12}^2}{n_1 n_2} = \Omega(1)$, which leads us to the conclusion that $\frac{M_{12}\Delta^2}{n_1 n_2} = O\left(\frac{M_{12}^3\Delta^2}{n_1^2 n_2^2}\right) = o(1)$.

To demonstrate that $\frac{M_{12}^2\Delta^2}{M_{11}n_1 n_2} + \frac{M_{12}^2\Delta^2}{M_{22}n_1 n_2}$ is $o(1)$, let's take $\frac{M_{12}^2\Delta^2}{M_{11}n_1 n_2}$ as a example.

We notice that $\frac{M_{12}^2\Delta^2}{M_{11}n_1 n_2} \cdot \frac{M_{11}M_{12}}{n_1 n_2} = \frac{M_{12}^3\Delta^2}{n_1^2 n_2^2}$. Therefore, if we can show that $\frac{M_{11}M_{12}}{n_1 n_2}$ is not negligibly small (i.e., $\frac{M_{11}M_{12}}{n_1 n_2} \neq o(1)$), it follows that $\frac{M_{12}^2\Delta^2}{M_{11}n_1 n_2} = O\left(\frac{M_{12}^3\Delta^2}{n_1^2 n_2^2}\right) = o(1)$.

Before we proceed, let's first examine the origin of the error term $\frac{M_{12}^2\Delta^2}{M_{11}n_1 n_2}$.

The error $O\left(\frac{\Delta^2}{M_{11}}\right)$ from Lemma 5.15 leads to an error of $O\left(\frac{M_{12}^2\Delta^2}{M_{11}n_1 n_2}\right)$ in the computation. Looking into the outcome (5.6) from Lemma 5.15, we notice that this error emerges for all types of switchings require a single edge in V_1 as the starting graph on which the switching occurs. Specifically, these switchings correspond to types D_{12}^i for $i \in \{1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 14, 15\}$.

Let's calculate the probability of choosing these types of switchings in each round and multiply it by the number of iterations, which is $O(\frac{M_{12}^2}{n_1 n_2})$. The probability of choosing these types of switchings in each round is as follows:

$$
\begin{aligned}
&\frac{\sum_{i\in\{1,2,3,4,5,7,8,9,10,11,14,15\}}\overline{f}_{D_{12}^i}}{\sum_{i\in[16]}\overline{f}_{D_{12}^i}}\\
=&\frac{\sum_{i\in\{1\}}\overline{f}_{D_{12}^i}}{\sum_{i\in[16]}\overline{f}_{D_{12}^i}}+\frac{\sum_{i\in\{2,3\}}\overline{f}_{D_{12}^i}}{\sum_{i\in[16]}\overline{f}_{D_{12}^i}}+\frac{\sum_{i\in\{4,5\}}\overline{f}_{D_{12}^i}}{\sum_{i\in[16]}\overline{f}_{D_{12}^i}}+\frac{\sum_{i\in\{7,8,9,10\}}\overline{f}_{D_{12}^i}}{\sum_{i\in[16]}\overline{f}_{D_{12}^i}}+\\
&\frac{\sum_{i\in\{11\}}\overline{f}_{D_{12}^i}}{\sum_{i\in[16]}\overline{f}_{D_{12}^i}}+\frac{\sum_{i\in\{14,15\}}\overline{f}_{D_{12}^i}}{\sum_{i\in[16]}\overline{f}_{D_{12}^i}}\\
<&\frac{\sum_{i\in\{1\}}\overline{f}_{D_{12}^i}}{\overline{f}_{D_{12}^2}}+\frac{\sum_{i\in\{2,3\}}\overline{f}_{D_{12}^i}}{\overline{f}_{D_{12}^6}}+\frac{\sum_{i\in\{4,5\}}\overline{f}_{D_{12}^i}}{\overline{f}_{D_{12}^7}}+\frac{\sum_{i\in\{7,8,9,10\}}\overline{f}_{D_{12}^i}}{\overline{f}_{D_{12}^{12}}}+\\
&\frac{\sum_{i\in\{11\}}\overline{f}_{D_{12}^i}}{\overline{f}_{D_{12}^{14}}}+\frac{\sum_{i\in\{14,15\}}\overline{f}_{D_{12}^i}}{\overline{f}_{D_{12}^{16}}}\\
=&\frac{4mM_{12}^2M_{11}^2}{2mM_{12}^3M_{11}}+\frac{4mM_{12}^3M_{11}}{mM_{12}^4}+\frac{16mM_{12}M_{11}^2M_{22}}{4mM_{12}^2M_{11}M_{22}}+\frac{16mM_{12}^2M_{11}M_{22}}{2mM_{12}^3M_{22}}+\\
&\frac{16mM_{11}^2M_{22}^2}{8mM_{12}M_{11}M_{22}^2}+\frac{16mM_{12}M_{11}M_{22}^2}{4mM_{12}^2M_{22}^2}\\
=&O\left(\frac{M_{11}}{M_{12}}\right)
\end{aligned}
$$

Thus, by multiplying it by the number of iterations, we derive:

$$
O\left(\frac{M_{11}}{M_{12}}\cdot\frac{M_{12}^2}{n_1n_2}\right)=O\left(\frac{M_{11}M_{12}}{n_1n_2}\right)
$$

If $\frac{M_{11}M_{12}}{n_1n_2}=o(1)$, the error resulting from these switchings can be disregarded. Consequently, the term $\frac{M_{12}^2\Delta^2}{M_{11}n_1n_2}$ in the rejection probability formula in (5.7) is deemed negligible.

On the other hand, if $\frac{M_{11}M_{12}}{n_1n_2}\neq o(1)$, then $\frac{M_{12}^2\Delta^2}{M_{11}n_1n_2}=O\left(\frac{M_{12}^3\Delta^2}{n_1^2n_2^2}\right)=o(1)$, in alignment with our claim earlier.

A similar discussion can be carried out for $\frac{M_{12}^2\Delta^2}{M_{22}n_1n_2}$. Consequently, we can conclude that $\frac{M_{12}\Delta^2}{n_1n_2}+\frac{M_{12}^2\Delta^2}{M_{11}n_1n_2}+\frac{M_{12}^2\Delta^2}{M_{22}n_1n_2}=o(1)$. Therefore, the probability of any rejection occurring during NoDoubles$_{12}$ is $o(1)$.

