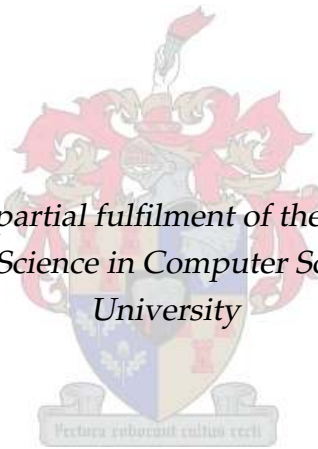# Modeling Online Social Networks using Quasi-clique Communities

by

Leendert W. Botha

*Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Computer Science at Stellenbosch University*
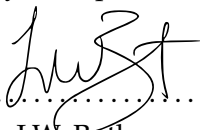
Department of Mathematics, Applied Mathematics and Computer Science,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Supervisor: Dr. R.S. Kroon

September 2011

# Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Signature: . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

LW. Botha

Date: . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

23 September 2011

i

# Abstract

With billions of current internet users interacting through social networks, the need has arisen to analyze the structure of these networks. Many authors have proposed random graph models for social networks in an attempt to understand and reproduce the dynamics that govern social network development.

This thesis proposes a random graph model that generates social networks using a community-based approach, in which users' affiliations to communities are explicitly modeled and then translated into a social network. Our approach explicitly models the tendency of communities to overlap, and also proposes a method for determining the probability of two users being connected based on their levels of commitment to the communities they both belong to. Previous community-based models do not incorporate community overlap, and assume mutual members of any community are automatically connected.

We provide a method for fitting our model to real-world social networks and demonstrate the effectiveness of our approach in reproducing real-world social network characteristics by investigating its fit on two data sets of current online social networks. The results verify that our proposed model is promising: it is the first community-based model that can accurately reproduce a variety of important social network characteristics, namely average separation, clustering, degree distribution, transitivity and network densification, simultaneously.

# Uittreksel

Met biljoene huidige internet-gebruikers wat deesdae met behulp van aanlyn sosiale netwerke kommunikeer, het die analise van hierdie netwerke in die navorsingsgemeenskap toegeneem. Navorsers het al verskeie toevalsgrafiekmodelle vir sosiale netwerke voorgestel in 'n poging om die dinamika van die ontwikkeling van dié netwerke beter te verstaan en te dupliseer.

In hierdie tesis word 'n nuwe toevalsgrafiekmodel vir sosiale netwerke voorgestel wat 'n gemeenskapsgebaseerde benadering volg, deurdat gebruikers se verbintenisse aan gemeenskappe eksplisiet gemodelleer word, en dié gemeenskapsmodel dan in 'n sosiale netwerk omskep word. Ons metode modelleer uitdruklik die geneigdheid van gemeenskappe om te oorvleuel, en verskaf 'n metode waardeur die waarskynlikheid van vriendskap tussen twee gebruikers bepaal kan word, op grond van hulle toewyding aan hulle wedersydse gemeenskappe. Vorige modelle inkorporeer nie gemeenskapsoorvleueling nie, en aanvaar ook dat alle lede van dieselfde gemeenskap vriende sal wees.

Ons verskaf 'n metode om ons model se parameters te pas op sosiale netwerk datastelle en vertoon die vermoë van ons model om eienskappe van sosiale netwerke te dupliseer. Die resultate van ons model lyk belowend: dit is die eerste gemeenskapsgebaseerde model wat gelyktydig 'n belangrike verskeidenheid van sosiale netwerk eienskappe, naamlik gemiddelde skeidingsafstand, samedromming, graadverdeling, transitiwiteit en netwerksverdigting, akkuraat kan weerspieël.

# Acknowledgements

I would like to express my sincere gratitude towards the following people:

- My supervisor, Dr. R.S. Kroon, for his guidance and commitment.
- MIH, for sponsoring the Media Lab and for providing us with valuable data sets for this study.
- My fellow lab colleagues, in particular Jacques Bruwer, Peter Hayward and Stephan Gouws for many hours of camaraderie.
- My parents, for their continuous support and for granting me the opportunity to study, and to do so at Stellenbosch University.
- Bennie and Rozanne, my siblings, for always providing the needed distractions outside the lab.
- My fiancé, Amy Becht, for her love, encouragement, support, and for supplying the needed commas in this thesis.

# Contents

# List of Figures

# List of Tables

# Nomenclature

**Abbreviations**

AS              Average separation

CC              Clustering coefficient

CN              Corporate network

DPL             Densification power law

ER              Erdős-Rényi

FF              Forest fire

FN              Friendship network

GL              Guillaume and Latapy

GLG             GL with grid-based initialization

MCMC            Markov chain Monte-Carlo

PA              Preferential attachment

PL              Power-law

**Variables**

$\beta$         Probability in our model that a new community node will enter the network in any given timestep.

$\delta$        Probability in our model that an existing user node will be connected to an existing community node in any given timestep.

| | |
|---|---|
| $\delta_{ik}$ | Commitment value of user $i$ to community $k$. |
| $\gamma$ | Probability in our model that a new user node will enter the network in any given timestep. |
| $\lambda$ | Overlap ratio in the GL model. |
| $\theta(c, c_k)$ | Overlap (number of mutual members) of communities $c$ and $c_k$ in our model. |
| $a_u$ | Activity value of a user node $u$ in our model. |
| $B$ | A bipartite network. |
| $C_{i,j,k}$ | The event that two user nodes, $i$ and $j$, in the social network are connected to each other based on their mutual affiliation to community $k$. |
| $e = (u, v)$ | Edge between nodes $u$ and $v$. |
| $E(G)$ | Set of edges/connections in the network $G$. |
| $f(\delta_{ik}, \delta_{jk})$ | Probability of occurrence of event $C_{i,j,k}$ |
| $G$ | A unipartite network. |
| $k_v^+$ | Out-degree of a node $v$ in a directed network. |
| $k_v^-$ | In-degree of a node $v$ in a directed network. |
| $k_v$ | Degree of a node $v$ in a undirected network. |
| $m$ | Number of edges/connections in a network. For a network $G$, $n = |E(G)|$. |
| $n$ | Number of nodes in a network. For a network $G$, $n = |V(G)|$. |
| $u, v$ | Nodes in a network. |
| $V(G)$ | Set of nodes of the network $G$. |

# Chapter 1

# Introduction

Online social networks are becoming increasingly popular, with the two biggest networks, Facebook [1] and Twitter [2], having a combined user base of almost a billion users.[1] As of July 2010, 70% of all Internet users have joined an online social network, making it the number one platform for creating and sharing content on the Internet [5]. Following this surge in popularity of online social networks, researchers have increased their focus on analyzing the structure of social networks. One possible way to gain insight into the dynamics of social network formation and evolution is to construct an accurate random graph model for modeling social networks, that generate structurally similar networks using a probabilistic process. Due to the privacy concerns that contribute to the scarcity of publicly available real-world social network data sets, such a random graph model can also be very valuable for generating artificial social network data sets.

## 1.1  Motivation

A social network is a structure made up of a set of entities, called nodes, which are connected to each other through some kind of interaction. These nodes can refer to individuals, groups, companies or even animals whereas the connections could represent friendship, collaboration, trade or communication, to name but a few. Social network analysis is used widely, with some application areas being primatology [6], sociology [7; 8], epidemiology [9; 10], economics [11; 12; 13], geography [14], information science [15; 16] and social psychology [17; 18].

---

[1]According to official press releases, Twitter had 200 million users (as of March 2011) [3] and Facebook 750 million (as of July 2011) [4].

In online social networks, the entities are typically individuals and the connections between them represent some form of personal relationship. The importance of understanding the structure and dynamics of these networks is immense. A recent study showed that 71% of people report a positive impression of a brand when interacting with it through their connections on a social network [5]. This is compared to 18% of people who report a positive impression after watching a television advertisement. The loyalty of people to others at a close social distance emphasizes the importance of understanding the way communities form, evolve and overlap in social networks. But it is not only in advertising that it pays to understand the structure of the networks. More and more online social networks are incorporating structural knowledge of the network into the design and functionality of the network. A recent example is the Google Plus [19] network, which is designed around social 'circles' or communities, requiring users to group their acquaintances into circles when creating a connection with them in the network. This may be seen as a direct effort to gain insight into the real-world communities that users are a part of.

During the *Social Network Analysis workshop* at the 2009 *Conference on Knowledge Discovery and Data Mining*, one of the biggest concerns expressed by the research community was the lack of benchmark data sets for social network analysis. The quality of existing data sets was also criticized due to incompleteness, sampling bias and the lack of evolutionary data. Due to privacy concerns, industry is poorly positioned to assist the research community in addressing these issues, and due to the complex structure of social networks, there exists no unbiased sampling technique that can be used to obtain samples of open networks, such as Twitter.[2] This study was completed in the MIH Media Lab[3], where we had access to two proprietary social network data sets to aid our research.

Random graph models offer a possible solution to the scarcity of data sets. A random graph model, if accurate, can randomly generate a collection of data sets with characteristics similar to those of current online social networks but without any privacy constraints. The processes used by the model to generate the networks would also provide valuable insight into the way real-world networks form. Many authors have presented random graph models for social network generation. These models have become increasingly accurate at modeling the various different characteristics of social networks. Recently, focus has started to shift towards a new

---

[2]Objectively defining "unbiasedness" for a social network sample is already a complex problem. Informally, an unbiased sample is defined as one that has the same "structure" as the original network. However, this structure can be defined according to a wide range of characteristics. This is further discussed in Section 2.2.7.

[3]MIH is short for *Myriad International Holdings*.

family of models, aimed at not only modeling the users and their connections in social networks, but also explicitly modeling the interactions between users and communities. None of the existing models, however, provide a realistic, intuitive way of modeling this behavior, making the naïve assumption that users will always be friends if they are affiliated to the same community.

The goal of this study was to create a random graph model that more accurately models this interaction between users and communities and to evaluate this model using the real-world data at our disposal.

## 1.2  Problem statement

To generate random social networks, we need a random graph model that accurately incorporates important characteristics of online social networks. The most commonly studied random graph model is the Erdős-Rényi (ER) model [20] which uses a fixed probability, $p$, of including any given connection in the network. The assumption that all connections are equally likely is very unrealistic in the case of social networks, however. In social networks, it has been found that the distribution of the degrees of nodes is highly skewed, with a small number of nodes having an unusually high degree [21]. Nodes in social networks also tend to cluster together: the amount of clustering in social networks is observed to be magnitudes larger than that present in networks generated by the ER model [22]. Because of these properties, and various others, traditional random graph models do not describe social networks accurately.

A large amount of work has been done to create a random graph model specifically for social networks. There are a number of desirable characteristics for such a model. Apart from accurately reproducing key social network characteristics, it is also desirable that a model be intuitive and mathematically tractable. In order to generate large networks, the model should also have minimal algorithmic complexity. Another important desideratum is the ability of the model to generate evolving, or dynamic, networks as opposed to static snapshots of the networks.

Most existing social network models use what we call a *bottom-up approach*, directly adding nodes and connections between them to a network in such a way that the network hopefully represent the structure of a social network. Although some of these models accurately reproduce some of the desired characteristics of social networks, we chose to use a *top-down approach*. With a top-down model, the affiliations of users to communities are explicitly modeled in a community struc-

ture which is then translated, or *flattened*, into a social network. The first advantage of this approach is that it is very intuitive, corresponding directly to real-world behavior where we meet our friends through the communities we belong to. A more important advantage, perhaps, is the extra level of information generated by the model. If a top-down model could generate accurate evolutionary social network data sets, the information provided by the community structure could be just as valuable; providing insight into how communities form, evolve and interact. However, the current state of the art in top-down models do not accurately model real-world networks. This is due to the deterministic flattening rule used to translate the community structure into a social network. All of the current models assume that each community in the community structure will result in a clique over its members in the final social network. In this study, we propose to study a dynamic top-down model that uses a probabilistic flattening rule, allowing for variable connection density within communities in the social network. We are not aware of any other existing models using this approach.

## 1.3   Objectives

The following are the objectives of this study:

- Identifying a set of key characteristics that distinguish social networks from random networks.
- Developing a top-down dynamic social network model that uses a probabilistic flattening rule.
- Fitting our model on two current online social network evolutionary data sets, and comparing its performance to that of existing models.

## 1.4   Data sets

In this study, we base our evaluation of our model, and the existing models we compare to, on the full evolutionary patterns of the networks, not just characteristics of the fully evolved networks. Very few studies to date have attempted to analyze the evolution of social networks, with the focus usually on static characteristics. Through our relationship with MIH we have obtained two data sets from current online social networks, both of which include complete historical records

for the evolution of the network[4].

Our first temporal data set is from a proprietary corporate social network owned by a multi-national holding company. It is a closed network in which employees can connect with colleagues in other companies owned by the parent company. Although the network is small (1265 nodes), it is a mature network, having being adopted by most of the individual companies since its launch in 2008. We refer to this network as the *Corporate Network* (CN).

The second network is a South African social network attracting young people through a local presence in entertainment venues. The data set contains 13 295 nodes and 40 679 connections between them. We refer to this network as the *Friendship Network* (FN).

## 1.5   Thesis outline

**Chapter 2** introduces basic network methodology and key network characteristics. A review of available literature analyzing these characteristics on social networks is presented.

**Chapter 3** reviews the development of random graph models of social networks.

**Chapter 4** proposes our community-based simulation model and discusses its relationships with existing models.

**Chapter 5** describes a method for searching the parameter space of our model for suitable parameters for modeling a given network.

**Chapter 6** gives a more technical description of the proprietary data sets that are used in the study. It presents the empirical results obtained from our simulations and compares them to those of state-of-the-art models.

**Chapter 7** summarizes our findings and describes possible extensions to this work.

---

[4]These historical records do not include information for users or connections that have been removed from the network.

Chapter 2

# Social network terminology and characteristics

This chapter introduces key graph theory concepts relevant to social network analysis. Section 2.1 gives an overview of important terminology that will be used throughout this study. In Section 2.2, a review of a number of distinctive characteristics of social networks is given.

## 2.1   Network terminology

Throughout this study, we will make extensive use of graph theory concepts. We give a brief introduction to these concepts below.

### 2.1.1   Fundamental concepts

A *graph* or *network G* consists of a non-empty set $V(G)$ of entities, called *nodes*, and a set $E(G)$ of connections between them, called *edges*. In the context of this study, nodes will represent individuals or communities in a social network and edges will represent social interaction between these individuals and/or communities. Graphically, nodes can be depicted as points in the plane and edges as lines between these points. Each edge $e = (u, v)$ consists of a pair of nodes, $u$ and $v$, and is said to be *incident* on $u$ and $v$. In this study, we assume each edge in $E(G)$ is unique and that no edge $(u, v)$ connects a node to itself, that is $u \neq v$. If the order of the nodes in each edge $e = (u, v)$ is relevant, then the edges are called *directed edges* and $e$ is said to be from $u$ to $v$. A graph whose edges are directed is called a

*directed graph*. If the order is not relevant, $G$ is said to be an *undirected graph*. In an undirected graph, the number of edges incident on a node $u$ is called the *degree $k_u$* of node $u$. If the degree of $u$ is zero, then $u$ is called an *isolated node*. In a directed graph, the number of edges to a node $u$ is called its *in-degree $k_u^-$* and the number of edges from node $u$ is called its *out-degree $k_u^+$*.

We use the convention that the graph $G$ has $n_G$ nodes and $m_G$ edges, that is

$$|V(G)| = n_G \quad \text{and} \quad |E(G)| = m_G$$

Such a network is said to be of *order $n_G$* and *size $m_G$*. When the context makes it clear which graph we are referring to, we usually omit these subscripts. A network is called *complete* if it contains all possible edges over the node set. A complete undirected network with $n$ nodes has $\frac{n(n-1)}{2}$ edges, so that $m$ is $O(n^2)$. If $m$ is close to this upper bound for a network $G$, then $G$ is said to be a *dense* network. On the other hand, if $m$ is of the same order of magnitude as $n$, $G$ is said to be a *sparse* network.

