Generation of Random Graphs with prescribed degree sequences

Yujia Wu
Supervised by Professor Brendan McKay
Overview

PART ONE
Introduction

PART TWO
Motivation and Objectives

PART THREE
Algorithms

PART FOUR
Plans
Random graphs with given vertex degrees have recently attracted great interest as a model for many real-world complex networks, including the World Wide Web, peer-to-peer networks, social networks, and biological networks.

There are many fields in which the structure of random networks is important, but there are only a few cases in which either an exact theoretical model or an exact simulation is available.
Our project

Our project will attempt to fill in the gap in the case of random graphs whose vertices have known degrees. Throughout, we are concerned with generating simple graphs, i.e., no loops or multiple edges are allowed.

**Former researches and challenges:**

Most previously studied algorithms for this problem sometimes either get stuck or produce loops or multiple edges in the output, which is handled by starting over and trying again. Often for such algorithms, the probability of a restart being needed on a trial rapidly approaches 1 as the degree parameters grow, resulting in an enormous number of trials being needed on average to obtain a simple graph.
Objectives

- Prepare a survey of algorithms for generating random graphs with specified degrees;
- Implement and test several algorithms for exact sampling;
- Investigate Markov chains for inexact sampling;
- Apply the algorithms to estimate several parameters of practical importance.
Existing algorithms

- Algorithms based on the paring model
- Markov chain Monte Carlo algorithms based on the switching moves
- The sequential algorithm

A major advantage of the sequential algorithm is that it never gets stuck. This is achieved using the Erdős-Gallai characterization and a carefully chosen order of edge selection.
Sequential Algorithm

Sequential Algorithm For Random Graph with Given Degrees
Input: a graphical degree sequence (d1, ..., dn).
1. Let E be an empty list of edges.
2. If d = 0, terminate with output E.
3. Choose the least \( i \) with \( d_i \) a minimal positive entry.
4. Compute candidate list \( J = \{ j \neq i : \{ i, j \} \notin E \text{ and } i, j \text{ d is graphical} \} \).
5. Pick \( j \in J \) with probability proportional to its degree in \( d \).
6. Add the edge \( \{ i, j \} \) to \( E \) and update \( d \) to \( d_{i,j} \).
7. Repeat steps 4-6 until the degree of \( i \) is 0.
8. Return to step 2.
Output: \( E \).
My plans

Write C programs to achieve the generation of random graphs according to the sequential algorithm.

Consider about other properties of graphs, i.e., the spanning trees.

Test and seek for improvements of the programs and algorithms.

Compare the results of experiments with some theoretical conclusions, make summaries and write reports.
Thank you!

Main reference:
A SEQUENTIAL IMPORTANCE SAMPLING ALGORITHM FOR GENERATING RANDOM GRAPHS WITH PRESCRIBED DEGREES, By Joseph Blitzstein and Persi Diaconis, 2011