Next, by Lemma 5.18 the probability of any rejection occurring during NoDoubles$_i$ for $i \in \{1,2\}$ is

$$O\left(\frac{M_{ii}^2}{n_i^2} \cdot \sum_{j \in \{1,2\}} \frac{\Delta^2}{M_{ij}}\right).$$

Use a similar discussion, such probability is bounded by $\frac{M_{ii}^3 \Delta^2}{n_i^4} = o(1)$ for $i \in \{1,2\}$.

Next, by Lemma 5.19 the probability of any rejection occurring during NoDoubles$_i$ for $i \in \{1,2\}$ is

$$O\left(\frac{M_{ii}}{n_i} \cdot \sum_{j \in \{1,2\}} \frac{\Delta^2}{M_{ij}}\right)$$

Use a similar discussion, such probability is bounded by $\frac{M_{ii}\Delta^2}{n_i^2}$ for $i \in \{1,2\}$.

Where we have $\frac{M_{ii}\Delta^2}{n_i^2} \cdot \frac{M_{ii}^2}{n_i^2} = \frac{M_{ii}^3 \Delta^2}{n_i^4} = o(1)$. Since we may assume that $\overline{L}_i = \frac{8M_{ii}}{n_i} > 1$, we have $\frac{M_{ii}}{n_i} = \Omega(1)$. Hence $\frac{M_{ii}\Delta^2}{n_i^2} = o(1)$ for $i \in \{1,2\}$. □

Lemma 5.21. *Provided that*

$$\frac{M_{ij}^3 \Delta^2}{n_i^2 n_j^2} = o(1) \textit{ for all } 1 \leq i \leq j \leq 2, \tag{5.9}$$

Under the assumption that no rejection occurs, the running time of NoDoubles$_{12}$, NoDoubles$_1$, NoDoubles$_2$, NoLoops$_1$*, and* NoLoops$_2$ *can be bounded by* $O(n_1 d_1 + n_2 d_2)$.

Proof. We start with NoDoubles$_{12}$, by Lemma 5.10, the running time of NoDoubles$_{12}$ is $O(\frac{M_{12}^2\Delta^2}{n_1 n_2})$, where we have

$$\frac{M_{12}^2\Delta^2}{n_1 n_2} \cdot \frac{M_{12}^2}{n_1 n_2} = \frac{M_{11}^3\Delta^2}{n_1^2 n_2^2} \cdot M_{12} = o(M_{12}) = O(n_1 d_1).$$

Since we may assume that $\overline{D}_{12} > 1$ and hence $\frac{M_{12}^2}{n_1 n_2} = \Omega(1)$. We have $\frac{M_{12}^2\Delta^2}{n_1 n_2} = O(n_1 d_1)$.

Similarly, we can show that the running time of NoDoubles$_i$ is $O(n_i d_i)$ for $i \in \{1,2\}$

For $\textsc{NoLoops}_1$, the running time is $O(\frac{M_{11}\Delta^2}{n_1})$ by Lemma 5.17 where we have

$$\frac{M_{11}\Delta^2}{n_1} \cdot \frac{M_{11}^3}{n_1^3} = \frac{M_{11}^3\Delta^2}{n_1^4} \cdot M_{11} = o(M_{11}) = O(n_1 d_1).$$

Since we may assume that $\overline{L}_1 > 1$ and hence $\frac{M_{11}}{n_1} = \Omega(1)$. We have $\frac{M_{11}\Delta^2}{n_1} = O(n_1 d_1)$. Similarly, we can show that the running time of $\textsc{NoLoops}_2$ is $O(n_2 d_2)$ $\square$

Proof of Theorem 3.3 for the case $q = 2$. By the assumption of the theorem, the probability of any rejection occurring during JDM_GEN is $\frac{3}{8} + o(1)$ by Lemma 3.2 and Lemma 5.20. Then by Lemma 4.6 and Lemma 5.21, the running time of JDM_GEN is $O(n_1 d_1 + n_2 d_2)$ where the time for the preparation of k_v, as discussed in Lemma 5.10, is also included.

Regarding uniformity, Lemma 4.3 guarantees the uniformity of G_0. Then the call of the subprocedure RelaxGraph in each phase, together with Lemma 5.5, ensures that the uniformity is preserved throughout the process. $\square$

Chapter 6

General case

In this chapter, we discuss how to extend the algorithm for $q = 2$ to the general case. The extension is natural and straightforward, and thus we only briefly describe the main difference.

Recall that in the general setting we have $\boldsymbol{V} = (V_1, V_2, ..., V_q)$ for some $q \geq 2$ that is a family of q disjoint vertex sets, M is a $q \times q$ matrix where M_{ij} specifies the number of required edges between V_i and V_j in the graph we aim to generate, and $\boldsymbol{d} = (d_1, \ldots, d_q)$ are positive integers specifying the degrees of the vertices in each vertex set.

JDM_GEN for general q is defined in the same way as for $q = 2$ in Section 5.3.5. After calling PairingGen, it calls a sequence of procedures to remove multiple edges of different types. I.e., procedures NoDoublesij for removing multiple edges crossing V_i and V_j, for $1 \leq i < j \leq q$, procedures NoDoubles$i$ for removing multiple edges inside V_i, and NoLoops$i$ for removing loops inside V_i, for $1 \leq i \leq q$. These procedures follow exactly the same frame as NoDoubles12 given in Section 5.3.1. In each of these procedures, a set of switching types and switching operations will be used, which are similar to the ones in Section 5.2 for $q = 2$. In each step, a switching type will be chosen according to distribution p_τ^m, and then a uniform switching of the selected type will be chosen. If the switching is not rejected, then a new multigraph is obtained by performing that switching. To complete the definitions of these procedures, it suffices to define the set of switchings, and specify the set of parameters p_τ^m, $\overline{f}_\tau^m$ and the $B_i^{(j)}$ used to perform the b-rejections. In Section 6.1 we explain how to extend the switchings for $q = 2$ to the switchings for general q. In Section 6.3, we set all parameters required by the algorithm. Finally, in Section 6.3 we discuss the uniformity and the running time, and complete the proof for the main theorem in the general case.