A network $H$ is called a *subgraph* of a network $G$ if $V(H) \subset V(G)$ and $E(H)$ is a subset of $E(G)$, restricted to edges between nodes in $V(H)$, i.e.

$$E(H) \subset \{(u,v) \in E(G) : u,v \in V(H)\} \,.$$

A *path* of length $l$ from $u$ to $v$ in a graph is a sequence of $l$ consecutive edges, $(u, u_1)$, $(u_1, u_2)$, $\cdots$, $(u_{l-1}, v)$.[1] If there is a path from $u$ to $v$, $v$ is said to be *reachable* from $u$, or *connected* to $u$. For undirected graphs, reachability is an equivalence relation over the set of nodes which partitions the nodes into equivalence classes called *connected components*.

A graph $B$ is called *bipartite* if its nodes can be partitioned into two disjoint subsets $A_1$ and $A_2$ such that each edge connects a node in $A_1$ to one in $A_2$. In this study, the set $A_1$ will typically contain community nodes, and the set $A_2$ will typically contain user nodes.

In the rest of this study, we prefer to use the term *connection* instead of *edge* and *network* instead of *graph*, since these directly correspond to the real-world entities we are investigating.

---

[1] $l$ may equal one, in which case the path consists of the sequence: $\{(u,v)\}$. There is also a path of length zero from every node to itself.

### 2.1.2   Cliques and quasi-cliques

In psychology, the term *clique* refers to an inclusive group of people. Such cliques are often the primary source of social interaction for its members [23] and are, therefore, extensively studied in social psychology. In graph theory, a *clique* refers to a complete subgraph. Such a fully connected network of order $k$ is commonly referred to as a *k-clique*. Examples of $k$-cliques for various $k$ are shown in Figure 2.1. A clique is called a *maximal clique* if it does not form a subgraph of any other clique.



(a) $k = 2$            (b) $k = 3$            (c) $k = 6$
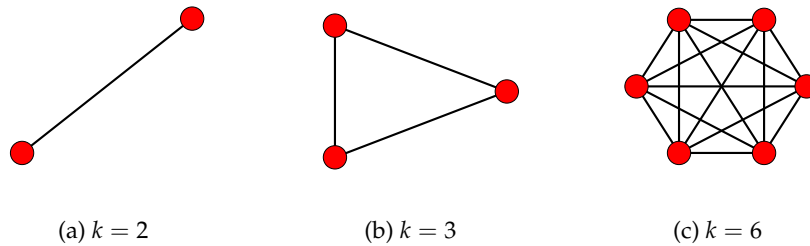
Figure 2.1: $k$-clique structures for various values of $k$

Cliques play an important role in social network analysis and many methods for extracting cliques from networks [24; 25] and building networks from cliques [26; 27] have been proposed. Generally, finding the size of the largest clique in a network is NP-complete [28].

An important objective of social network analysis is to infer information about real-world communities (social circles, family, school, work colleagues, sport clubs, etc.) through connections in a social network. Such real-world communities are often present in social networks as cliques. In many cases though, some nodes within the community will not be connected. This is the result of the inactivity of some people on online social networks as well as human social behavior. As the size of a real-world community increases, it becomes more likely that at least one pair of people in the community will not befriend each other. For these reasons, we feel that real-world communities are more accurately modeled in social networks as dense subgraphs or *quasi-cliques*. This definition of a quasi-clique as a dense subgraph has been used by various authors in the community detection literature [29; 30].

### 2.1.3   $k$-stars and $k$-triangles

A network of order $(k + 1)$ is called a $k$-star if it has size $k$ and there is some node $i$ that is connected to all $k$ other nodes. Figure 2.2 shows $k$-stars for three different values of $k$.



(a) $k = 2$          (b) $k = 3$          (c) $k = 5$
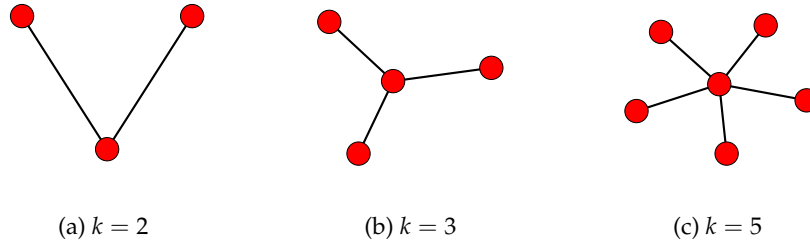
Figure 2.2: $k$-stars for various values of $k$.

A 3-clique is commonly referred to as a *triangle* and a *k-triangle* is a set of $k$ triangles all sharing an edge. Figure 2.3 shows the structure of 1-, 3- and 5- triangles. The simplest of these, the normal triangle ($k = 1$), is by far the most studied in social network analysis. The 1-triangle plays an important role in social networks and is seen by many as the building block of social networks [31].

## 2.2   Social network characteristics

There are a number of characteristics that clearly distinguish social networks from random networks. [2] We discuss the most prominent characteristics below.

---

[2]One example of such a 'random network' is the ER model, discussed in Section 3.1.1.



(a) $k = 1$          (b) $k = 3$          (c) $k = 5$
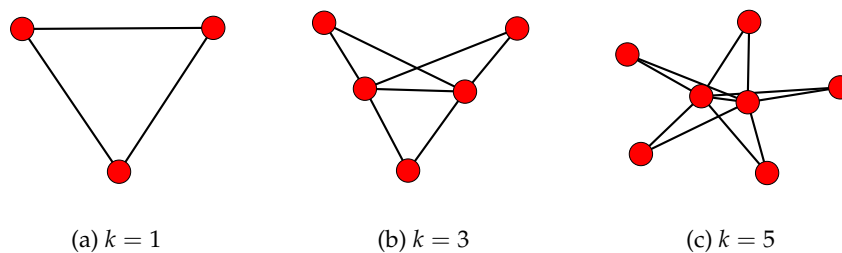
Figure 2.3: $k$-triangles for various values of $k$

### 2.2.1   Small world phenomenon

*Small-world* networks are networks with a small *average separation*, i.e. a small average distance between random pairs of nodes in the network. Kochen and Pool [32] began investigating the small world problem in the early 1950s. Motivated by his interaction with Kochen and Pool, social psychologist Stanley Milgram designed an experiment to measure the average degree of separation between people in the United States.  He gave letters to random subjects who each were instructed to pass the letter on to an acquaintance who they thought would be the most likely to know the addressee. He found the average number of people required for the letter to reach its destination to be only about six [18], which sparked the social phrase "Six Degrees of Separation".[3]

The small-world phenomenon is also observed in online social networks where a relatively short path can be found between any pair of nodes, even in large networks.  This has been confirmed by a number of independent studies [22; 34; 35], including a recent study of the Microsoft Messenger Instant-Messaging System performed by Leskovec and Horvitz [36], in which they found the average separation to be 6.6 in a social network containing 180 million nodes.  This is in contrast to random networks, where the average path length is much longer.[4]

### 2.2.2   Shrinking diameter

The *diameter $D(G)$* of a network is the maximal shortest path length between two nodes in the network.  Because of the small average separation present in social networks, the diameter is typically smaller than in a random network of the same order and size.[5]

Barabási, Albert and Jeong [39] first observed through experimentation that in social networks, $D(G)$ increases very slowly, typically as a logarithmic function of $n$. This result was confirmed by Newman et al. using heuristic methods [40].  More recently, Leskovec et al. [41] studied some major online social networks using a more robust measure called the *effective diameter*, which is not easily influenced by degenerate structures in the network, like chains of nodes. The effective diameter is the minimum path length within which some quantile $q$ of the pairs of nodes

---

[3]This phrase was further popularized by John Guare's play of the same title [33].

[4]For a comprehensive analysis of average path lengths in random graphs, see the work of Fronczak et al. [37].

[5]For a detailed discussion and analytical analysis of the diameter of random networks, refer to the work of Chung et al. [38].

can reach each other.[6]  They were surprised to find the effective diameters of the networks to slowly decrease with network size. They referred to this phenomenon as the *shrinking diameter*.

### 2.2.3  Clustering coefficient

A common property of social networks is that highly connected clusters occur in the networks. These clusters, also called *quasi-cliques*, consist of groups of densely interconnected nodes. We refer to these highly connected clusters as *communities*, and they often have real-world parallels in that many people from the same social circle such as a family, school, company or sport club will befriend each other.

In 1998, Watts and Strogatz [22] introduced the *clustering coefficient* (CC) as a measure of the degree of clustering in a network. For a given node $i$, with degree $k_i > 1$, the CC is defined to be the ratio of the number of connections that exist between node $i$'s neighbors and the total number of potential connections that could exist between them; when $k_i \leq 1$, the CC of the node is defined to be zero. If $E_i$ is the number of connections that actually exist between the $k_i$ neighbors of node $i$, the CC of the network is given by

$$CC(G) = \frac{1}{n_G} \sum_{i:k_i > 1} \frac{2E_i}{k_i(k_i - 1)} \ ,$$

the average of all the nodes' CCs. Note that we can view the CC as a function of $n$ when interested in the evolution of this measure as a network grows. Since the CC is defined to be 0 for isolated nodes and nodes with only one connection, using only the *giant component*[7] for analysis will result in an over-estimation of the CC.

In social networks, the CC is usually several orders of magnitude greater than in random networks, an observation first made by Watts and Strogatz [22]. Mislove et al. [43] recently estimated the CC on the major online social networks Flickr [44], LiveJournal [45], Orkut [46] and YouTube [47]. They found Flickr to be the most clustered network, with a CC of 0.31, which is 47200 times the expected CC of an ER network of the same order and size. They found YouTube to be the least clustered network, with a CC of 0.137, which is 36900 times the expected CC of an ER network of the same order and size.

---

[6]In their studies, Leskovec et al. use $q = 0.9$.

[7]Most social networks have a connected component comprising a high proportion of the nodes, known as the giant component [42].

### 2.2.4  Transitivity

Transitivity reflects a propensity for the formation of triangles in social networks. This is often quantified by the probability that a randomly chosen pair of neighbors of a node are connected. This probability is determined by the number of 2-stars and the number of triangles in the network [48]:

$$T = \frac{3 \text{ x } \text{ (number of triangles)}}{\text{(number of 2-stars)}} \; .$$

The factor of three is used for normalization, since there are three 2-stars in every triangle. The value of $T$ is thus an estimate of the probability of closure in a 2-star. This probability is orders of magnitude greater in social networks than in random networks and corresponds with the higher CC observed in social networks [48].

A more detailed measure of transitivity is the *node triangle participation* distribution over $\psi$, that gives the proportion of nodes that form a part of $\psi$ triangles. This distribution is typically long-tailed for social networks.

### 2.2.5  Degree distribution

The degree distribution of a network is a distribution function $P(k)$ that gives the probability that a randomly selected node in the network has degree $k$. In a purely random ER network, the degree distribution is binomial, so that the vast majority of the nodes have degree close to the mean degree. In the limit of large $n$, the binomial distribution can be approximated by the Poisson distribution. However, empirical results [34] show that for social networks, the degree distribution has a heavier tail which approximately follows a *truncated power-law* (PL) of the form[8]:

$$P(k) \propto k^{-\alpha} \quad \text{for } k > k_{min}.$$

Figure 2.4 shows the difference between the Poisson distribution and power-law distribution.

### 2.2.6  Network densification

Twentieth century literature on the evolution of real-world social networks implicitly assumes that the number of connections scales roughly linearly with the num-

---

[8]We require $k_{min} > 0$, otherwise the distribution diverges [49]. When $k_{min} = 1$, the distribution is referred to as a *power law*. In the context of social network analysis the term power-law distribution is used more loosely, referring to distributions that have a power-law tail [49]. To preserve consistency with existing literature, we will refer to distributions with power-law tails as power-law (PL) distributions, even though they might technically be truncated power-law distributions.

(a) Poisson distribution          (b) Power law distribution
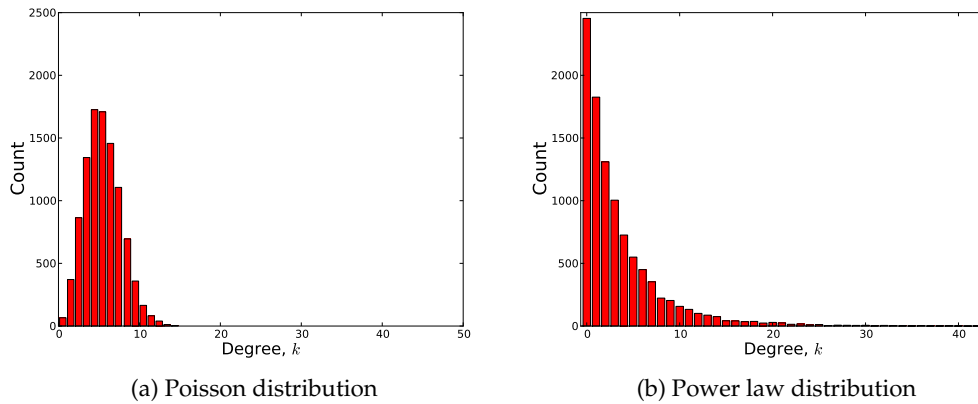
Figure 2.4: The degree distributions of two networks, one following a Poisson distribution and the other a power-law distribution. Even though both networks have the same average degree, the maximum degree in the network with a power-law degree distribution is three times higher.

ber of nodes and, therefore, the average degree is approximately constant. In 2000, Dorogovtsev and Mendes [50] were the first to note that the number of connections in real-world networks increases at a faster rate than the number of nodes. They incorporated this in their *accelerated growth model* (discussed in Section 3.1.4.2). Leskovec et al. [41] recently confirmed this result by observing that on many major online social networks, the number of connections grows superlinearly in the number of nodes, i.e:

$$m_G \propto n_G^\rho$$

for some densification exponent $\rho > 1$. This phenomenon is commonly referred to as the *densification power law* (DPL) with exponent $\rho$.[9]

### 2.2.7  Problems with sampling from social networks

Due to the complex nature of social networks, no standard sampling technique seems to simultaneously preserve all the properties described in the previous sections [51; 43]. The most used sampling technique, *snowball sampling* (also referred to as 'crawling' a network), is often the only option for online social networks since researchers are limited by the functionality provided by the programming interfaces of the social networks. Snowball sampling starts from a set of pre-selected nodes and follows connections from these nodes, recursively adding all nodes and

---

[9]Note the ambiguity of the term 'power law': here it does not refer to the power-law distribution.

connections it encounters to the sampled network. Due to their highly connected nature, dense communities are over-sampled, producing connected networks with significantly higher CCs and shorter average path lengths than the original networks [51]. Also, this method is extremely likely to only sample from the giant component of the network and gives no indication of how many other connected components or isolated nodes there are in the network.

Leskovec et al. [52] presented a thorough analysis of the most used sampling techniques and introduced a new method, called *forest fire* (FF) sampling, which is based on their work in temporal network analysis (see Section 3.1.10). The FF sampling method eliminates the bias towards higher degree nodes, but the authors note that no sampling technique succeeds in preserving all of the desired properties of social networks and the choice of algorithm should be made based upon which properties are the most important to preserve.

These restrictions imposed by sampling from existing social networks emphasize the importance of an accurate model for social networks, which can be used to generate smaller data sets that exhibit social network characteristics without the need to sample from large networks.

# Chapter 3

# Existing models of social networks

In graph theory, a *random graph* is a graph that is generated through some probabilistic process. The theory of random graphs was pioneered in the late 1950s by Paul Erdős and Alfréd Rényi [20], when Erdős started applying probabilistic methods to graph theory problems. Despite the widespread use of their basic model in a variety of other fields, it has been shown that it does not capture any of the important characteristics of social networks [34; 22]. Many different approaches have been proposed to find a model that can accurately generate social networks. In this chapter, we present the most important of these models. All of them aim at one or more of the following:

- Capturing some or all of the key characteristics of social networks presented in Section 2.2.
- Building the network in a realistic and intuitive way that corresponds to how real-world networks form.
- Providing mathematical tractability as a base for analytical analysis of the model.
- Minimizing algorithmic complexity, enabling the model to quickly generate large networks as data sets.

All of the models presented in Section 3.1 use a *bottom-up approach*, adding nodes and connections at the *microscopic* level in a certain way in order to mimic social network structure on a *macroscopic* level. One important characteristic of this macroscopic structure is the potentially complex way in which communities evolve and overlap.