6.1 Switchings

The set of switchings for general q extends naturally from those for $q = 2$. We briefly describe the differences.

6.1.1 Switchings to remove crossing double edges between V_i and V_j

For the case when $q = 2$, there are 16 different types of switchings to remove double edges between V_1 and V_2. As shown in the image in Section 5.2.1, the number 16 corresponds to all possible combinations of the vertex sets to which the vertices v_1, v_3, v_4, and v_6 can belong, where each of these vertices can be in either V_1 or V_2. Therefore, there are $2^4 = 16$ different types of switchings available.

Now for the general case where there are q vertex sets $V_1, V_2, ..., V_q$. If we want to remove double edges between any two distinct parts V_i and V_j, there will be q^4 different types of switchings. Since it is impractical to use images to display all switchings, we will provide a general definition for all the switching.

Suppose we have a double edge v_2v_5 where $v_2 \in V_i$ and $v_5 \in V_j$ for $i \neq j$. The switching that removes this double edge involves an ordered set of ten vertices ($v_1, v_2, v_3, v_4, v_5, v_6, u_1, u_2, u_3, u_4$) such that

1. $v_1, v_2, v_3, v_4, v_5, v_6, u_1, u_2, u_3, u_4$ are distinct vertices
2. $u_1, u_3 \in V_i, u_2, u_4 \in V_j$ are allocated for edge balancing purposes.
3. $v_1u_1, v_3u_3, v_4u_2, v_6u_4$ induce four simple edges.
4. $v_1v_2, v_2v_3, v_4v_5, v_5v_6, u_1u_2, u_3u_4$ are non edges.

The switching replaces the edges mentioned above with $v_1v_2, v_2v_3, v_4v_5, v_5v_6, u_1u_2, u_3u_4$.

Indeed, the degree sequence of the 10 vertices involved stays constant. Concerning the number of edges between distinct vertex sets, the edges v_1u_1 and v_3u_3 are replaced by v_1v_2 and v_2v_3 respectively. As v_2, u_1, and u_3 all belong to V_i, this guarantees that the numbers of edges between V_i and the vertex sets where v_1 and v_3 belong remain unchanged. In a similar manner, the numbers of edges between V_j and the vertex sets where v_2 and v_4

belong are preserved. Finally, the double edge v_2v_5 is removed, and two simple edges u_1u_2 and u_3u_4 are added. Hence, the count of edges between V_i and V_j remains constant.

The type of switchings is determined by the vertex sets to which v_1, v_3, v_4, and v_6 are assigned. Each of these vertices may belong to any of the q sets $V_1, V_2, ..., V_q$, leading to q^4 possible types of switchings.

6.1.2 Switchings to remove double edges within each V_i

There are q^4 different types of switchings available to remove double edges within each V_i. Suppose we have a double edge v_2v_5 in V_i for some i. The switching that removes this double edge involves an ordered set of ten vertices $(v_1, v_2, v_3, v_4, v_5, v_6, u_1, u_2, u_3, u_4)$ such that

1. $v_1, v_2, v_3, v_4, v_5, v_6, u_1, u_2, u_3, u_4$ are distinct vertices
2. $u_1, u_2, u_3, u_4 \in V_i$ are allocated for edge balancing purposes.
3. $v_1u_1, v_3u_3, v_4u_2, v_6u_4$ induce four simple edges.
4. $v_1v_2, v_2v_3, v_4v_5, v_5v_6, u_1u_2, u_3u_4$ are non edges.

The switching replaces the edges mentioned above with $v_1v_2, v_2v_3, v_4v_5, v_5v_6, u_1u_2, u_3u_4$.

It can be verified that the degree sequence of the vertices remains unchanged, and the number of edges between any two vertex sets remains unaffected by the switchings.

Similarly, the type of switchings is determined by the vertex sets to which v_1, v_3, v_4, and v_6 are assigned.

6.1.3 Switchings to remove loops within each V_i

There are q^2 different types of switchings available to remove loops within each V_i. Suppose we have a loop at vertex $v_2 \in V_i$ for some i. The switching that removes this loop involves an ordered set of five vertices $(v_1, v_2, v_3, v_4, v_5)$ such that

1. v_1, v_2, v_3, v_4, v_5 are distinct vertices

2. $v_4, v_5 \in V_i$

3. v_1v_4, v_3v_5 induce two simple edges.

4. v_1v_2, v_2v_3, v_4v_5 are non edges.

The switching replaces the edges mentioned above with v_1v_2, v_2v_3, v_4v_5.

Such switching preserves the degree sequence and maintains the number of edges between vertex sets, and the type of switchings is determined by the vertex set to which v_1, v_3 belong.

6.2 JDM_GEN

Now, we define the algorithm JDM_GEN for the general case. The parameters used for validation will be defined in Section 6.3.

Algorithm JDM_GEN($\boldsymbol{V}, M, \boldsymbol{d}$)

```
(P, G) = PairingGen(V, M, d)
if not (B_ij^(k) > 0 for 0 ≤ k ≤ 3 for all 1 ≤ i ≤ j ≤ q and
        B_i^(k) > 0 for 0 ≤ k ≤ 1 for i ∈ [q]) then
    Reject if G is not a simple graph
end if
for each 1 ≤ i < j ≤ q do
    NoDoubles_ij(G)
end for
for each i ∈ [q] do
    NoDoubles_i(G)
end for
for each i ∈ [q] do
    NoLoops_i(G)
end for
return G
```

Each procedure called in the algorithm can be defined by replacing the parameters in NoDoubles$_{12}$ with the parameters defined in Section 6.3.

By applying the approach we used to derive Corollary 5.9, it is equivalent to define RelaxGraph called in each procedure as follows. The parameters used in them are also defined in Section 6.3.

For RelaxGraph(G, S) called in NoDoubles$_{ij}$ for some G and S:

---

procedure RelaxGraph(G, S)

  Reject with probability: $1 - \frac{B_{ij}^{(0)}}{b_{ij}^{(1)}(S(G),S)} \frac{B_{ij}^{(1)}}{b_{ij}^{(2)}(S(G),S)} \frac{B_{ij}^{(2)}}{b_{ij}^{(3)}(S(G),S)} \frac{B_{ij}^{(3)}}{b_{ij}^{(4)}(S(G),S)}$

end procedure

For RelaxGraph(G, S) called in NoDoubles$_{i}$ for some G and S:

---

procedure RelaxGraph(G, S)

  Reject with probability: $1 - \frac{B_{ii}^{(0)}}{b_{ii}^{(1)}(S(G),S)} \frac{B_{ii}^{(1)}}{b_{ii}^{(2)}(S(G),S)} \frac{B_{ii}^{(2)}}{b_{ii}^{(3)}(S(G),S)} \frac{B_{ii}^{(3)}}{b_{ii}^{(4)}(S(G),S)}$

end procedure

For RelaxGraph(G, S) called in NoLoops$_{i}$ for some G and S:

---

procedure RelaxGraph(G, S)

  Reject with probability: $1 - \frac{B_{i}^{(0)}}{b_{i}^{(1)}(S(G),S)} \frac{B_{i}^{(1)}}{b_{i}^{(2)}(S(G),S)}$

end procedure

6.3 Parameters

6.3.1 Parameters for NoDoubles$_{ij}$

Suppose we are removing double edges from some graph G during the execution of NoDoubles$_{ij}$, where G contains m crossing double edges between V_i and V_j for some $0 < m \leq \overline{D}_{ij}$. We use D_{ij}^1 to $D_{ij}^{q^4}$ to represent all the types of switchings that are used to remove the crossing double edges between V_i and V_j.

For each type $\tau \in \{D_{ij}^k : 1 \leq k \leq q^4\}$, $f_\tau(G)$ represents the number of type τ switching that can be applied on G.

Given a specific type $\tau \in \{D_{ij}^k : 1 \leq k \leq q^4\}$, where the type τ switching involves vertices $v_1 \in V_a$, $v_3 \in V_b$, $v_4 \in V_c$, and $v_6 \in V_d$ for some $a, b, c, d \in [q]$, we define $\overline{f}_\tau^m$ as the value

$$\overline{f}_\tau^m = mM_{ia}M_{ib}M_{jc}M_{jd} \cdot 2^t, \tag{6.1a}$$

where each case where $a = i$, $b = i$, $c = j$, or $d = j$ contributes a value of 1 to the variable t, otherwise, t = 0.

Then for each $\tau \in \{D_{ij}^k : 1 \leq k \leq q^4\}$, we define the probability p_τ^m as

$$p_\tau^m = \frac{\overline{f}_\tau^m}{\sum_{k \in [q^4]} \overline{f}_{D_{ij}^k}^m}.$$

For the parameters in b-rejection, we define

$$B_{ij}^{(0)} = n_i d_i (d_i - 1) - 4\overline{D}_{ij} d_i - 8\overline{D}_{ii} d_i - 2\overline{L}_i d_i; \tag{6.2a}$$
$$B_{ij}^{(1)} = n_j d_j (d_j - 1) - 4\overline{D}_{ij} d_j - 8\overline{D}_{jj} d_j - 2\overline{L}_j d_j - 8 d_j \Delta - d_j^2 \Delta; \tag{6.2b}$$
$$B_{ij}^{(2)} = M_{ij} - 2\overline{D}_{ij} - 10\Delta - 2\Delta^2; \tag{6.2c}$$
$$B_{ij}^{(3)} = M_{ij} - 2\overline{D}_{ij} - 12\Delta - 2\Delta^2. \tag{6.2d}$$

Recall the definitions of $b_{12}^{(k)}(G', S)$ for $1 \leq k \leq 4$ in Section 5.3.4. Similarly, for some S used in NoDoubles$_{ij}$ and S produces G', we define $b_{ij}^{(k)}(G', S)$ for $1 \leq k \leq 4$ by replacing V_1, V_2 in the definition of $b_{12}^{(k)}(G', S)$ with V_i, V_j.

6.3.2 Parameters for NoDoubles$_i$

Suppose we are removing double edges from some graph G during the execution of NoDoubles$_i$, where G contains m double edges within V_i for some $0 < m \leq \overline{D}_{ii}$. We use D_{ii}^1 to $D_{ii}^{q^4}$ to represent all the types of switchings that are used to remove the double edges within V_i.

For each type $\tau \in \{D_{ii}^k : 1 \leq k \leq q^4\}$, $f_\tau(G)$ represents the number of type τ switching can be applied on G.

Given a specific type $\tau \in \{D_{ii}^k : 1 \leq k \leq q^4\}$, where the type τ switching involves vertices $v_1 \in V_a$, $v_3 \in V_b$, $v_4 \in V_c$, and $v_6 \in V_d$ for some $a, b, c, d \in [q]$, we define $\overline{f}_\tau^m$ as the value

$$\overline{f}_\tau^m = 2mM_{ia}M_{ib}M_{ic}M_{id} \cdot 2^t, \tag{6.3a}$$

where each case where $a = i$, $b = i$, $c = i$, or $d = i$ contributes a value of 1 to the variable t, otherwise, t = 0.

Then for each $\tau \in \{D_{ii}^k : 1 \leq k \leq q^4\}$, $f_\tau(G)$, we define the probability p_τ^m as

$$p_\tau^m = \frac{\overline{f}_\tau^m}{\sum_{k\in[q^4]} \overline{f}_{D_{ii}^k}^m}.$$

Then for the parameters in b-rejection, define

$$B_{ii}^{(0)} = n_i d_i(d_i - 1) - 8\overline{D}_{ii}d_i - 2\overline{L}_i d_i; \tag{6.4a}$$

$$B_{ii}^{(1)} = n_i d_i(d_i - 1) - 8\overline{D}_{ii}d_i - 2\overline{L}_i d_i - 9d_i^2 - d_i^3; \tag{6.4b}$$

$$B_{ii}^{(2)} = M_{ii} - 4\overline{D}_{ii} - \overline{L}_i - 10d_i - 2d_i\Delta; \tag{6.4c}$$

$$B_{ii}^{(3)} = M_{ii} - 4\overline{D}_{ii} - \overline{L}_i - 12d_i - 2d_i\Delta. \tag{6.4d}$$

For some S used in $\textsc{NoDoubles}_i$ and S produces G', we define $b_{ii}^{(k)}(G', S)$ for $1 \leq k \leq 4$ by replacing V_1, V_2 in the definition of $b_{12}^{(k)}(G', S)$ with V_i, V_i.