**15**

Section 3.2 presents a promising new class of models that uses a *top-down approach*, making use of a bipartite, or two-level, structure. These models first model the affiliations of users to communities in a bipartite network containing both user and community nodes. This bipartite network is then transformed into a social network containing just user nodes. This new approach is aimed at intuitively reproducing real-world behavior where people interact through social circles. In the social sciences, this behavior has been studied as far back as Breiger's study in 1973 of the the affiliation of people to groups  [53]. The importance of community modeling in social networks is becoming more and more evident, with most online networks now trying to elicit and make use of some form of community information from users. Perhaps the most prominent example is the recent launch of the Google Plus network [19], where the entire user interface is based on the top-down approach, requiring users to group their acquaintances into social 'circles' when creating a connection with them in the network.

## 3.1  Bottom-up models

The vast majority of models in the literature use a bottom-up approach. These models build networks from a microscopic perspective, focusing on how nodes and connections should be formed in the network so that the global structure represents that of a social network. This global structure is characterized by the measurements presented in Section 2.2.

In this section, we present the development of the major bottom-up models, roughly chronologically. The first of these models, the basic ER model, is presented in Section 3.1.1. The WS model, the first model to produce networks exhibiting small-world behavior, is presented in Section 3.1.2. In Section 3.1.3, we discuss one of the most prominent models in the literature, the PA model. Not only was it the first model to produce dynamic networks, but it was also the first to produce scale-free networks. Many authors proposed variations of the PA model, and we discuss five of these PA-based models in Section 3.1.4. An interesting generalization of the ER model aimed at keeping the analytical simplicity of the model, but allowing the formation of arbitrary degree distributions, is discussed in Section 3.1.5.

The prominent high level of transitivity present in social networks has led many authors to propose models that generate networks using a process that explicitly includes transitivity. We present two of these models, Ebel's transitive model (Section 3.1.6) and Newman's transitive model (Section 3.1.7). Ebel's model was also one of the first to model not only the addition of nodes to the network, but also the

removal of nodes, a direction of study that is still impaired by the lack of available supporting data sets.

Another important family of models, exponential random graph models, is presented in Section 3.1.8. These models explicitly define a probability distribution over all possible graphs, in order to assess how likely it is to observe a given graph. The formulation of such a probability distribution is a delicate process since underfitting, over-fitting and computational complexity are factors to consider. We discuss various formulations of the probability distribution.

Lastly, in Sections 3.1.9 and 3.1.10, we include two models proposed by Leskovec et al., based on a recent study of current online social networks. These models were the first models to exhibit shrinking diameters and network densification.

### 3.1.1 The Erdős-Rényi model

The most commonly studied random graph model is the ER model proposed by the Hungarian mathematicians Erdős and Rényi [20] in 1959.[1] This model is often referred to as the $G(n,p)$ model, where $n$ is the number of nodes in the network and $p$, the density parameter, is the probability of a connection between any pair of nodes. Each node is thus connected to any of the other $(n-1)$ nodes in the network with independent probability $p$. It follows that the degree of any node is binomially distributed,

$$P(k) = \left( \begin{array}{c} n\text{ - }1 \\ k \end{array} \right) p^k (1-p)^{n-1-k} , \tag{3.1}$$

and that the expected clustering coefficient is $p$.

A variant of this model is the $G(n,m)$ model, where $n$ is the number of nodes in the network, and $m$ is the number of connections in the network. In this case, the resulting network is sampled uniformly at random from the collection of all networks with $n$ nodes and $m$ connections. The distribution of graphs under the $G(n,m)$ model is identical to that of the $G(n,p)$ model, for $p = \frac{m}{\binom{n}{2}}$, conditioned on the number of edges in the graph being $m$.

To estimate the value of $p$ for generating a network with a desired number of vertices and connections, one can use maximum likelihood (ML) estimation. Since connections in the network generated by the ER model appear independently with

---

[1]It is worth noting that Erdős and Rényi's work long preceded social network analysis and although their work is frequently cited in social network literature, their model was never intended to preserve social network characteristics.

probability $p$, the likelihood of a network with $n$ nodes and $m$ connections is

$$p^m(1-p)^{n(n-1)/2-m} \; ,$$

thus the ML estimate of $p$ minimizes the negative log-likelihood

$$-m\log p - [n(n-1)/2 - m]\log(1-p).$$

Differentiating and setting to zero yields

$$[n(n-1)/2 - m]/(1-p) = m/p$$

so that

$$pn(n-1)/2 - pm = (1-p)mk$$

$$\Rightarrow p = 2m/n(n-1). \tag{3.2}$$

Thus, the maximum likelihood estimate of $p$ is the ratio of the actual number of connections in the network to the maximum possible number of connections.

### 3.1.1.1   Directed Erdős-Rényi model

Gui and Dutton [54] proposed an extension to the ER model that generates directed networks, given a desired out-degree distribution $\mathcal{D}$. To construct a random directed network $G$ with $n$ nodes, each node $v$ independently chooses its out-degree, $k^+(v)$, according to $\mathcal{D}$ and then randomly chooses a subset of $k^+(v)$ nodes to assign the outgoing connections to. The authors show that when the expected value of $\mathcal{D}$ is finite, the distribution of the in-degrees approaches a Poisson distribution as $n \to \infty$.

### 3.1.2   The Watts and Strogatz model

The Watts and Strogatz (WS) model [22], published in 1998 and also known as the *Watts beta model*, was the first model designed to generate networks which exhibit small world properties, i.e. which have short average path lengths and high clustering (discussed in Section 2.2.1). The model takes as input the number of nodes $n$, the mean degree $k \leq n-1$ (assumed to be an even integer) and a parameter $\beta$ ($0 \leq \beta \leq 1$). It constructs a network in the following way:

1. A network with a ring lattice is constructed. This is a network with $n$ nodes each connected to $k$ neighbors, $\frac{k}{2}$ on each side.

Figure 3.1: The initial configuration for the Watts and Strogatz model with $n = 20$ and $k = 4$ (left); the resulting network for some $0 < \beta < 1$ (middle); and the resulting network for $\beta \approx 1$ (right). Diagram reproduced from [22].

2. For every node $n_i$, the connection between $n_i$ and every $n_j$ on the counter-clockwise side of the lattice from $n_i$ is 'rewired' with probability $\beta$. Rewiring is done by replacing the connection between $n_i$ and $n_j$ with a connection be-tween $n_i$ and $n_l$ where $l$ is chosen randomly from all values that avoid self-loops and duplication of connections.

An initial configuration and two resulting networks are shown in Figure 3.1.

The WS model has two shortcomings, the first of which is the assumption that the network contains a fixed number of nodes. In contrast, most real-world networks form dynamically by the continuous addition of nodes to the network. The second shortcoming is that all the nodes have approximately the same degree. For $0 < \beta < 1$, the degree distribution has a pronounced peak around the mean, similar to the ER model, and in the limiting case of $\beta \to 1$ the degree distribution becomes a Poisson distribution, meaning the generated networks are not scale-free [55].

### 3.1.2.1 Newmann and Watts' improved model

In 1999, Newmann and Watts [56; 57] proposed a variant to the original WS model which is easier to analyze since it does not lead to the formation of isolated clus-ters as the original model sometimes does. In this model, connections are added between random pairs of nodes, but no connections are removed from the original lattice. Although their model offers analytical simplicity, it is still subject to the same shortcomings as the original WS model.

### 3.1.3   The preferential attachment model

In 1999, Barabási and Albert [34] addressed the two shortcomings of the WS model by creating the first dynamic model for small-world, scale-free networks. They incorporated the principal of *proportional selection*, where some nodes are more likely to form connections than others.  Proportional selection of an object relative to a characteristic $c$ over a set $\mathcal{V}$ of objects means that the probability of object $v_i$ being chosen is given by

$$P(v_i) = \frac{c(v_i)}{\sum_{v_j \in \mathcal{V}} c(v_j)}$$

Proportional selection is often described by the catchphrase *the rich get richer*, a concept first applied to the growth of networks by de Solla Price in 1976  [58]. In Barabási and Albert's model, the nodes are chosen with proportional selection relative to their degree.  Thus, the probability that the new node is connected to node $i$ is

$$P(k_i) = \frac{k_i}{\sum_{j=1}^{n} k_j} \, , \tag{3.3}$$

where $k_i$ is the degree of node $i$. This special case of proportional selection is commonly referred to as *preferential attachment* (PA), and we refer to Barabási and Albert's model as the PA model.

The algorithm used for generating an undirected network using the PA model with parameter $m_0$ is presented below:

1. A random initial network with $n_0 > \max\{m_0, 2\}$ nodes are created.  There are several ways to generate the initial network, and all of them lead to the same asymptotic behavior [59].

2. New nodes are added to the network one at a time.  When a new node is inserted, it is assigned connections to $m_0$ other nodes using preferential attachment, disallowing self-loops and duplicate connections. This means that new nodes prefer to form connections to highly connected nodes.

There are a couple of drawbacks to the PA model:

- The model provides little flexibility, with the only controllable characteristic being the average degree.  Experimental results indicate that the degree distribution resulting from this model is a power-law with parameter $\alpha = 2.9 \pm 0.1$ [59]. Most real-world networks have degree distributions with heavier tails, but this behavior can not be reproduced using the PA model.

- Networks generated using the PA model have no nodes with degree $k < m_0$, and in particular no isolated nodes.
- The clustering coefficient decreases strongly as the network size increases, which contradicts observations on real-world networks [60].
- Because of the lower bound on the minimum degree, the PA model is extremely unlikely to produce networks that contain long paths.
- In the PA model, there is a strong positive correlation between the age of a node and the degree of the node. This kind of correlation is not observed in real-world networks [61].

The next section presents variants of the original PA model that aim at eliminating some of these drawbacks.

### 3.1.4   Variants of the PA model

#### 3.1.4.1   Kumar's copy model

Kumar et al. [62] proposed a directed model for modeling the structure of the world-wide web that implicitly employs PA through a *copying mechanism* by which new nodes that enter the network copy a subset of outgoing connections from existing nodes. When a node is added to the network, a *prototype node* (corresponding to the close friend) is chosen at random. With probability $(1 - p)$, the $i$-th connection is taken to be the prototype's $i$-th connection, otherwise a node is chosen at random to connect to. Barabási and Albert noted that the copying mechanism effectively amounts to using preferential attachment [42].

#### 3.1.4.2   Directed models by Dorogovtsev, Mendes and Samukhin

Dogorovtsev, Mendes and Samukhin have proposed many variants of the PA model, for generating scale-free directed networks, in an attempt to model the way sites on the internet link to each other. In 2000, they introduced the concept of *initial attractiveness* [63] in a directed model that adds one node with $m_0$ connections per timestep, directed at nodes chosen using proportional selection relative to $(A_i + k_i^-)$, where $A_i$ is the (randomly assigned) initial attractiveness of node $i$ and $k_i^-$ is the in-degree of node $i$. They show that the in-degree of the generated networks follow a power-law distribution with $\alpha = 2 + \frac{A}{m_0}$ where $A$ is the sum over the initial attractiveness of all nodes.

The same authors further generalized the above model by creating a directed model [64] that, in addition to the $m_0$ connections assigned to a new node, also assigns two other sets of connections in each timestep that do not depend on the initial attractiveness:

- $m_p$ nodes are chosen randomly and a single connection is made between each of these nodes and a node chosen using proportional selection relative to in-degree; and
- $m_r$ connections are made randomly, without any preference.

This model leads to a power-law in-degree distribution with parameter

$$\alpha = 2 + \frac{m_p + m_r + A}{m_0} \quad .$$

Dorogovtsev and Mendes also observed that connections form at a rate superlinear in the addition of new nodes in real-world networks. They tried to reproduce this behavior through their *accelerated growth model* [65]: in addition to the $m_0$ connections assigned to a new node in the original PA model, this model also assigns $c_0 n^\tau$ additional directed connections per timestep, from randomly selected nodes to nodes chosen by proportional selection relative to initial attractiveness, where $\tau > 0$. The authors show analytically that this model generates networks with power-law in-degree distributions with parameter

$$\alpha = 1 + \frac{1}{1 + \tau} \quad \in (1, 2).$$

Dorogovtsev and Mendes [50] also introduced the concepts of *developing* and *decaying* networks. In their *developing network model*, a network is grown as in the PA model but a fixed number of new directed connections are added at each timestep between unconnected existing nodes $i$ and $j$, selected using proportional selection relative to the product of their degrees. In the *decaying network model*, a fixed number of random connections are removed from the network at each timestep.

The same authors also worked on a class of models that aim to eliminate the correlation between the age of a node and its degree. In their *gradual aging model* [66], older nodes lose their ability to attract new connections. In this model, the probability that a new node will connect to an existing node depends on the existing node's in-degree and its age, $a_i$: proportional selection being based on $k_i^- a_i^{-\nu}$ is used, where $\nu$ is a model parameter.[2]

---

[2]$a_i = (t - t_i)$ where $t_i$ denotes the timestep at which node $i$ entered the network and $t$ denotes the current timestep.

### 3.1.4.3  Non-linear preferential attachment model

In 2000, Krapivsky, Redner, and Leyvraz [67] proposed a generalization of the PA model that aims at increasing the flexibility of the model in producing networks with a variable power-law parameter, $\alpha \in \mathbb{R}$. Instead of using PA, proportional selection relative to some possibly non-linear function $f(k)$ of the nodes' degrees is used. However, they found that $f(k)$ needs to be asymptotically linear for the network to remain scale-free, i.e. $f(k) \in \Theta(k)$. The authors show that the resulting power-law degree distribution can be tuned to have any parameter $2 < \alpha < \infty$ in this case.

### 3.1.4.4  Fitness models

Bianconi and Barabási [68; 69] created a model which incorporates what they call the *competitive aspect* of real-world networks, in which nodes compete for connections, sometimes at the expense of other nodes. At each timestep, a new node $j$ with fitness $\eta_j$ is added to the network. $\eta_j$ is fixed for node $j$ and is chosen from a distribution $\rho(\eta)$. Each new node connects to $m$ other nodes in the network, chosen using proportional selection relative to $k_i\eta_i$, the product of each nodes' degree and fitness.

The resulting networks have power-law degree distributions, with the power-law parameter depending on the choice of $\rho(\eta)$. For a uniform $\rho(\eta)$, the degree distribution is proportional to $P(k) \propto \frac{k^{-2.225}}{\log(k)}$, a generalized power-law with an inverse logarithmic correction. [42].

Ergün and Rogers [70] proposed a generalization of this model that associates with each node $i$ a pair $(\eta_i, \xi_i)$, where $\eta_i$ is the random *additive fitness* of node $i$ and $\xi_i$ is the *multiplicative fitness* of node $i$. The additive fitness symbolizes that some nodes may be more attractive to connect to than others and the multiplicative fitness is used to create different categories of nodes which can form new connections at different rates. The network is grown as in the PA model, except that the proportional selection is now relative to $\xi_i(k_i - 1) + \eta_i$. Experimental results by the authors show that the resulting degree distribution still follows a power-law if the fitnesses are drawn from a power-law distribution.

### 3.1.4.5  The Klemm and Eguíluz model

In 2001, Klemm and Eguíluz [61; 71] proposed an extension to the PA model focused on including longer paths between nodes, high clustering for large networks

and a more flexible power-law degree distribution.  It is a dynamic model, keeping track of a subset of nodes called *active* nodes.  Starting from a fully connected network of $n_0$ active nodes, it cycles through three steps to develop the network:

1. A new node $u$ joins the network and for each active node $v$:

   - With probability $\mu$, node $u$ is connected to $v$.
   - With probability $(1 - \mu)$, node $u$ is connected to a random node in the network (active or non-active) using proportional selection relative to degree.

2. The new node becomes active.

3. One of the active nodes is deactivated. The node to be deactivated is selected using proportional selection relative to $\frac{1}{a+k_i}$, where $a > 0$ is a constant bias.

It is shown in [61] that the resulting degree distribution is a power-law with parameter:

$$\alpha = 2 + \frac{a}{n_0} \ .$$

Note that in the case where $\mu = 0$, this model reduces to the PA model.


#### 3.1.4.6  Dangalchev's two-level model

In 2004, Chavdar Dangalchev [72] proposed a model that extends the PA model by taking into account not only the degree of a node, but also the degrees of all of its neighbors.  Their intuition is to base the proportional selection not only on the number of neighbors a node has, but also on the popularity of its neighbors. The probability $p_i$ of connecting to node $i$ with $j$ neighbors is then proportional to $k_i + C \sum_j k_j$, where $C \in [0,1]$ is a constant weight for the importance of the second-level connections. If $C = 0$, then this model reduces to the PA model.  Through their experimental results, the authors found $C = 0.5$ to be a good choice.