6.3.3 Parameters for NoLoops$_i$

Suppose we are removing loops from some graph G during the execution of $\textsc{NoLoops}_i$, where G contains m loops within V_i for some $0 < m \leq \overline{L}_i$. We use L_i^1 to $L_i^{q^2}$ to represent all the types of switchings that are used to remove the loops within V_i.

For each type $\tau \in \{L_i^k : 1 \leq k \leq q^2\}$, $f_\tau(G)$ represents the number of type τ switching can be applied on G.

Given a specific type $\tau \in \{L_i^k : 1 \leq k \leq q^2\}$, where the type τ switching involves vertices $v_1 \in V_a$, $v_3 \in V_b$ for some $a, b \in [q]$, we define $\overline{f}_\tau^m$ as the value

$$\overline{f}^m_\tau = mM_{ia}M_{ib} \cdot 2^t, \tag{6.5a}$$

where each case where $a = i$, $b = i$ contributes a value of 1 to the variable t, otherwise, t $= 0$.

Then for each $\tau \in \{L_i^k : 1 \le k \le q^2\}$, $f_\tau(G)$, we define the probability p^m_τ as

$$p^m_\tau = \frac{\overline{f}^m_\tau}{\sum_{k\in[q^2]} \overline{f}^m_{L_i^k}}.$$

Then for the parameters in b-rejection, define

$$B_i^{(0)} = n_i d_i(d_i - 1) - \overline{L}_i d_i(d_i - 1); \tag{6.6a}$$

$$B_i^{(1)} = M_{ii} - \overline{L}_i - 6d_i - 2d_i\Delta. \tag{6.6b}$$

Due to the variations between the structures for removing loops and those for removing double edges, we provide a detailed definition for $b_i^{(k)}(G', S)$ for $1 \le k \le 2$ in cases where S is used in $\textsc{NoLoops}_i$ and produces G'.

1. $b_i^{(1)}(G', S)$ as the number of distinct 3-tuples (w_1, w_2, w_3) where $w_2 \in V_i$, $w_1w_2, w_2w_3 \in E_S(G')$ and no loops at w_2.

2. $b_i^{(2)}(G', S)$ as the number of simple edges w_1w_2 in V_i such that $\{w_1, w_2, v_1, v_2, v_3\}$ are distinct vertices, and $w_1v_1, w_2v_3 \notin E(G)$.

6.4 Running time and proof of Theorem 3.3

In this section, we extend our analysis of running time for $q = 2$ to general q.

The running time analysis follows the same approach as in Lemma 5.10. Similarly to the case $q = 2$, there is no need to compute $\overline{f}^m_\tau$ for performing the f-rejection. For the b-rejection, the computation of $b(G', S)$ involves counting the number of 2-paths, or edges, satisfying certain adjacency constraints. Thus all these b functions can be computed in time $O(\Delta^2)$. Thus we obtain the following.

Lemma 6.1. *Provide that no rejection occurs, the expected running time of* JDM_GEN *is*

$$O\left(\mu_1 + \mu_2 + \mu_3\right),$$

where

$$\begin{aligned}
\mu_1 &= \sum_{i=1}^{q} n_i d_i; \\
\mu_2 &= \sum_{1\le i<j\le q} \left(\frac{M_{ij}^2 \Delta^2}{n_i n_j}\right); \\
\mu_3 &= \sum_{i=1}^{q} \left(\frac{M_{ii}\Delta^2}{n_i}\right).
\end{aligned}$$

Similarly, by observing that in each phase, $f_\tau(G) = (1 + O(\xi))\overline{f}_\tau^m$, and $b^k(G', S) = (1 + O(\xi))B^{(k)}$, for ξ given in the lemma below, we immediately obtain the following lemma.

Lemma 6.2. *The probability of any rejection occurring after the call of* PairingGen *in* JDM_GEN *is $O(\xi)$, where*

$$\xi = \sum_{1\le i<j\le 1} \frac{M_{ij}^2}{n_i n_j} \cdot \left(\sum_{m\in\{i,j\},n\in[q]} \frac{\Delta^2}{M_{mn}}\right) + \sum_{i\in[q]} \frac{M_{ii}^2}{n_i^2} \cdot \left(\sum_{m\in[q]} \frac{\Delta^2}{M_{im}}\right) + \sum_{i\in[q]} \frac{M_{ii}}{n_i} \cdot \left(\sum_{m\in[q]} \frac{\Delta^2}{M_{im}}\right).$$

Proof. The proof follows the same strategy as for Lemma 5.16, applied to each phase of the general case. □

Lemma 6.3. *Provided that*

$$\frac{M_{ij}^3 \Delta^2}{n_i^2 n_j^2} = o(1) \text{ for all } 1 \le i \le j \le 2, \tag{6.7}$$

the probability of any rejection occurring after the call of PairingGen *in* JDM_GEN *is $o(1)$.*

Proof. The proof idea is identical to Lemma 5.20. □

Lemma 6.4. *Provided that*

$$\frac{M_{ij}^3\Delta^2}{n_i^2 n_j^2} = o(1) \text{ for all } 1 \le i \le j \le 2, \tag{6.8}$$

under the assumption of no rejection occurs, the running time of all procedures after the call of PairingGen *in* JDM_GEN *is* $O\left(\sum_{i=1}^{q} n_i d_i\right)$.

Proof. The proof idea is identical to Lemma 5.21. □

Proof of Theorem 3.3. By the assumption of the theorem, the probability of any rejection occurring during JDM_GEN is $\frac{3}{8} + o(1)$ by Lemma 3.2 and Lemma 6.3. Then by Lemma 4.6 and Lemma 6.4, the running time of JDM_GEN is $O\left(\sum_{i=1}^{q} n_i d_i\right)$ where the time for the preparation of k_v, as discussed in Lemma 5.10, is also included.