### 3.1.5  Newman, Watts and Strogatz' model

Newmann, Watts and Strogatz [21] noted that the most serious limitation of the ER model is the Poisson degree distribution of the generated networks.  In 2002, they introduced a generalized version of the ER model that allows the formation of arbitrary degree distributions, whilst keeping the simplicity of the original model.

Their static model for undirected social networks takes as input the degree distribution $P(k)$ of the specific social network that is to be modeled. For each node, a value, $k$, is drawn from the prescribed distribution and the degree of the node is set to $k$ by assigning $k$ *stubs* to the node. Once the degree of every node in the network is known, the connections are randomly generated by repeatedly choosing two stubs from two different nodes and connecting them. If the number of stubs is odd, then one random stub is removed.

The main drawback of this model is that there is no simple way to extend it to the dynamic case. Also, the process may fail since unpaired stubs could remain. This could result in the stubs having to be re-distributed a number of times. It is also worth noting that the explicit inclusion of transitive behavior in this way may not incorporate the community structure that we observe in real-world social networks.

### 3.1.6  Ebel's transitive model

Few models in the literature deal with the removal of nodes and connections. This is mostly because of the lack of available data sets that include removed entities. Ebel et al. [73; 74] proposed one of the first models that constantly remove nodes from the generated network. Their model generates small-world, scale-free networks from the stationary state of a simple process. The model iteratively performs two actions, starting with an initial network with $n$ isolated nodes:

1. A random node is chosen from the network and two of the node's neighbors are randomly selected and connected to form a new triangle. If the node has less than two neighbors, it is connected to a randomly chosen node in the network.

2. With probability $p$ (a model parameter), a randomly chosen node is removed from the network. This node is then replaced by a new node with one random connection.

If $p > 0$, each node in the graph has a finite expected lifetime.[3] This leads to a stationary state of the network approximating the behavior of real social networks: numerical simulations have shown that this method generates networks with power-law degree distributions, short average path lengths and high clustering. The authors note that, for large enough networks, the parameter $\alpha$ of the

---

[3]The node's lifetimes are independent exponentially distributed variables with mean $\frac{N}{p}$.

power-law degree distribution depends only on the model parameter $p$. Although they do not give an analytical expression for $\alpha$, they give some obtained values of $\alpha$ for different $p$. In their experiments, they found small values of $p \ll 1$ to work best in modeling real-world networks. The major drawback of this model is that its stationary state outputs a single static social network. No evolutionary information for the network is available.

### 3.1.7  Newman's transitive model

Many models [71; 75; 76; 77; 78] have attempted to incorporate transitivity using some form of *triadic closure*[4] process, but because of the nature of the generation processes used by the models, their properties could only be calculated using numerical approaches. In 2009, Newman et al. [31] proposed a model that explicitly incorporates clustering and transitivity and for which they analytically obtained exact solutions for various properties of the resulting network.

The model takes as input the size of the network, $n$, together with $n$ tuples, $(s_i, t_i)$, where $t_i$ is the number of triangles in which node $i$ participates and $s_i$ is the number of connections of node $i$ which do not form part of any triangles. The degree of node $i$ is thus given by $k_i = s_i + 2t_i$, and the resulting degree distribution is given by

$$P(k) = \frac{\#\{i : k = s_i + 2t_i\}}{n} .$$

To construct this network, $t_i$ *triangle corners* and $s_i$ stubs are assigned to node $i$. The connections are created by choosing pairs of stubs uniformly at random and connecting them. After all the stubs have been paired, the triangle corners are randomly grouped into trios of distinct nodes and joined to form triangles. Note that in the process of generating single connections, some triangles may form by chance, but these are allowed, since the authors found their effect on the overall structure of the network for large $n$ to be negligible [31]. The only constraints are that the total number of stubs be even and the total number of triangle corners be a multiple of three.

### 3.1.8  Exponential random graph models

Exponential random graph models (ERGMs), also known as $p^*$ models, are widely studied for use in modeling social networks [79; 80; 81; 82; 83; 84]. ERGMs are prob-

---

[4]Triadic closure is the process of connecting two nodes based on the knowledge that they have a mutual neighbor.

abilistic models that explicitly define a probability density function for networks. The general form of such a density function for the class of ERGMs is given by

$$P(G = g) = \frac{1}{\kappa} \exp \left\{ \sum_{A \in \mathcal{A}} \eta_A \omega_A(g) \right\} , \tag{3.4}$$

where

- the summation is over a set $\mathcal{A}$ of *configurations*. A configuration is a subgraph with a specific structure (e.g. stars, triangles);
- $\eta_A$ is a parameter for the configuration $A$;
- $\omega_A(G)$ is the *network statistic* corresponding to configuration $A$ for the network $G$ (the number of occurrences of configuration $A$ in $G$); and
- $\kappa$ is a normalizing constant.

The configurations with non-zero parameters in Equation (3.4) specify a set of *conditional independence assumptions* about the occurrence of connections in the network: these conditions specify when the occurrence of a connection $e_1$ in the network is conditionally independent of the occurrence of another connection $e_2$ given the state of the rest of the network. Let $G'$ be the result of adding $e_1$ and $e_2$ to the rest of the given network. The occurrence of $e_1$ and $e_2$ are conditionally independent given the rest of the network if $G'$ has no subgraph containing $e_1$ and $e_2$ that matches a configuration. A number of different configuration sets have been used with ERGMs [79]; we will present the two most studied ones below.

### 3.1.8.1  Markov random graphs

The Markov random graph model was proposed by Frank and Strauss [80] in 1986, based on developments in spatial statistics [85]. It is built on the *Markov independence assumption*, which assumes that the occurence of a connection between node $i$ and node $j$ is dependent only on the other possible connections involving $i$ and $j$. That means the probability of occurrence of connection $(i, j)$ is independent of the probability of occurrence of any connection $(k, l)$ for $i \neq j \neq k \neq l$. An ERGM satisfying the Markov assumption must have the form:

$$P(G = g) = \frac{1}{\kappa} \exp \left[ pm_g + \sum_{k=2}^{n-1} \lambda_k S_k(g) + \tau T(g) \right] , \tag{3.5}$$

where:

- $p$ is the density parameter;
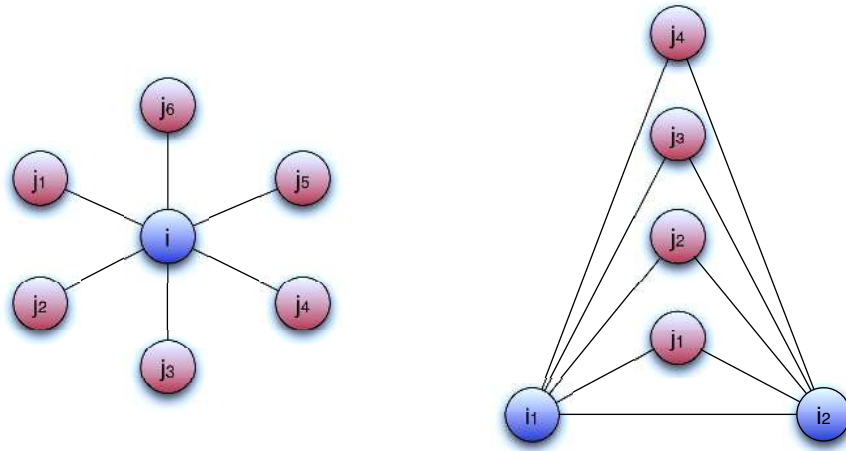
Figure 3.2: A 6-star (left) and a 4-triangle (right).

- $\lambda_k$ is the parameter associated with $k$-star effects;
- $S_k(g)$ is the number of $k$-stars in $g$;
- $\tau$ is the parameter associated with triangles; and
- $T(g)$ is the number of triangles in $g$.

This is because the Markov independence assumption disallows exactly those configurations containing a simple path[5] of length three: for a path $(u_1, u_2)$, $(u_2, u_3)$, $(u_3, u_4)$, the edge $(u_3, u_4)$ is not incident on $u_1$ or $u_2$, so its occurrence may not affect the probability of occurrence of $(u_1, u_2)$ .

To see why the inclusion of $k$-stars in (3.5) does not violate the Markov independence assumption, refer to Figure 3.2: it is clear that this graph contains no simple paths of length greater than 2. For higher-order stars ($k > 3$), $\lambda_k$ is often assumed to be 0 due to their relatively infrequent occurrence in real-world networks and in order to limit the number of parameters that need to be estimated in order to achieve a computationally feasible model for which parameters can be estimated [81]. An alternative is to use a single parameter for all $k$-triangle configurations. This is discussed in Section 3.1.8.2.

We see that the Markov independence assumption is violated if $k$-triangles for $k > 1$ are included in the configuration set since these graphs then contain simple paths with length greater than two.[6] When $k = 1$ (normal triangles), the Markov independence assumption is not violated, so the triangle configuration is included in the Markov random graph model. The parameter $\tau$ allows the model to explicitly

---

[5]A simple path is a path with no repeated nodes.
[6]For example, the 4-triangle contains the simple path $(i_1, j_1)$, $(j_1, i_2)$, $(i_1, j_2)$.

include some transitivity effects, allowing for modeling of the transitive behavior of social networks. When the star and triangle parameters are set to zero, so that the density parameter $p$ is the only non-zero effect in the model, all connections form independently of each other with a constant probability, reducing the model to the ER model [81].

Markov random graph models have been found to be unreliable in modeling social networks though, because the graph distributions obtained by estimating the parameters of the model are often *near degenerate* [83; 81]. A graph distribution is termed near degenerate if it implies only a small number of distinct graphs with substantial non-zero probabilities. For certain parameter values, Markov random graph distributions exhibit this property, with only nearly empty or complete graphs likely under the distribution.

### 3.1.8.2   Curved exponential family models

Recently, Snijders et al. [83] proposed violating the Markov assumption by adding higher-order *k*-triangle configurations to Markov random graphs. Violating the Markov independence assumption has been found to be crucial for modeling desirable global properties of social networks [86]. They also proposed two model constraints for including higher-order stars and triangles, the *alternating k-star* and *alternating k-triangle* constraints. By using the *alternating k-stars* constraint, the higher-order star parameters are no longer forced to be zero, but instead are related by:

$$\lambda_{k+1} = \frac{-\lambda_k}{\theta} \, , \tag{3.6}$$

for some $\theta > 1$. The authors formulated these constraints after they observed that higher-order star parameters often followed a pattern of decrease in magnitude, while alternating in sign, for successfully fitted network data sets. The single parameter $\lambda_2$ then captures the role of all star configurations.

Analogous to alternating *k*-stars, the authors also proposed *alternating k-triangles*, using the constraint of (3.6), but with $\lambda_k$ the parameter for a *k*-triangle. This moves beyond the Markov assumption in assessing the transitivity of a network. The use of alternating k-triangles expresses not only that there are comparatively many triangles in social networks, but also that there are not too many high-order *k*-triangles. For this reason, the use of *k*-triangles can include transitivity in the network, but also avoid the formation of overly dense, or even complete, networks.

#### 3.1.8.3  Estimation and simulation

Estimation of ERGM parameters has been the topic of much work [79; 81; 87; 88]. Standard maximum likelihood estimation is not tractable for real-world networks of significant size. Currently, these problems are circumvented by the use of techniques such as Markov chain Monte-Carlo (MCMC) [88] maximum likelihood estimation and pseudo-likelihood estimation [79; 81].

Given the set of parameters, a network can be sampled from the distribution using Monte Carlo techniques. The procedures for sampling networks using exponential random graph distributions are discussed by Strauss and Frank [80] and Robins et al. [89]. In general, these methods start with an initial (randomly chosen) adjacency matrix after which all the elements are repeatedly updated, in turn, until the matrix converges. An entry in the matrix is updated by generating its new value from the conditional distribution for that entry given all the other entries. The two major drawbacks of ERGM models are the high computational complexity that makes it infeasible for larger networks, and the fact that the generated networks are static.

### 3.1.9  Kronecker graphs

In 2005, Leskovec et al. [90] introduced a model that uses the *Kronecker multiplication* operation on the adjacency matrix of a network to grow the network. When a matrix is Kronecker multiplied by itself, each entry in the matrix is replaced by a scaled version of the original matrix:

$$
\mathbf{A}^{'} = \mathbf{A} \otimes \mathbf{A} = \begin{bmatrix} a_{1,1}\mathbf{A} & a_{1,2}\mathbf{A} & \cdots & a_{1,m}\mathbf{A} \\ a_{2,1}\mathbf{A} & a_{2,2}\mathbf{A} & \cdots & a_{2,m}\mathbf{A} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{A} & a_{n,2}\mathbf{A} & \cdots & a_{n,m}\mathbf{A} \end{bmatrix} .
$$

This model is based on the notion that power-law distributions are usually observed in self-similar structures, i.e. structures that consist of miniature copies of themselves [91]. In social networks, this means that communities are typically composed of structurally similar sub-communities. The authors suggest that this composition takes the form of an onion-like *core-periphery*, consisting of similar layers that increase in density as one moves towards the centre of the network.

The aim of this model is not to provide a realistic or intuitively natural way of generating networks, but to provide a model that can be rigorously analyzed ana-

lytically. The authors provide analytical proofs that their Kronecker model matches most of the static and dynamic properties of social networks.

Using Kronecker multiplication on a network $G$ with binary adjacency matrix $\mathbf{A} = [a_{i,j}]$, each non-zero entry of $\mathbf{A}$ is replaced by a copy of $\mathbf{A}$ and each zero entry is replaced by a zero-matrix of the same dimension as $\mathbf{A}$ to form the network $G'$ with adjacency matrix $\mathbf{A}'$. Recursively applying the Kronecker multiplication to an initial network $G_1$ with $n_1$ nodes and $e_1$ connections will, after $t$ timesteps, yield the network $G_t$ with $n_1^{2^t}$ nodes and $e_1^{2^t}$ connections. The original model, using a binary adjacency matrix, is deterministic for any given initial graph $G_1$. The authors noted this and also proposed a stochastic variant that treats the entries in the matrix as probabilities of the existence of connections. Although the fractal nature of the model reproduces desirable social network characteristics, this model has some intrinsic drawbacks:

- The number of nodes increases super-exponentially in the number of iterations. Although the model is dynamic, this means that only snapshots of the evolution are available, and the number of nodes and connections added between two snapshots increases drastically. This is not conducive to understanding the microscopic evolution of the network when nodes are added on an individual basis.
- The authors note that there is no way to determine a suitable initial graph $G_1$ other than a time-consuming brute-force search.

### 3.1.10   Forest fire model

As noted in Sections 2.2.2 and 2.2.6, Leskovec et al. [41] studied a collection of large online social networks in 2007, and found that their diameters decreased and their number of connections increased superlinearly in the number of nodes. They proposed a set of models, aimed at reproducing these characteristics, starting with their *community guided attachment* (CGA) model which they used to demonstrate that densification can naturally arise in a hierarchical community structure.

The *forest fire* (FF) model takes this a step further by incorporating both shrinking diameters and densification into a directed model. The forest fire model is a combination of proportional selection, Kumar's copy mechanism and their CGA model. It uses two parameters, the *forward burning probability* ($q$), and the *backward burning ratio* ($r$). When a new node $v$ joins the network:

1. $v$ chooses an *ambassador* node $w$ uniformly at random. $v$ is connected to $w$.

2. $v$ copies $x$ out-links and $y$ in-links from $w$. The values $x$ and $y$ are drawn from geometric distributions with mean $q/(1-q)$ and $rq/(1-q)$ respectively.

3. Step 2 is repeated for each of the $x+y$ new neighbors of $v$, allowing $v$ to recursively copy connections from its new neighbors, until the 'fire' dies out.

An intrinsic problem with this process is that no isolated nodes are present in the resulting network, since a node always connects to at least its ambassador. In fact, the entire network consists of one giant connected component. The authors note that this can be rectified by introducing another parameter which specifies the probability that a node will be inserted without any connections. This modification allows for multiple connected components, although these components can never merge, an aspect that does not translate well to real-world behavior. To address this, the authors proposed another modification to the algorithm, where a new node is allowed to choose more than one ambassador, possibly from two different communities. This allows communities to overlap and speeds up the decrease in diameter.