Regarding uniformity, Lemma 4.3 guarantees the uniformity of G_0. Then the call of the subprocedure RelaxGraph in each phase, together with Lemma 5.5, ensures that the uniformity is preserved throughout the process. □

6.5 Proof of Theorem 3.5

Lemma 6.5. *Provided that* M_{ij} *are of the same asymptotic order* m *for all* $1 \le i \le j \le q$ *and*

$$\frac{m\Delta^2}{n_i n_j} = o(1) \text{ for all } 1 \le i \le j \le q,$$

the probability of any rejection occurring after the call of PairingGen *in* JDM_GEN *is* $o(1)$.

Proof. Under this assumption, the probability demonstrated in Lemma 6.2 can be simplified to $O(\xi)$ where

$$\xi = \sum_{1 \le i < j \le q} \frac{M_{ij}\Delta^2}{n_i n_j} + \sum_{i \in [q]} \frac{M_{ii}\Delta^2}{n_i^2} + \sum_{i \in [q]} \frac{\Delta^2}{n_i} = o(1) + \sum_{i \in [q]} \frac{\Delta^2}{n_i}.$$

The term $\frac{\Delta^2}{n_i}$ is the rejection probability of NoLoops_i for $i \in [q]$. Since we may assume that $\overline{L}_i = \frac{2q^2 M_{ii}}{n_i} > 1$, hence $\frac{M_{ii}}{n_i} = \Omega(1)$. Thus $\frac{\Delta^2}{n_i} \cdot \frac{M_{ii}}{n_i} = \frac{M_{ii}\Delta^2}{n_i^2} = o(1)$ implies that $\frac{\Delta^2}{n_i} = o(1)$.

Thus we have $o(1) + \sum_{i \in [q]} \frac{\Delta^2}{n_i} = o(1)$. □

Corollary 6.6. *Provided that M_{ij} are of the same asymptotic order m for all $1 \leq i \leq j \leq q$ and*

$$\frac{m\Delta^2}{n_i n_j} = o(1) \text{ for all } 1 \leq i \leq j \leq q,$$

$B_{ij}^{(k)} > 0$ for $0 \leq k \leq 3$ for all $1 \leq i \leq j \leq q$, and $B_i^{(k)} > 0$ for $0 \leq k \leq 1$.

Proof. The proof idea is identical to Lemma 5.13 where Lemma 6.5 fulfills the proof. $\square$

Lemma 6.7. *Let P be a uniformly random pairing in $\Phi(\boldsymbol{V}, M, \boldsymbol{d})$. Provided that M_{ij} are of the same asymptotic order m for all $1 \leq i \leq j \leq q$ and*

$$\frac{m\Delta^2}{n_i n_j} = o(1) \text{ for all } 1 \leq i \leq j \leq q,$$

then,

$$\Pr(P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d})) \geq \frac{3}{8} + o(1).$$

Proof. Using the idea from the proof for Lemma 3.2, it remains to show that the probability of $G(P)$ contains triple edges or loops of multiplicity at least two is $o(1)$

By Lemma 4.5, we need to show that the following hold:

(a) $O\left(\frac{M_{ij}^3}{n_i^2 n_j^2}\right) = o(1)$ for all $1 \leq i \leq j \leq q$;

(b) $O\left(\frac{M_{ii}^2}{n_i^3}\right) = o(1)$ for $i \in [q]$.

For (a), we have $\frac{M_{ij}^3}{n_i^2 n_j^2} \cdot \frac{n_i n_j \Delta^2}{M_{ij}^2} = \frac{M_{ij}\Delta^2}{n_i n_j}$. Since $M_{ij} \leq n_i d_i \leq n_i \Delta$, we have $\frac{M_{ij}^3}{n_i^2 n_j^2} = O\left(\frac{M_{ij}\Delta^2}{n_i n_j}\right) = o(1)$.

For (b), we have $\frac{M_{ii}^2}{n_i^3} \cdot \frac{n_i \Delta^2}{M_{ii}} = \frac{M_{ii}\Delta^2}{n_i^2}$. Since $M_{ii} \leq n_i d_i \leq n_i \Delta$, we have $\frac{M_{ii}^2}{n_i^3} = O\left(\frac{M_{ii}\Delta^2}{n_i^2}\right) = o(1)$. $\square$

Lemma 6.8. *Provided that M_{ij} are of the same asymptotic order m for all $1 \leq i \leq j \leq q$ and*

$$\frac{m\Delta^2}{n_i n_j} = o(1) \text{ for all } 1 \leq i \leq j \leq q,$$

under the assumption of no rejection occurs, the running time of all procedures after the call of PairingGen *in* JDM_GEN *is $O\left(\sum_{i=1}^{q} n_i d_i\right)$.*

Proof. By Lemma 6.1, it is equivalent to show the following

(a) $\sum_{1\leq i<j\leq q}\left(\frac{M_{ij}^2\Delta^2}{n_in_j}\right)=O\left(\sum_{i=1}^q n_id_i\right)$;

(b) $\sum_{i=1}^q\left(\frac{M_{ii}\Delta^2}{n_i}\right)=O\left(\sum_{i=1}^q n_id_i\right)$.

For (a), we have $\frac{M_{ij}^2\Delta^2}{n_in_j}=M_{ij}\cdot\frac{M_{ij}\Delta^2}{n_in_j}=o(M_{ij})=O(n_id_i)$ for all $1\leq i\leq j\leq q$. Hence, (a) holds.

For (b), we have $\frac{M_{ii}\Delta^2}{n_i}=n_i\cdot\frac{M_{ii}\Delta^2}{n_i^2}=o(n_i)=O(n_id_i)$ for $i\in[q]$. Hence, (b) holds. □

Proof of theorem 3.5. By the assumption of the theorem, the probability of any rejection occurring during JDM_GEN is $\frac{3}{8}+o(1)$ by Lemma 6.7 and Lemma 6.5. Then by Lemma 4.6 and Lemma 6.8, the running time of JDM_GEN is $O\left(\sum_{i=1}^q n_id_i\right)$ where the time for the preparation of k_v, as discussed in Lemma 5.10, is also included.

Regarding uniformity, Lemma 4.3 guarantees the uniformity of G_0. Then the call of the subprocedure RelaxGraph in each phase, together with Lemma 5.5, ensures that the uniformity is preserved throughout the process. Additionally, Corollary 6.6 ensures that the process of incremental relaxation is valid during the subprocedure RelaxGraph. □

Chapter 7

Future work

In this chapter, we discuss possible improvements that can be made to further optimize our results.