## 3.2   Top-down models

In contrast to the bottom-up models presented in the previous section, top-down models start by modeling the global community structure of the network and then translate this structure into a social network by inferring microscopic interactions between the nodes. All of the top-down models grow a bipartite network that represents affiliations of users to communities, information not present in bottom-up models. This bipartite network is then converted or *flattened* into a social network using some *flattening rule* that specifies the conditions under which connections in the social network are created. An example of such a bipartite structure and the resulting social network is shown in Figure 3.3. Note that all three models presented in this section use the same flattening rule depicted in Figure 3.3: converting communities into cliques over all their members.

It is important to realize that the communities we refer to here are the communities implicitly defined by the structure of the social network itself, typically corresponding to real-world communities such as families, schools, clubs and other social circles.[7] For an analysis of the evolution and structure of *explicitly* defined communities in social networks, i.e. communities/groups that the user joins on the network, refer to the work of Backstrom et al. [93].

---

[7]These communities are often identified as quasi-cliques in the social network. For an overview of methods for community detection in social networks, refer to [92].
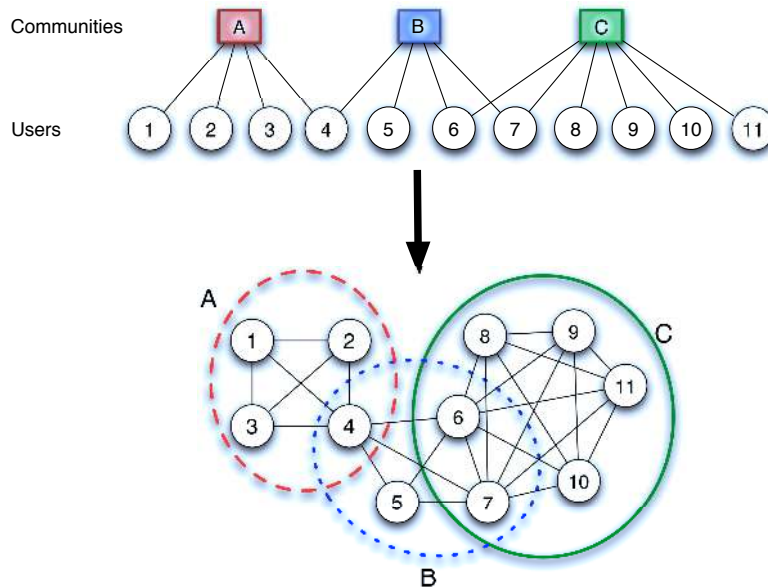
Figure 3.3: An example of a bipartite community structure (above) and the result-ing social network (below) formed using the deterministic flattening process used by all the existing top-down models.

In 2006, Guillaume and Latapy [27] showed that most real-world social and bi-ological networks can be modeled using bipartite networks.  They proposed an algorithm to create a bipartite network from a unipartite network by introducing a new set of nodes referred to as the *top* nodes, with the original nodes of the net-work referred to as the *bottom* nodes.  Each top node represents a maximal clique in the original network, and each bottom node is connected to all the top nodes corresponding to cliques which the node is part of in the original network.  They showed that this decomposition of social networks typically results in power-law degree distributions for the bottom nodes and Poisson or heavy-tailed distributions for the top nodes. This result was further confirmed by Latapy et al. [94].

These insights into social network structure gave rise to the study of top-down random graph models that can generate such bipartite networks, and convert them into social networks.  In this section, we present three existing top-down models in chronological order.  Two of these models, the model of Guillaume and Latapy (Section 3.2.1) and the model of Lattanzi and Sivakumar (Section 3.2.3) are dynamic models, adding new nodes and communities as the network evolves.  Birmelé's model (Section 3.2.2) does not allow for dynamic network generation: the entire bipartite structure must be built before it can be converted, as a whole, into a social network.

### 3.2.1   Guillaume and Latapy's model

Guillaume and Latapy proposed a dynamic model [27] (which we refer to as the *GL model*) for generating bipartite random graphs which takes as input two parameters, the mean $\mu$ of the Poisson degree distribution of the *top* nodes, and the *overlap ratio* $\lambda$. The parameter $\lambda$ is defined as

$$\lambda = 1 - \frac{n}{\sum_i |c_i|} \ ,$$

where $|c_i|$ is the size of the $i$-th maximal clique in the real-world network that is being modeled.  Then, in each iteration, a new *top* node is added, its degree $k$ is sampled from the Poisson degree distribution and $k$ connections are created in the following way:

- with probability $\lambda$, the connection is made to an existing *bottom* node using proportional selection relative to degree;
- otherwise a new *bottom* node is added and connected to the new top node.

They argue that the evolution of a network using this model is similar to how bipartite networks form in social contexts.  If a new group is created on Facebook, then a set of already active users join the group, but the group may also be joined by users who have never before participated in a group on Facebook and, as such, did not form part of the network previously.

Although this simple model yields an intuitive way of building the network, it has two notable shortcomings:

1. The way the communities form is not entirely consistent with real-world behavior.  In online social networks, communities typically comprise *quasi-cliques*, i.e. groups of nodes which are highly connected, but not completely.  Guillaume and Latapy's model generates networks using fully connected communities, generally resulting in a higher level of clustering than desired.

2. As they note, their model fails to incorporate *bipartite clustering*, with communities showing very little neighborhood overlap.  This is a major drawback, since, in real-world networks, if two communities have one node in common, they are likely to have more. In fact, real-world networks show a hierarchical community structure, with a recent study on a mobile social network revealing up to six levels in the community hierarchy [95].

### 3.2.2 Birmelé's model

In 2009, Birmelé [96] proposed a static model that samples a bipartite network and projects the network onto a unipartite counterpart. The bipartite network is sampled in the following steps:

1. Create a *top* set of $q$ nodes, denoted $v_1, \ldots, v_q$ and a *bottom* set of $n$ nodes, denoted by $w_1, \ldots, w_n$.

2. For each top node $v_i$ a set of $n + 1$ random variables are sampled:

   - $n$ variables are drawn from the uniform distribution $[0, 1]$ and are denoted $s_{i1}, \ldots, s_{in}$
   - A variable $p_i$ is drawn from the truncated power-law distribution over the range $[1, n]$. $p_i$ represents the expected resulting degree of $v_i$.

3. Given the set of random variables, node $v_i$ and node $w_j$ are connected if

$$s_{ij} < \frac{p_i}{n} \ .$$

   The degree of $v_i$ is thus binomially distributed with parameters $n$ and $p_i$.

Once the bipartite network is created, its unipartite projection is obtained by converting each top node into a clique over the bottom nodes connected to it. The authors proceed to show analytically that the degree distribution of the resulting network follows a power-law. They also show analytically that the generated networks show transitive behavior and that, with high probability, the generated networks have a high clustering coefficient.

Apart from being a static model, this model has another inherent drawback in that it does not model community overlap. If two bottom nodes $v_i$ and $v_j$ are connected to the same top node $w_k$, it does not increase the probability of them having more mutual neighbors in the bipartite network, which is not consistent with real-world behavior.

### 3.2.3 Lattanzi and Sivakumar's model

In 2009, Lattanzi and Sivakumar proposed a model that builds the user-community bipartite structure using a copy method similar to that of Kumar's model (Section 3.1.4.1). This model takes as input three parameters, $\beta, c_1, c_2 > 0$ and during each timestep:

- With probability $\beta$, a new user node $u$ is added and is assigned a prototype user node $u'$ from the existing network, chosen with proportional selection relative to degree. The node $u$ then copies $c_1$ connections from $u'$, that is, $u$ joins $c_1$ communities randomly chosen from the communities $u'$ belongs to.
- If no new user node is added, a new community node $q$ is added using a similar process, copying $c_2$ connections from a prototype community $c'$.

Like the previous two models, the bipartite network is converted to a social network using a simple flattening rule: each community node is translated to a complete graph on the community members. The authors prove that the resulting social networks have power-law degree distributions, obey the DPL, and their effective diameters stabilize to a constant.

One issue with this model is that it assumes the network does not have any inactive nodes. Each user node $u$ is connected to at least $c_1$ community nodes. In the resulting social network, $u$ will be connected to all other user nodes connected to any of its $c_1$ neighbors in the bipartite network. For $u$ to be an isolated node in the social network, all $c_u$ of these communities need to be empty, but this can not happen, since each community node is assigned $c_2$ connections upon its entry to the bipartite network. The social network can thus not contain any isolated nodes. In fact, depending on the choices for $c_1$ and $c_2$, there could be a very high minimum degree for the nodes in the social network. This would cause the tail of the degree distribution to be shorter than desired, similar to what was observed with the PA model in Section 3.1.3.

## 3.3   Conclusion

In Table 3.1, we present a summary of all of the models. It is worth noting that all of the models, except the ER model, produce small-world networks.

In Section 3.1, we presented a set of bottom-up models, some of are accurate at generating social networks, although they do not provide any information on how communities form and evolve in real-world networks.

The recent shift in research focus towards top-down models, as well as the industry's adoption of community-based approaches for online social networks, suggests that further understanding of the formation and structure of communities in social networks is indeed valuable. Section 3.2 presented the state-of-the-art in top-down models. However, all of these models have serious limitations. The model of Birmelé (Section 3.2.2) does not allow for dynamic generation of networks, whereas

the model of Lattanzi and Sivakumar (Section 3.2.3) imposes a restrictive lower bound on the degree of nodes, failing to include any isolated nodes in the generated networks. The model of Guillaume and Latapy is not subject to these shortcomings, although it uses the same naïve flattening rule as the other two models, assuming that all nodes in the same community will be connected. This is contrary to real-world behavior as exhibited by the implementation of the online social network Google Plus. In this social network, users define their own communities, or 'circles', explicitly, since each user has his/her own perspective of an implicit community. This view may often contain only a subset of the nodes that form the actual implicit community structure in the social network. This motivates the use of a more complex flattening rule, allowing for variable density of connections within communities, whilst retaining the transitive behavior within and across communities. This is the aim of our model, which is presented in the next chapter.

| Type | Section | Model (Year) | Static/Dynamic | Parameters | Degree distribution |
|---|---|---|---|---|---|
| **Bottom-up** | 3.1.1 | ER model (1959) | Static | $p$ | Binomial, Poisson as $n \to \infty$ |
| | 3.1.2 | Watts and Strogatz (1998) | Static | $K, \beta$ | Not PL, Poisson as $\beta \to 1$ |
| | 3.1.3 | PA model (1999) | Dynamic | $m_0$ | PL, $\alpha \approx 3$ |
| | 3.1.4.1 | Kumar's copy model (2000) | Dynamic | $m, p$ | PL, $\alpha = \frac{2-p}{1-p}$ |
| | 3.1.4.2 | Initial attractiveness (2000) | Dynamic | $m_0, P(A_i), m_r, m$ | PL (in-degree), $\alpha = 2 + \frac{m_p+m_r+A}{m_0}$ |
| | 3.1.4.2 | Accelerated growth (2000) | Dynamic | $m, \gamma$ | PL (in-degree), $\alpha = 1 + \frac{1}{1+\gamma} \in (1,2)$ |
| | 3.1.4.2 | Developing (2000) | Dynamic | $m, c$ | PL (in-degree), $\alpha = 2 + \frac{1}{1+2c}$ |
| | 3.1.4.2 | Gradual aging (2000) | Dynamic | $m, v$ | PL (in-degree), $\alpha \to 2$ as $v \to -\infty$ |
| | 3.1.4.3 | Non-linear PA (2000) | Dynamic | $m, f(k)$ | PL for $f(k) \in \Theta(k)$, $2 < \alpha < \infty$ |
| | 3.1.4.4 | Fitness (2000) | Dynamic | $m, \rho(n)$ | PL, $\alpha$ depends on $\rho(n)$ |
| | 3.1.4.4 | Multiplicative fitness (2001) | Dynamic | $m, (k_i, \eta_i, \xi_i)$ | PL, $\alpha = 2 + \frac{\sum_i \eta_i}{n}$ |
| | 3.1.4.5 | Klemm and Eguíluz (2001) | Dynamic | $n_0, \mu, a$ | PL, $\alpha \sim 2 + \frac{a}{n_0}$ |
| | 3.1.4.6 | Dangalchev's two level model(2004) | Dynamic | $m, C$ | PL |
| | 3.1.6 | Ebel's transitive model (2002) | Static | $p$ | PL |
| | 3.1.5 | Newman, Watts and Strogatz (2002) | Static | $\mathcal{D}$ | $\mathcal{D}$ |
| | 3.1.8.1 | ERGM - Markov (1986) | Static | $p, m_g, \lambda_k, \tau$ | PL |
| | 3.1.8.2 | ERGM - Curved exponential (2006) | Static | $p, m_g, \lambda_k, \tau$ | PL |
| | 3.1.7 | Newman's transitive model (2009) | Static | $\{(s_i, t_i) : i \in [1,\dots,n]\}$ | $P(k) = \frac{\#\{i:k=s_i+2t_i\}}{n}$ |
| | 3.1.9 | Kronecker graphs (2005) | Dynamic | Initiator graph $G$ | PL |
| | 3.1.10 | Forest fire model (2007) | Dynamic | $p, r, q$ | PL |
| **Top-down** | 3.2.1 | Guillaume and Latapy (2006) | Dynamic | $\mu, \mathcal{D}$ | PL |
| | 3.2.2 | Birmelé (2009) | Static | $q, r, \mathcal{D}$ | PL |
| | 3.2.3 | Lattanzi and Sivakumar (2009) | Dynamic | $c_u, c_q, \beta$ | PL |

Table 3.1: A summary of the models presented in this chapter. If an analytical expression is available for the power-law parameter $\alpha$, it is included in the last column.

# Chapter 4

# The proposed model

The literature review in Chapter 3 shows that researchers have recently started to focus more on top-down approaches, which build the social network using an intermediate bipartite structure, representing affiliations between users and communities. This is a very intuitive approach which translates directly to real-world behavior where one makes friends through interactions in the communities one belongs to. The existing top-down models have a major shortcoming though: the conversion of the affiliation network into a social network is a deterministic process that creates connections between all users that share a community. In this chapter, we present what is, to the best of our knowledge, the first top-down model that uses a probabilistic process to convert the affiliation network into a social network by generating a probability of existence for each potential connection in the social network. Preliminary results on an early version of this model was presented at the *Fourth ACM workshop on Social Network Mining and Analysis* at the KDD conference [97] in 2010 (proceedings in preparation). Section 4.1 gives a high-level description of our model and explains the intuition behind it. Section 4.2 formulates the building of the community structure, and Section 4.3 describes how social networks can be constructed from this community structure. In Section 4.4, we briefly relate our model to existing models.

## 4.1 Model Outline

A major shortcoming of the current dynamic top-down models, the GL model (Section 3.2.1) and the LS Model (Section 3.2.3), is that they assume that all users in the same community are friends. This is inconsistent with real-world scenarios where

there are many social factors that may influence whether two people will befriend each other within a community. This is evident from the fact that most state-of-the-art community identification methods use quasi-cliques to model communities [92].

In this section, we outline the first dynamic top-down model we are aware of that constructs a social network from communities consisting of quasi-cliques. Our model is thus the first top-down model that uses a probabilistic process to build the social network from the bipartite network, rather than the deterministic flattening process used in the models described in Section 3.2. An example illustrating our approach is depicted in Figure 4.1, which can be compared to the corresponding diagram for the deterministic flattening shown in Figure 3.3.
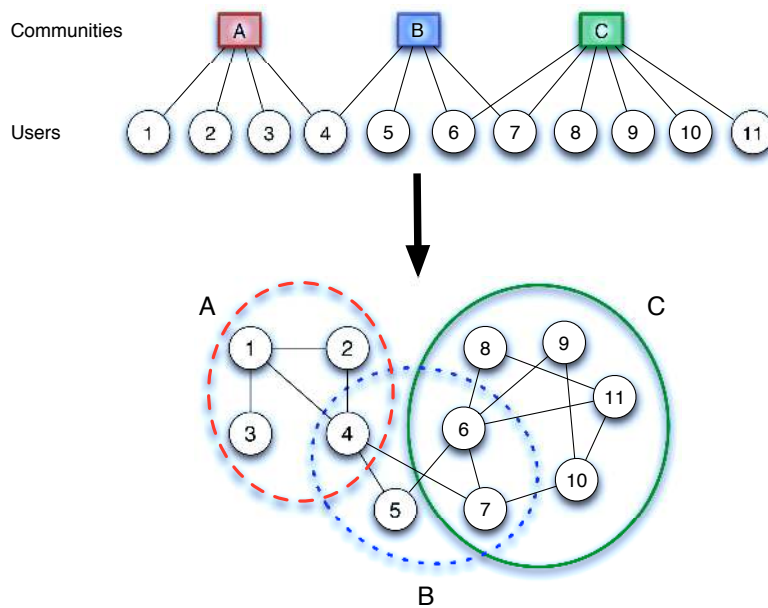


Figure 4.1: An example of a bipartite community structure (above) and a possible sampled social network (below).