Currently, the condition $\frac{M_{ij}^3 \Delta^2}{n_i^2 n_j^2} = o(1)$ for $1 \le i \le j \le q$ is required in Theorem 3.3 to guarantee that the rejection probability of NoDoubles and NoLoops is $o(1)$. Let's revisit the $q = 2$ case and investigate possible improvements that could be made.

While establishing Lemma 5.16, we noticed that the probability of any rejections occurring during the execution of NoDoubles12 is mainly determined by f-rejections, where the probability of an f-rejection occurring is mainly determined by the count of forbidden cases involving non-edges, as discussed in Lemma 5.15.

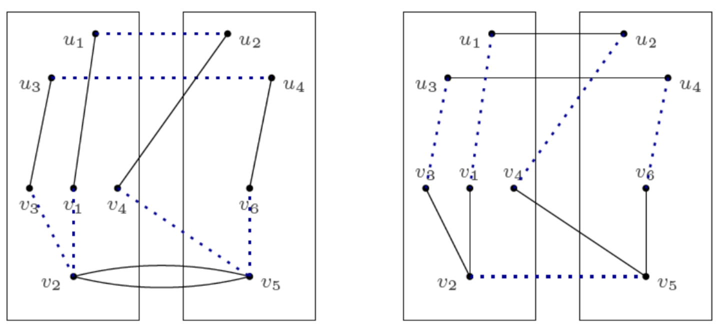


Figure 7.1: D_{12}^5 switching

Let's revisit the type D_{12}^5 switching as an example. As shown in Figure 7.1, the blue dotted lines indicate forbidden edges. Using our current approach to count forbidden cases that involve a non-edge being present, for instance, v_2v_3, we count the number of ways to choose all edges other than v_3u_3. This calculation gives us $O(mM_{11}^2M_{22}M_{12})$, where m represents the number of double edges between V_1 and V_2. We then find a neighbor for v_2, called v_3, and then a neighbor for v_3, named u_3. This gives $O(\Delta^2)$ possible choices, leading to a total of $O(mM_{11}^2M_{22}M_{12}\Delta^2)$ possible forbidden cases. This then gives the term $\frac{\Delta^2}{M_{11}}$ in the rejection probability.

However, when M_{11} is relatively small, this rejection probability can become significant. At the same time, with a small M_{11}, it becomes less likely for the edge v_2v_3 to be present. This suggests that the instances where v_2v_3 appears might be fewer than our current estimation of $O(mM_{11}^2M_{22}M_{12}\Delta^2)$. Hence, there's potential to further improve this estimate.

The method introduced by [10, **Lemma 9**] offers an approach to determine the probability of a specific edge being present, considering particular structures. In our scenario, if we can compute the probability of v_2v_3 being present, given that the tuple $(v_1, \ldots, v_6, u_1, \ldots, u_4)$ satisfies the requirements for performing a D_{12}^5 switching, then it might yield a more accurate estimate than what's currently available. However, the design of specific 'subsidiary switchings' is essential for computation. Given our problem's complexity, we might have to create various types of these switchings, which makes it more challenging.

References

[1] Georgios Amanatidis, Bradley Green, and Milena Mihail. Graphic realizations of joint-degree matrices. *CoRR*, abs/1509.07076, 2015.

[2] Georgios Amanatidis and Pieter Kleer. Rapid mixing of the switch markov chain for strongly stable degree sequences and 2-class joint degree matrices. In *Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 966–985.

[3] Andrii Arman, Pu Gao, and Nicholas Wormald. Fast uniform generation of random graphs with given degree sequences. *Random Structures & Algorithms*, 59(3):291–314, 2021.

[4] András Békéssy. Asymptotic enumeration of regular matrices. *Stud. Sci. Math. Hungar.*, 7:343–353, 1972.

[5] Edward A Bender and E Rodney Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3):296–307, 1978.

[6] Béla Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316, 1980.

[7] Colin Cooper, Martin Dyer, and Catherine Greenhill. Sampling regular graphs and a peer-to-peer network. *Combinatorics, Probability and Computing*, 16(4):557–593, 2007.

[8] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics.* Springer, Heidelberg; New York, fourth edition, 2010.

[9] Péter L Erdos, István Miklós, and Zoltán Toroczkai. A decomposition based proof for fast mixing of a markov chain over balanced realizations of a joint degree matrix. *SIAM Journal on Discrete Mathematics*, 29(1):481–499, 2015.

[10] Pu Gao and Nicholas Wormald. Uniform generation of random regular graphs. *SIAM Journal on Computing*, 46(4):1395–1427, 2017.

[11] Catherine Greenhill. The switch markov chain for sampling irregular graphs. In *Proceedings of the twenty-sixth annual acm-siam symposium on discrete algorithms*, pages 1564–1572. SIAM, 2014.

[12] Catherine Greenhill and Matteo Sfragara. The switch markov chain for sampling irregular graphs and digraphs. *Theoretical Computer Science*, 719:1–20, 2018.

[13] Mark Jerrum and Alistair Sinclair. Fast uniform generation of regular graphs. *Theoretical Computer Science*, 73(1):91–100, 1990.

[14] 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.

[15] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. Systematic topology analysis and generation using degree correlations. *ACM SIGCOMM Computer Communication Review*, 36(4):135–146, 2006.

[16] Brendan D McKay and Nicholas C Wormald. Uniform generation of random regular graphs of moderate degree. *Journal of Algorithms*, 11(1):52–67, 1990.

[17] Gottfried Tinhofer. On the generation of random graphs with given properties and known distribution. *Appl. Comput. Sci., Ber. Prakt. Inf*, 13:265–297, 1979.

APPENDICES

Appendix A

Proofs

A.1 Proof of 4.2

Proof. For any $G \in \mathcal{G}_0(\boldsymbol{V}, M, \boldsymbol{d})$, we prove the equation by giving bijection on sets $\mathcal{S}$ and $\mathcal{T}$. We first define these two sets.