To accurately build a social network using a bipartite community structure, there are two important social factors that should be examined and incorporated in the model:

1. The dynamics governing the involvement of users in communities, a process partially captured in the degree distributions of the users and communities in the bipartite network.

2. The factors that influence the probability of two people befriending each other given their set of mutual communities.

For the former, we can refer to the work of Birmelé [96] and Guillaume and Latapy [27] for an indication of the shape of these degree distributions in real-world networks. Their results on fully connected communities, a well as our intuition on social behavior, suggests that the degree distribution of the users in the bipartite network is consistent with that of social networks, in which the degree distribution follows a power-law. For the communities, the experimental results of Birmelé [96] suggest that the degree distribution seems to vary based on the specific structure of the network, with both Poisson distributions and heavy-tailed distributions observed in the analyzed networks.

Taking this into consideration, it follows quite naturally that users could join communities based on some form of proportional selection, which should yield the desired power-law degree distribution for the user nodes. However, there are two obstacles that need to be avoided:

- The observed 'rich-get-richer' phenomenon that results from traditional degree-based PA. This leads to a correlation between the age and the degree of a node, which is not observed in real-world data sets (as discussed in Section 3.1.3).
- PA offers little flexibility in the degree distributions of the resulting networks. This would definitely constrain the generality of the model.

To ensure that no correlation between the age and degree of a node will exist, we note that this problem is inherent to PA since an increment in the degree of a node $u$ immediately increases the probability of $u$ forming more connections. This can be remedied by using proportional selection relative to some measure of 'popularity' of the node which is not correlated with its degree. We proposed to assign to each user node a constant *activity*, sampled from an arbitrary *activity distribution* $\mathcal{D}_1$ on the positive real values. The freedom to specify $\mathcal{D}_1$ also allows us a great deal of flexibility in constructing the community structure, unlike the constrained development of PA.

The first major divergence of our model from the GL and LS models is that communities in the bipartite network are modeled as quasi-cliques rather than cliques. Determining the probability of two users befriending each other in a community is a complex task because real-world social behavior plays an important role and is hard to quantify. Here, we take a first step in modeling this complex process by observing that in real-world communities, the probability of two people meeting

definitely depends on their level of involvement in the community. We incorporate this into the model by associating a constant *commitment value* for each community a user belongs to, drawn from an arbitrary *commitment distribution* $\mathcal{D}_2$ on the positive real values. The probability that two users are connected in the social network will then be based on their commitments to all of their mutual communities.

We choose not to assume any specific shape for the degree distribution of the communities, but rather use an intuitive construction process based on real-world behavior. We note that, in real life, a person usually joins communities that overlap with at least one of his/her existing communities. This is both because of social behavior and geographical restrictions. Thus, when choosing a new community for a user to connect to, we only consider those communities that overlap with the existing communities of the user. The decision is then made using proportional selection based on the overlap sizes.

Since these connections are formed using only structural knowledge from the bipartite network, the development of the community structure is independent of any social networks sampled from it. This allows us to use the model for data set generation in two different ways:

1. If we are interested in the evolution of the social network, the network can be built incrementally together with the community structure.

2. If we are interested in a static social network data set of order $n$, we can generate a community structure with $n$ users and sample a social network from it.

Since we use a probabilistic method to sample the social networks from the community structure, we can also generate multiple social networks from any proposed community structure, unlike the other top-down models.

## 4.2   Community model construction

Our model constructs two networks, a community structure $B$ and a social network $G$. In this section, we present the algorithm for building $B$ and, in the next section, we describe how $B$ is converted into a social network $G$.

Our model is initialized by inserting one community and one user node, and connecting them. We grow the community structure, represented as a bipartite network, over a series of time steps as follows. During each time step:

1. With probability $\beta$, a new community node $c_i$, with no connections, is added to $B$.

2. With probability $\gamma$, a new user node $u_j$ is added to $B$. An *activity* value, $a_j$, is associated with this user node. Upon joining the network, the node is connected to one community, chosen uniformly at random, and the connection is assigned a commitment.

3. With probability $\delta$, an existing user node $u_j$ is connected to an existing community node $c_i$, and the connection is assigned a commitment:

   - The node, $u_j$, is chosen using proportional selection relative to activity, i.e. the probability of node $j$ being chosen is equal to

   $$p_j = \frac{a_j}{\sum_{k=1}^{n} a_k} \; .$$ (4.1)

   - The community $c_i$ is chosen using a two-step process: First, a community $c$ that $u_j$ is connected to is selected using proportional selection based on the commitments of $u_j$. Then, $c_i$ is selected from the set of communities $u_j$ is not a member of, using proportional selection based on the *overlap* between $c$ and these communities.[1] The overlap $\theta(c, c_k)$ is defined as the number of mutual members of $c$ and $c_k$.[2]

   Whenever a user node $u_j$ is connected to a community node $c_i$, a weighted connection is created in the bipartite network between $u_j$ and $c_i$. The connection weight $\delta_{ji}$ indicates the user's *commitment* to the community.

The activity values $a_j$ are sampled from the activity distribution $\mathcal{D}_1$ and the commitments $\delta_{ik}$ are sampled from the commitment distribution $\mathcal{D}_2$. For the purposes of this study, we decided to use power-law distributions with parameters $\alpha_1$ and $\alpha_2$ respectively. Using a power-law for the activities is a natural choice, since one basic measure of the activity of users on social networks, the users' degrees, follows a power-law. For the commitments, there is less evidence to go on, but our intuition leads us to believe that, in real life, this distribution should be heavy-tailed since it is common to fully commit to only a few communities, while having low involvement in many other communities.

---

[1] If the node $u_j$ is already connected to all the communities in the network, we skip this step and continue with the next iteration of the algorithm.

[2] In the special case where $\theta(c, c_k) = 0$ for all $k$, i.e. no communities overlap with the communities $u_j$ is a member of, we choose $c_i$ uniformly at random from the communities $u_j$ is not a member of.

## 4.3   Social network construction

To construct a social network from the obtained bipartite community structure, we infer the probability of existence of all the connections in the social network. Two users become friends if the connection between them is enabled by any mutual community. Let $C_{i,j,k}$ be the event that the connection between users $i$ and $j$ is enabled by their mutual community $k$. We then define the probability of this event occuring as:

$$P(C_{i,j,k}) = f(\delta_{ik}, \delta_{jk}) = \frac{1}{w} \exp\left(\frac{-1}{\delta_{ik}}\right) \cdot \exp\left(\frac{-1}{\delta_{jk}}\right) . \tag{4.2}$$

where $w > 0$ is a global constant scaling parameter that can be used to increase/decrease the level of connectivity across all communities.

We explain the intuition behind this definition through a theoretical analogy, which we refer to as a *community grid*.

### 4.3.1   The community grid

In Figure 4.2, we present a community grid which takes the form of a layered structure where each layer corresponds to a single community in the bipartite network. Figure 4.2(a) shows the layout for a single community $k$ in a social network with $n = 4$ users, whereas Figure 4.2(b) represents a community grid for a bipartite network with 4 users and 3 communities. The connection weights are given by the commitments of the users to the communities, which is obtained from the bipartite network. We set this value to zero if there is no connection between a user and the community in the bipartite network.

Each sphere in the community grid represents a light bulb which glows if the two nodes connected to it befriend each other in that community. $C_{i,j,k}$ is then the event that the light bulb in community $k$, connected to users $i$ and $j$, are glowing. Because we are modeling undirected networks, none of the communities in the community grid in Figure 4.2 contain connections in the lower triangle. There are also no connections along the diagonal because a user can not befriend itself.

A light bulb will glow under the following conditions:

- The bulb itself is not defective. This occurs with probability $P(\text{Bulb } k \text{ working}) = \frac{1}{w}$ where $w$ is a constant.
- Both of the bulb's connections (wires) are conducting electricity. The probability that a wire connecting user $i$ and a sphere in community $k$ will conduct

(a) Community $k$ with n = 4 users

(b) A Social Network with n = 4 users and 3 communities
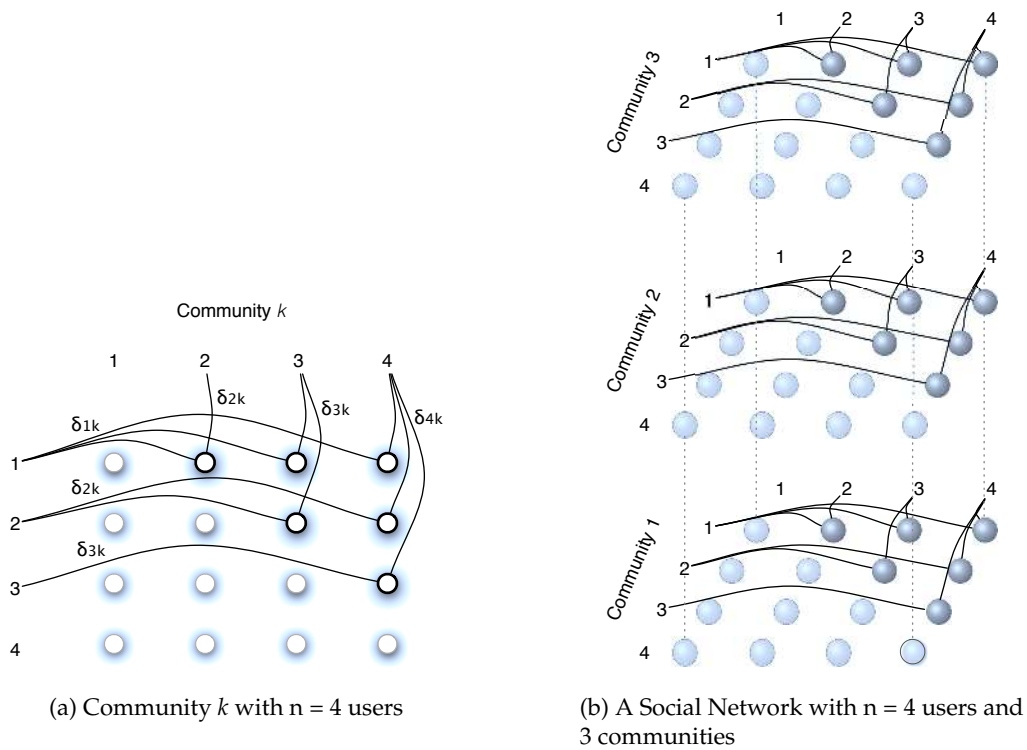
Figure 4.2: A community grid, illustrating how the communities and commitments are used to determine the probabilities of connections in the social network. See Section 4.3 for a full discussion.

electricity is given by:[3]

$$P(\text{ Wire } ik \text{ conducting}) = \begin{cases} \exp\left(\frac{-1}{\delta_{ik}}\right) & \text{if user } i \text{ is connected to community } k, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

Since $\delta_{ik} \in (0, \infty)$, the unbounded commitment value is thus transformed into a probability in the interval $(0, 1)$. This transformation is monotonically increasing, so that high commitment values correspond to likely connections through the community.

The bulb will only glow if both conditions hold, so the probability that the bulb

---

[3]Note that in the limiting case $\lim_{\delta_{ik} \to 0} \exp\left(\frac{-1}{\delta_{ik}}\right) = 0$, which makes the zero probability a natural choice when a user is not connected to a community.

connecting users $i$ and $j$ in community $k$ will glow is given by:

$$
\begin{aligned}
P(C_{i,j,k}) &= P(\text{Bulb } k \text{ working}) \cdot P(\text{ Wire } ik \text{ conducting}) \cdot P(\text{ Wire } jk \text{ conducting}) \\
&= \frac{1}{w} \exp\left(\frac{-1}{\delta_{ik}}\right) \cdot \exp\left(\frac{-1}{\delta_{jk}}\right) \ ,
\end{aligned}
$$

if $\delta_{ik}, \delta_{jk} > 0$, which is Equation (4.2). At any stage in the construction of the bipartite network, there will be a connection between users $i$ and $j$ in the social network if any of the bulbs connecting them in the community grid are glowing. We refer to all the bulbs connecting users $i$ and $j$ in the community grid as $column(i, j)$. Four of these *columns* are depicted in Figure 4.2 by vertical dashed lines.

As discussed in Section 4.1, our model allows for both the static and dynamic construction of social networks from the bipartite community structure. We present both cases below.

### 4.3.2  Generating social networks

A very desirable property of a social network model is to have the ability to construct networks dynamically. Using our model, a social network $G$ can be constructed dynamically from a bipartite structure $B$, as follows:

- Whenever a new user node $u_j$ is added to $B$, it is copied into $G$ as an isolated node.
- Whenever a new commitment between a user $u_j$ and a community $c_k$ is added to $B$, the commitment is also added to the community grid. If this causes a bulb to switch on in any $column(i, j)$ for $i \in (1, n) \neq j$, then a connection is added between users $i$ and $j$ in the social network.

Since users can only befriend each other once in a social network (assuming the removal of connections are not allowed), we are only interested in the *first* time a bulb in a particular column switches on. This bulb will keep glowing, and render the state of all the other bulbs in that column irrelevant.

This means that whenever a new commitment is inserted into $B$ between $u_j$ and $c_i$, connections are inserted in the social network between $u_j$ and all other members of $c_i$ with probabilities calculated using (4.2). Thus, whenever user node $u_j$ is connected to a community node $c_i$ in $B$, $u_j$ is connected in $G$ to each member $u_k$ of $c_i$ to which $u_j$ is not already connected to in G, with probability $f(\delta_{ji}, \delta_{ki})$.

We define $D_{i,j}$ as the event that two users $u_i$ and $u_j$ are connected in the social network after all the communities and commitments have already been added to

the bipartite network. To calculate $P(D_{i,j})$ we note that two users will be connected in the social network if the connection between them is enabled by any of their mutual communities, so

$$
\begin{aligned}
P(D_{i,j}) &= P\left(\bigvee_{k=1}^{r} C_{i,j,k}\right) \\
&= \sum_{k=1}^{r} P\left(C_{i,j,k} \bigwedge_{l=1}^{k-1} \overline{C}_{i,j,l}\right) \\
&= \sum_{k=1}^{r}\left[P(C_{i,j,k}) \cdot \prod_{l=1}^{k-1} P(\overline{C}_{i,j,l})\right] \quad \text{(by independence of } C_{i,j,l}) \\
&= \sum_{k=1}^{r}\left[f(\delta_{ik}, \delta_{jk}) \cdot \prod_{l=1}^{k-1}(1 - f(\delta_{ik}, \delta_{jk}))\right],
\end{aligned}
\tag{4.4}
$$

where the initial disjunction and final sum are over the $r$ mutual communities of users $i$ and $j$. Due to the probabilistic nature of this sampling process, it can be repeated many times to yield a set of distinct, yet structurally similar social networks.

## 4.4   Relationships to other models

From (4.4) it follows that the probabilistic flattening rule in our proposed model simplifies to the deterministic version used in the GL and LS models when $f(\delta_{ik}, \delta_{jk}) = 1$. This corresponds to the limiting case of our model where $w = 1$ and $\delta_{ik}, \delta_{jk} \to \infty$, i.e. all user nodes have an infinitely strong commitment to all of their communities. If, in this case, we define $w > 1$, each community will flatten into an ER network with $p = 1/w$. Note that this does not mean the resulting social network will be an ER network since the communities will overlap, resulting in some connections being more likely to occur than others.

The only case in which our model will reduce to the ER model is when the bipartite structure is complete, $\beta = 0$, and all commitments are equal. That is, all user nodes are connected to the single community in the network with equal commitment $\delta$. In this case, the model will produce social networks equivalent to ER networks with parameter $p = \frac{1}{w}\exp\left(\frac{-2}{\delta}\right)$, where $c$ is the number of community nodes in the bipartite network.