For each vertex $v \in V(G)$, let C_v denote the cell corresponding to v in the pairing P associated with the graph G. We define N_v as the set containing all points in C_v, and $\mathcal{S}_{N_v}$ as the permutation group on N_v. The set $\mathcal{T}$ is defined as the Cartesian product over all $\mathcal{S}_{N_v}$, i.e., $\mathcal{T} = \prod_{v \in V(G)} \mathcal{S}_{N_v}$. Therefore, we have:

$$|\mathcal{T}| = \prod_{v \in V(G)} |\mathcal{S}_{N_v}| = \prod_{v \in V(G)} (\deg(v)!) = \prod_{i=1}^{q} (d_i!)^{n_i}.$$

Let $P_G = \{P : P \in \Phi_0(\boldsymbol{V}, M, \boldsymbol{d}), G(P) = G\}$.

For any $P \in P_G$, we define a colored pairing of P as follows:

1. For each pair in P corresponding to a simple edge in G, the edge is colored red.
2. For each pair in P corresponding to a loop in G, the pair is colored red, and the two endpoints of the pair are colored with different colors: one is colored red, and the other is colored blue. Thus, there are two possible ways to color a loop.

3. For every two pairs in P corresponding to a double edge in G, the two pairs are colored with different colors: one is colored red, and the other is colored blue. Hence, there are two possible ways to color these two pairs.

We define $Cl(P)$ as the set of all colored pairings derived from P. Let $\mathcal{S} := \bigcup_{P \in P_G} Cl(P)$, which represents the set of all colored pairings for the pairings in P_G.

Since the number of loops and double edges in G is given by $\boldsymbol{D}$ and $\boldsymbol{L}$, for any pairing P in P_G, we have $|Cl(P)| = 2^{\sum_{i=1}^{q} L_i + \sum_{1 \le i \le j \le q} D_{ij}}$. Additionally, $Cl(P) \cap Cl(P') = \emptyset$ for $P \neq P'$. Therefore, we have the following:

$$|\mathcal{S}| = |P_G| \cdot 2^{\sum_{i=1}^{q} L_i + \sum_{1 \le i \le j \le q} D_{ij}}.$$

We complete the proof by demonstrating a bijection between $\mathcal{S}$ and $\mathcal{T}$.

To facilitate this bijection, we label the edges of G. For a graph $G \in \mathcal{G}_0(\boldsymbol{V}, M, \boldsymbol{d})$, we label the edges in G arbitrarily with distinct labels from 1 to $m = \frac{|\boldsymbol{d}|_1}{2}$. Each edge is assigned a unique label, and every double edge is assigned two consecutive labels.

Here is an example of a labelled graph in Figure A.1:

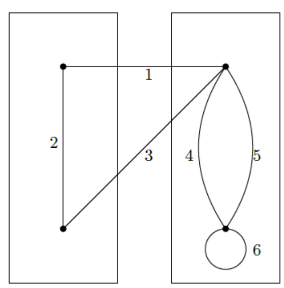


Figure A.1: labelled graph

Let G be the graph shown in Figure A.1. In Figure A.2, the right image represents a possible pairing $P \in P_G$, where the pairs in P are labelled based on the labelling in the graph G. The left image is a possible colored pairing belonging to $Cl(P)$.

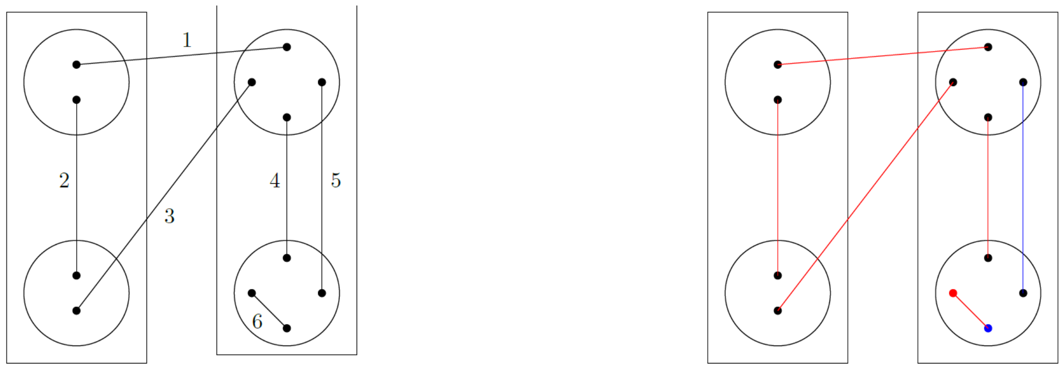


Figure A.2: coloured pairing

Note that the method of labeling a graph is not unique. However, for the purpose of our discussion, we assume that every graph $G \in \mathcal{G}_0(\boldsymbol{V}, M, \boldsymbol{d})$ has a predetermined labeling. This means that for each pair in a pairing, we can retrieve the label of the edge corresponding to that pair.

We define the function $f : \mathcal{S} \to \mathcal{T}$. Given a colored pairing from $\mathcal{S}$, its corresponding element in $\mathcal{T}$ is well-defined if we can determine the permutation σ for each set N_v, where $v \in V(G)$.

For any $v \in V(G)$, we define the permutation σ on N_v such that for any $i, j \in N_v$, $\sigma(i) < \sigma(j)$ if and only if $l_i < l_j$, where l_i is the label of the edge corresponding to the pair that contains i as one of its endpoints. Note that for pairs corresponding to a double edge (i.e., the pair labeled 4 and 5 in Figure A.2), we do not have specific rules on which pair corresponds to which edge. Therefore, the permutation order for the four points involved is not determined at this point.

In addition, for the pair (i, j) that corresponds to a loop in G, we have $\sigma(i) < \sigma(j)$ if and only if i is colored red and j is colored blue.

Furthermore, for the pair (i_1, j_1) and (i_2, j_2) corresponding to a double edge in G, where i_1 and i_2 are in the same cell, and j_1 and j_2 are in the same cell, we have $\sigma(i_1) < \sigma(i_2)$ and $\sigma(j_1) < \sigma(j_2)$ if and only if (i_1, j_1) is colored red and (i_2, j_2) is colored blue.

Using this approach, the permutation on N_v is well-defined, and consequently, f is well-defined. By applying a similar idea, we can obtain f^{-1}, which maps $\mathcal{T}$ back to $\mathcal{S}$. Therefore, f and f^{-1} together form a bijection between $\mathcal{S}$ and $\mathcal{T}$. $\square$