A more thorough experimental comparison of our model to the PA model and the GL model is presented in Chapter 6.

## 4.5  Conclusion

In this chapter, we proposed the first top-down model that uses quasi-cliques to build communities in the social network. Top-down models build the social network using an intermediate bipartite structure representing the affiliations of users to communities. In building the bipartite structure, our model uses proportional selection relative to the activity of the user nodes in the network when choosing a user node to join a community. Our model explicitly models community overlap through bipartite clustering by choosing the community a user connects to from the set of communities that overlap with an existing community of the user, using proportional selection relative to overlap size. We proposed a probabilistic flattening process for converting the community structure into a social network by generating a probability of existence for each possible connection in the network. These probabilities are based on the commitments of the two users to their mutual communities and only allows connections between users that have at least one mutual community. We explained the sampling of both dynamic and static social networks from this structure. Finally, we showed that for certain limiting cases, our model reduces to the simpler top-down models as well as the ER model.

# Chapter 5

# Fitting the Model

*Fitting* a model to a specific real-world network refers to the process used for assigning values to the parameters of the model, usually with the aim of minimizing the structural differences between the original real-world network and the network(s) generated by the model. The importance of fitting a social network model to real-world networks is two-fold: first, to evaluate the ability of the model to reproduce social network characteristics; and second, for practical use, such as prediction of the future growth of the network. In Section 5.1 we give an overview of the fitting methods used by existing random graph models. In many cases, fitting the model involves searching a parameter space. Searching this parameter space exhaustively is usually not computationally tractable. In Section 5.2 we give a general introduction to metaheuristics used for guiding the search of this parameter space. In particular, Section 5.2.1 and Section 5.2.2 present overviews of two metaheuristics, gradient-descent and simulated annealing. We implemented a modification of simulated annealing for fitting our model and we discuss our adaptations to the original algorithm in Section 5.3. An evaluation of these adaptations is presented in Section 5.4. Section 5.5 presents a comparison of our technique to existing stochastic approximation techniques.[1]

---

[1]During the early development of the model, when the parameter selection technique described in this chapter was developed, we were not aware of stochastic approximation techniques which might be applicable to the characteristics of the optimization problem encountered. Although the optimization technique presented in this chapter was derived without prior knowledge of stochastic approximation techniques, we discovered at the eleventh hour that the resulting approach is similar to some of these techniques. The motivation and development of our approach are presented here for completeness, since our source code and all experiments in this study make use of our proposed technique.

## 5.1  Background

There are many different ways to fit a model to a network. The simpler single-parameter models, like the ER and PA models (Sections 3.1.1 and 3.1.3) use simple statistics from the real-world network to determine values for the model parameters. For example, if we assume that we require the model to produce networks that closely match the order and size of the original network, then the choice for the parameter of the ER model might be the maximum likelihood estimate $p = 2m/n(n-1)$ where $n$ is the number of nodes in the network and $m$ is the number of connections (see equation (3.2)). For the PA model it would be usual to choose $m_0 = 2m/n$, since this would yield a network with exactly $n$ nodes and $m$ connections.

Some other models use more advanced statistical descriptions of the original network to choose the model parameters. An example of such a model is the model of Newmann, Watts and Strogatz (Section 3.1.5) which uses the degree distribution of the original network as a parameter. The model of Guillaume and Latapy (Section 3.2.1) uses values obtained from the maximal clique decomposition of the original network to calculate the parameters.

Advanced network statistics are also used for maximum likelihood estimation for ERGM models. This often leads to problems, due to the computational infeasiblility of calculating these statistics in larger networks and the near-degeneracy of the distributions (see Section 3.1.8.1).

It would be convenient to be able to estimate parameters for our model using statistics of the original network. However, there is no clear direct relationship between parameter values in our model and the resulting network characteristics. We are thus forced to use a different approach. A common method used in cases where there is little knowledge of how to calculate parameters based on network statistics, is to determine good parameters by evaluating different parameter combinations, i.e. searching the parameter space. This method can be used if there is some way of evaluating the quality of networks generated with a specific parameter set. Typically, such an evaluation considers a combination of characteristics of the original and generated networks, and combines them into a goodness-of-fit metric.

An intrinsic issue with most parameter estimation techniques is that they are trained on a single snapshot of the real-world network, implicitly assuming that all networks with the same characteristics at a given point followed the same evolutionary pattern to that point. In general, this assumption is not reasonable. Since searching the parameter space does not directly use any of the statistics of the orig-

inal network to calculate parameters for the model, but only to evaluate the model, it is more suitable for fitting dynamic networks. This is because it is easier to evaluate the similarity between two evolving networks than to summarize the complex evolutionary patterns of a network in a real-valued model parameter.

The most naïve approach to searching the parameter space is evaluating all possible parameter combinations. The processing required for this grows exponentially with the number of parameters, so for models with many and/or real-valued parameters, this method is generally not computationally feasible. In the next section, we present some heuristics that can speed up such a search process.

## 5.2 Algorithms for multi-dimensional optimization

*Metaheuristics* are a set of heuristics that attempt to minimize some objective function $E(s)$ (also known as an energy function) by starting with some initial candidate solution vector $s$ and then iteratively trying to improve on the candidate solution in order to find a solution with low value. Metaheuristics are typically applied to problems where there is little or no guidance as to how to find an optimal solution, but there is a way to evaluate $E(s)$, the quality of any solution $s$. Many metaheuristics do this by implementing some form of *stochastic optimization* [98].[2]

One of the most basic metaheuristics is known as *hill-climbing*. The hill-climbing algorithm starts with a random solution $s$ for a discrete objective function. Then, a small modification to one of the components of $s$ is made, and the new version is evaluated. If the new version is better, it is kept, otherwise it is discarded. This process is repeated as long as is feasible.

### 5.2.1 Gradient-descent algorithm

The choice of which algorithm to use often depends on the knowledge one has of the objective function's behavior in the search space. When the objective function is known to have a certain shape, that extra information can be utilized by the algorithm. When optimizing a differentiable objective function, one can often improve the hill-climbing algorithm by making use of gradient information. One such technique is called the *gradient-descent algorithm*.

---

[2]Stochastic optimization techniques employ some degree of randomness, attempting to optimize complicated functions.

The gradient-descent algorithm minimizes a differentiable function $E(s)$. The value of $E$ at $s$ need not be computed; instead the gradient of $E$ must be computed at $s$. During each iteration the gradient $\nabla E(s)$ is determined and the candidate solution point is shifted by an amount proportional to the negative of the gradient, i.e.

$$s^{'} = s + \kappa \cdot \nabla E(s)$$

for some constant $\kappa > 0$.[3] This means that unlike the hill-climbing algorithm, all of the components of $s$ may change during each iteration. Because the change is made in the direction of steepest descent, and bigger changes are applied when the gradient is steep, the algorithm usually approaches the minimum much faster than the hill-climbing algorithm.

### 5.2.2  Simulated annealing

Both the hill-climbing and gradient-descent algorithms are designed to find the minimum of $E(s)$ within the basin of attraction[4] in which $s$ lies. This means that when multiple local minima are present, the search process must be repeated from a set of starting points with elements in all the basins of attraction to ensure that the global optimum is found. Since the locations of these basins are unknown, in practice the process is repeated from a number of randomly chosen starting points. This process, referred to as performing *random restarts*, will be computationally expensive if one wishes to perform enough restarts to guard against the possibility that the objective function contains many local optima.

*Simulated annealing* (SA) is another metaheuristic commonly used for avoiding local optima. It is based on a method used in metallurgy, where the heating and controlled cooling of a material can increase the size of its crystals to make it less brittle. Each point $s$ in the search space is analogous to a state of the physical system. The goal is to find a state $s$ that minimizes the energy function $E(s)$. The original simulated annealing algorithm, formulated by Kirkpatrick [99], is presented in Algorithm 5.1. [5]

---

[3]Note that the value of $\kappa$ may change over time, typically decreasing with the number of iterations.

[4]A basin of attraction is a concept from the theory of dynamic systems: in our context, it refers to a set of points from where the same local minimum will be reached by continually moving downhill.

[5]Technically, the original formulation does not allow for storing good solutions encountered along the search path, because this breaks the analogy with the physical process where the material can not be restored to a previously encountered state. However, this is a standard optimization, so we include it in the algorithm.

---

**Algorithm 5.1** Simulated annealing

---

$s \leftarrow$ randomState()                    ▷ Starting state is chosen at random
$s_{best} \leftarrow s$                                   ▷ Store initial best solution

$T \leftarrow T_0$                                              ▷ Initial temperature
$k \leftarrow 0$
**while** $k < k_{max}$ **do**                          ▷ While there's time left
    $s' \leftarrow$ neighbor($s$)                     ▷ Pick a candidate neighbor
    **if** $E(s') < E(s_{best})$ **then**      ▷ If the neighbor's energy is the lowest so far
        $s_{best} \leftarrow s'$                          ▷ Store new best solution
    **end if**

    $T \leftarrow$ Temp($k/k_{max}$)              ▷ Determine the current temperature
    $\Delta E \leftarrow E(s) - E(s')$            ▷ Compute energy difference
    **if** $P(\Delta E, T) >$ random() **then**   ▷ Are we moving to the neighbor?
        $s \leftarrow s'$                               ▷ Move to the neighbor
    **end if**
    $k \leftarrow$ k+1
**end while**

**return** $s_{best}$                                      ▷ Return the best solution found

---

During each iteration, the algorithm considers a neighbor $s'$ of the current state $s$, and moves to the state $s'$ with a probability that depends on the change in energy between the states, $\Delta E = E(s) - E(s')$, as well as the current *temperature*, $T_k$.

The *acceptance probability function* $P(\Delta E, T_k)$ is used to determine the probability that the algorithm will move from state $s$ to state $s'$ during iteration $k$. An essential requirement for the function $P$ is that it must be non-zero when $E(s) > E(s')$, i.e. it must allow the system to move to a worse state with non-zero probability. This enables the algorithm to avoid local optima in the search space. In the original formulation of the algorithm, Kirkpatrick et al. [100] defined the acceptance probability function as:

$$P(\Delta E, T_k) = \begin{cases} 1 & \text{if } E(s') < E(s) \\ \exp\left(\Delta E\right)/T_k & \text{if } E(s') \geq E(s) \end{cases}$$

Thus, the algorithm always moves to the new state $s'$ if it has lower energy, but if $s'$ has higher energy a probabilistic choice is made based on the energy difference and the temperature.

The effectiveness of the algorithm depends heavily on the manner in which the temperature is changed over time. The changing temperature is simulated using a

variable

$$T_k = \text{Temp}(k/k_{max}) \, ,$$

that is high for small values of $k$ and then gradually decreases according to some *annealing schedule*, as $k$ increases.[6]

The result of using the annealing schedule is that, initially, the system moves almost randomly in the search space, trying to find a broad region of low energy and then, as $T_k$ decreases, it becomes increasingly focused on moving to states with lower energy. In the final stages, the algorithm almost behaves like the hill-climbing algorithm, where it almost certainly chooses to move to states with lower energy in an effort to find the optimum within the local region.

## 5.3  Searching for parameters for our model

As mentioned in Section 5.1, in order to perform a search for optimal parameters, one needs a way of evaluating the quality of the networks generated by a specific set of parameters. In Chapter 2 we listed a number of key network characteristics which can be used to compare a generated network $G$ to a real-world network $H$. We combine these characteristics into an energy function which attempts to quantify the difference in structure between $G$ and $H$. This energy function can then be used to evaluate the parameters from which $G$ was generated. Since we are interested in the evolution of the network $G$, we refer to the snapshot of network $G$ that contains $i$ nodes as $G_i$. The evolution of the network is then given by the series of snapshots $\{G_0, G_1, \ldots, G_i, \ldots, G_n\}$.

### 5.3.1  Energy function

In order to make our comparison of the networks as complete as possible, we base our energy function on a combination of temporal and static characteristics. For each characteristic $C$ we define $C(G)$ to be the characteristic value in network $G$ and $\Delta\left[C(G), C(H)\right]$ to be the difference between $G$ and $H$ with respect to characteristic $C$. The energy function $E_H(G)$ of the network $G$ relative to a network $H$ is then a weighted sum over these differences for various characteristics,

$$E_H(G) = \sum_{C \,\in\, Characteristics} w_C \cdot \Delta\left[C(G), C(H)\right] \, , \tag{5.1}$$

---

[6]The most popular choice for the annealing schedule is: $T_{k+1} = \alpha T_k$, where $\alpha$ is a constant, typically in the range $0.8 < \alpha < 0.95$ [101]. This corresponds to the temperature function $\text{Temp}(x) = T_0 \alpha^{x k_{max}}$.

where $w_C$ is the weight assigned to characteristic $C$.[7]. The weights allow our model to concentrate more on some characteristics than others. Our goal is then to find a parameter set yielding a low energy.

We chose the following set of characteristics to use in (5.1):

1. The evolution of the clustering coefficient over the interval $i \in [0, n]$. This is a bounded function: $CC(G_i) \in [0, 1]$.

2. The evolution of the power-law parameter of the degree distribution over the interval $i \in [0, n]$. This function typically lies within the bounds: $PL(G_i) \in [1, 3]$.

3. The average degree of the nodes of $G_n$. This value is bounded: $deg(G_n) \in [0, N - 1]$.

4. The densification power law exponent calculated over the interval $i \in [0, n]$. This is a bounded value: $DPL(\{G_0, \ldots, G_n\}) \in [0, 2]$. Together with the average degree, this summarizes the connection growth pattern.

5. The proportion of nodes that are isolated in the full network. This value is bounded: $iso(N) \in [0, 1]$.

For the static, scalar-valued characteristics (proportion of isolated nodes, average degree, DPL exponent), we define $\Delta\left[C(G), C(H)\right]$ as the relative deviation of $C(G)$ from the characteristic $C(H)$ of the original network:[8]

$$\Delta\left[C(G), C(H)\right] = \frac{|C(G) - C(H)|}{C(H)} \; .$$

For the clustering coefficient and power-law parameter we use a temporal measure of difference:

$$\Delta\left[C(G), C(H)\right] = \frac{1}{n}\sum_{i=1}^{n}\frac{|C(G_i) - C(H_i)|}{C(H_i)} \; . \tag{5.2}$$

Note that this process requires the generation of a full network, which could be computationally expensive. The computation required for the generation of a network is also a random quantity. An important factor in this process is the choice

---

[7]When it is clear from the context which network $H$ the network $G$ is being compared to, we omit the subscript and refer to the energy function as E(G)

[8]Note that $\Delta\left[C(G), C(H)\right]$ (also used by Toivonen et al. [102]) is not symmetric over the domain of the parameters since, for small values of the denominator, overestimates can attain much higher values than under-estimates, which can attain a maximum value of 1. Experimentally, we found this asymmetry to benefit our search process since overestimates of three of the characteristics (clustering coefficient, average degree and DPL) correspond directly to denser networks, which take longer to process.

of weights in (5.1). Different values for the weights could bias the search process to concentrate more on certain characteristics of the generated networks. We used an ad-hoc method for choosing the weights, aimed at avoiding unwanted bias towards certain characteristics, and found the following relationship to work well:

$$w_{CC} \approx 1.25w_{PL} \approx 2w_{DPL} \approx 2w_{deg} \approx 5w_{iso}$$

We now discuss our search strategy for attempting to minimize the objective function.

### 5.3.2 Overview of our stochastic optimization metaheuristic

In Section 5.2, we discussed the gradient-descent and simulated annealing algorithms. If the energy function contains few local optima and its gradient can be computed efficiently, then the simple gradient-descent algorithm, combined with random restarts, should suffice to find a solution close to the global optimum. In our case though, the objective function may contain a high number of local optima. Using the gradient-descent algorithm would require estimating the gradient at every timestep,[9] and many random restarts would likely be required to obtain a good solution. For these reasons the gradient-descent algorithm is not a computationally feasible option in our case. Simulated annealing is much more efficient when dealing with local optima; the annealing schedule allows the search to move across local regions to identify those that contain the lowest energy and then focus on these. This capability of the simulated annealing algorithm to avoid local optima, without requiring random restarts, led us to choose it as a base algorithm for our search process.

As noted before, the accuracy of the gradient-descent algorithm in minimizing an objective function within a basin of attraction that contains the solution stems from the fact that it repeatedly moves in the direction of steepest descent. It could be rewarding to combine such a local optimization technique with the simulated annealing process. The problem here is that there is no way for us to exactly calculate the gradient of the energy function. However, we can still incorporate approximate gradient information: we do this by approximating the partial derivative in each direction and choosing which dimension to move in based on proportional selection relative to these partial derivatives. We estimate the partial derivative in a dimension by evaluating two different solution points varying in that dimension

---

[9]Since we can not compute the exact gradient, this process will only approximate gradient descent.

and performing linear interpolation. We store these estimated partial derivatives for each parameter as we move through the search space and update them periodically.

What further complicates our problem is that the computational cost of evaluating the energy function is not constant, but depends on the parameters used by the model. Some parameter choices can result in very dense networks, which can take a long time to generate. This led us to introduce a method of lazy evaluation, which we call *early rejection*. The aim here is to estimate the energy at a given point early on in the generation of the network and to terminate the energy calculation and network generation if the energy estimate is too low.

We now provide more details of our adaptations to the simulated annealing algorithm and we present our final search algorithm in Algorithm 5.2.

---

**Algorithm 5.2** Our modified simulated annealing

---

$s \leftarrow$ getStartState()                                                        ▷ Starting state is chosen
$s_{best} \leftarrow$ s                                                               ▷ Store initial best solution

$k \leftarrow 0$
**while** $k < k_{max}$ **do**                                                        ▷ While there's time left
    $p \leftarrow$ getDimension()                    ▷ Preferentially choose the parameter to change
    $s', s'' \leftarrow$ neighbors($s, p, T$)              ▷ Get candidate neighbors w.r.t dimension $p$

    updateGradient($\hat{E}(s') - \hat{E}(s''), p$)                      ▷ Store the new slope for $p$

    **if** $E(s') < E(s_{best})$ **then**                           ▷ If $s'$ has the lowest energy so far
        $s_{best} \leftarrow$ s'                                          ▷ Store new best solution
    **end if**
    **if** $E(s'') < E(s_{best})$ **then**                         ▷ If $s''$ has the lowest energy so far
        $s_{best} \leftarrow$ s''                                        ▷ Store new best solution
    **end if**

    $T \leftarrow$ Temp($k/k_{max}$)
    **if** $P(\left| \hat{E}(s') - \hat{E}(s'') \right|, T)) >$ random() **then**       ▷ We are moving downhill
        $s \leftarrow$ argmin$\{\hat{E}(s'), \hat{E}(s'')\}$
    **else**                                                                      ▷ We are moving uphill
        $s \leftarrow$ argmax$\{\hat{E}(s'), \hat{E}(s'')\}$
    **end if**
    $k \leftarrow$ k+1
**end while**

**return** $s_{best}$                                                                 ▷ Return the best solution found

---

### 5.3.3  Approximate gradient-based decisions

We noted that, even though we can not calculate the gradient of the energy function directly, we can approximate it to some degree. Incorporating this information into our search algorithm could improve the performance of the algorithm, since repeatedly choosing a direction to move in using proportional selection based on exact partial derivatives would cause updates in the same expected direction as gradient-descent.

To achieve this, we choose two candidate neighbors, $s'$ and $s''$, instead of just the one in the original simulated annealing formulation. The two neighbors differ from the current state in the same dimension, that is, one parameter of the current solution $s$ is incremented by $\omega_T$ to form the neighbor $s'$ whereas the same parameter is decreased by $\omega_T$ to form $s''$. The difference in energies, scaled by the step size, $\frac{E(s') - E(s'')}{2\omega_T}$, then represents the approximate partial derivative in the dimension corresponding to the parameter that was changed. By storing these approximated partial derivatives, we can choose the parameter to alter at each timestep using proportional selection based on partial derivative estimates. Note that only one parameter's partial derivative estimate is updated each timestep. An alternative is to update all the partial derivatives in every timestep, but this is computationally too expensive. Especially in the case where the partial derivatives in one or two directions greatly exceed the others, the extra computation required to recalculate all the partial derivatives is not justified by the performance increase.

Another subtle difference from the original algorithm is that we do not allow the algorithm to remain at a state for longer than one time step. The algorithm moves to the state with higher energy with probability:

$$P(\Delta E, T) = \frac{1}{1 + \exp\left(\frac{|\Delta E|}{T}\right)} \, ,$$

otherwise it moves to the state with lower energy.[10]  Assuming the current solution's energy lies between those of the candidate neighbours, this rule boils down to proportional selection based on the probability that the original one-sided decision process of simulated annealing would have chosen the point.

---

[10]Since both $|\Delta E|$ and $T$ are positive, the probability of accepting a worse state is between 0 and 1/2. The bigger the energy difference, the smaller the probability of accepting a state with higher energy will be.

### 5.3.4   Early rejection

It is often clearly visible early on in a simulation that the current set of parameters yield a bad fit for the network.  In such a case, it is a waste of time to complete the simulation and calculate the exact energy at that state. The speed of the search process can be greatly increased by estimating the energy of a state $s$ early on in the simulation process, with the use of a *proxy energy function* $\hat{E}(s)$, and terminating the simulation if the estimated energy is too far above the energy of the best state we have encountered so far.  We refer to this process as *early rejection*.  In order to be able to move to a state even if it has been rejected early, we instead base the probability of moving to the neighboring states on estimated, rather than actual, energies.  Our simulated annealing algorithm thus searches the parameter space to optimize the proxy energy function, but along the trajectory, we calculate the actual energy function at promising locations: the proxy energy function decides which states might be candidate *best* states and, for these states, the simulation is completed and an exact energy value is calculated.

We decided to base the proxy energy function $\hat{E}(s)$ on a static snapshot of the graph after the first $n_0$ nodes joined.  There are two reasons why we chose to use a static snapshot rather than using a dynamic approach:

1. Network data at the birth of the network are usually noisy and the evolution in the initial stages is highly dependent on the methods of deployment used. Only after a while does the network 'stabilize', or lose influence from external factors.  This discrete point in the network's evolution is often easily identifiable in the network data.  By choosing $n_0$ past this *stabilization point*[11], we can get a better estimate of the energy by looking at this static snapshot of the network than by incorporating the noisy data from the birth of the network.[12]

2. Calculating the estimated energy on a static snapshot is much faster than calculating it for every graph up to that point. Since this operation is performed during every single iteration of the algorithm, this greatly reduces the computational requirements of the algorithm.

Since many simulations will possibly be rejected and $E(s)$ will not be calculated for

---

[11]For the Corporate Network (CN) and Friendship Network (FN) we use $n_0 = n/3$ and $n_0 = n/6$ respectively.

[12]Note that the noisy data are included in the temporal calculations but the effect of the inclusion thereof is minimal for a full network.

(a) Average degree                                    (b) Clustering coefficient

Figure 5.1: Contour plots over the $\gamma, \beta$ parameter space. The other parameters were fixed at: $\alpha_1 = 2.5,\ \alpha_2 = 2.5,\ w = 0.15,\ n = 1200$.

these states, we need to adapt the definition of the energy function to:

$$E'(s) = \begin{cases} \infty & \text{if } \hat{E}(s) > \hat{E}(s_{best}) \times (1 + \delta) \\ E(s) & \text{otherwise} \end{cases}$$

where $\delta > 0$ sets the threshold for the early rejection.[13]

### 5.3.5  Contour-based initialization

Through experimentation we found that areas associated with high energy often result in highly connected networks which take a very long time to process. In an effort to avoid these areas of high energy (the average degree of these networks greatly exceeds the desired degree), we considered choosing the initial parameters in such a way that the average degree and clustering coefficient of generated networks are close to their target values.

In order to choose such parameters for any desired average degree, we performed a set of simulations over the $\gamma$ and $\beta$ parameters of our model, whilst keeping the other parameters fixed. We used the characteristics of the resulting networks to construct a contour plot of the average degree as well as the clustering coefficient. These contour plots are shown in Figure 5.1. Given a network $G$ that we want to model, we then choose the initial parameters in the following way:

1. Calculate the average degree $d_G$ and clustering coefficient $CC_G$ of the original network $G$ at $n = 1,200$ nodes.

---

[13]Experimentally, we found $\delta = 0.4$ to be a good threshold.

2. From the contour plot of the average degree, extract the contour of $(\gamma, \beta)$ pairs that provides the closest match to $d_G$.

3. Using these $(\gamma, \beta)$ pairs and the contour plot for the clustering coefficient, determine which pair provides the closest match for $CC_G$.

4. Set the other parameters to their default values: $\alpha_1 = 2.5$, $\alpha_2 = 2.5$, $w = 0.15$, $n = 1200$.

## 5.4  Evaluation of our method

In the previous section we discussed our search algorithm based on the simulated annealing algorithm, but with three adaptations: contour-based initialization, early rejection and gradient-based decision making. We now give a experimental analysis of these adaptations. All experiments were performed on a Unix server with an Intel Xeon® 2.50GHz CPU and 8Gb of RAM.

### 5.4.1  Initialization



(a) Running Time  (b) Best energy found
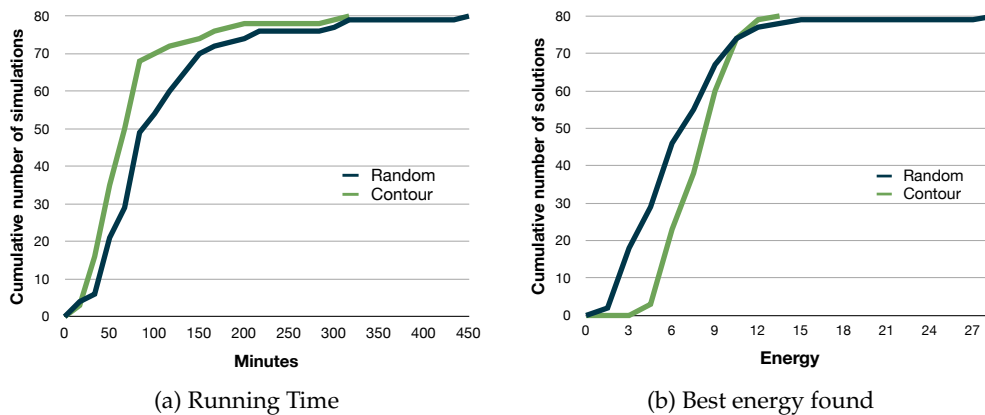
Figure 5.2: Cumulative plots of the best energy found and the running time of 80 simulations using a random initialization procedure and 80 simulations using our contour-based initialization procedure. We terminate each curve at the maximum data point in that series.

To evaluate our initialization method, we fitted our model to the Friendship Network (see Section 1.4) through two sets of 80 simulations, each consisting of a single run of our adapted simulated annealing algorithm, with 20 iterations ($k_{max} = 20$).

For the first set, we used a random initialization process and for the other 80, we used contour-based initialization. All simulations made use of both early rejection and gradient-based decisions. We recorded the running time of each simulation as well as the minimum energy obtained from the simulated annealing process. These results are summarized in Figure 5.2. As we set out to do, our initialization process reduced the average runtime of the simulated annealing process, yielding an average runtime of 66 minutes compared to the 93 minutes average runtime obtained with random initialization. The runtime of the longest simulation was also decreased by 29.7% to 304 minutes. However, random initialization achieved a somewhat better average minimum energy and an overall best energy of 1.52 which is much lower than the 3.9 obtained with the contour-based initialization. In fact, the random initialization procedure yielded 22 solutions with energy lower than 3.9.



|                      |                      |
|:--------------------:|:--------------------:|
| (a) Running Time     | (b) Best energy found |
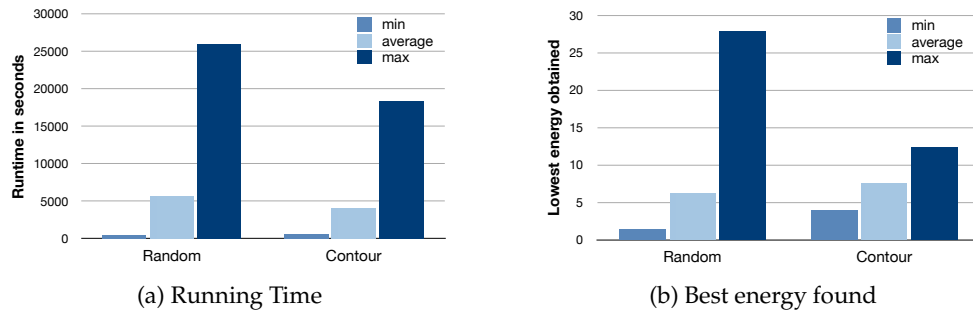
Figure 5.3: Results of 80 simulations using a random initialization procedure and 80 simulations using our contour-based initialization procedure.

An important difference between the two sets of simulations is the variance in the runtimes and energies. Our proposed initialization method reduces the amount of fluctuation in the runtimes, with a standard deviation of 53 minutes compared to the 70 minute standard deviation of the random initialization procedure. The same occurs with the minimum energies, however, with our initialization technique reducing the standard deviation from 3.95 to 2.0. These results are summarized in Figure 5.3 which clearly show that our initialization process reduces the randomness of the search. Although this is beneficial in terms of resources, it seems our proposal bounds the search space and limits exploration of some low-energy regions, which impedes the algorithm's ability to find an optimal solution. For all further simulations, throughout this study, we used random initialization, rather than our proposal, in the simulations.

### 5.4.2   Early rejection

To test whether our early rejection method reduces the amount of time required to find a good solution, we allowed two different versions of our adapted simulated annealing algorithm to complete as many simulations as possible in 40 hours. The one version used simulated annealing without early rejection and in the other we introduced early rejection. The objective was to measure how much accuracy we lose and how much time we gain with the use of early rejection. The results are shown in Table 5.1.

|                                      | No early rejection | Early rejection |
| ------------------------------------ | ------------------ | --------------- |
| Simulations completed                | 16                 | 34              |
| Average lowest energy                | 5.04               | 5.86            |
| Standard deviation of lowest energy  | 3.44               | 2.89            |

Table 5.1: The results from two 40-hour simulations, one without early rejection and the other with.

As expected, simulated annealing without early rejection yielded a better minimum energy on average. This is caused by the fact that every single solution point is considered as a candidate best solution and the trajectory of the search is guided by the actual energy function, whereas with early rejection, some solution points are thrown away without being fully analyzed and the trajectory is guided by the proxy energy function. The time gained by using early rejection is substantial though. On average, the simulations without early rejection took more than twice as long to complete, resulting in only 16 simulations completing within the time frame of the experiment, compared to the 34 simulations completed in the same time frame when using early rejection.

To determine the loss in accuracy incurred by using early rejection, we took the first 16 simulations in each set and estimated the probability that a simulation using early rejection would yield an inferior result to one not using early rejection. We did this by considering all $16^2$ pairs of simulations from different sets and calculating the proportion of pairs in which the simulation without early rejection was better than the simulation with early rejection. Using this estimator, we found the probability that using early rejection would yield inferior results to be $p = 0.68$.

Although this probability is well above the desired 50%, it must be kept in mind that the simulations using early rejection required less than half the running time. This means that it is possible to complete two simulations when using early rejec-

tion for every one simulation not using early rejection. This led us to re-perform the previous experiment, but this time allowing two simulations using early rejection for every simulation not using early rejection. We then keep only the best one of the two simulations that used early rejection and compare this set of 16 simulations to the original set of 16 simulations not using early rejection. In this case, we found the probability that early rejection would yield an inferior result to be $p = 0.39$, and the average lowest energy to be 3.4 with a standard deviation of 1.6. Thus, it is a better option to complete more simulations using early rejection than to increase the running time of every simulation by fully evaluating every single solution point. For all the simulations in this study, we used early rejection.

### 5.4.3  Approximate gradient-based decisions

Lastly, we analyzed the effectiveness of the gradient-based method we use when deciding in which dimension to move in the search space. To see whether this improves the quality of the search results, we completed two sets of 50 simulations, with each simulation consisting of a single run of the adapted simulated annealing using early rejection, with $k_{max} = 20$. For the first set, the decision of which dimension to move in was made at random. For the second set, we used our method of choosing preferentially based on the estimated partial derivative in each dimension. In Figure 5.4 we show a cumulative plot of the lowest energies obtained in the simulations.
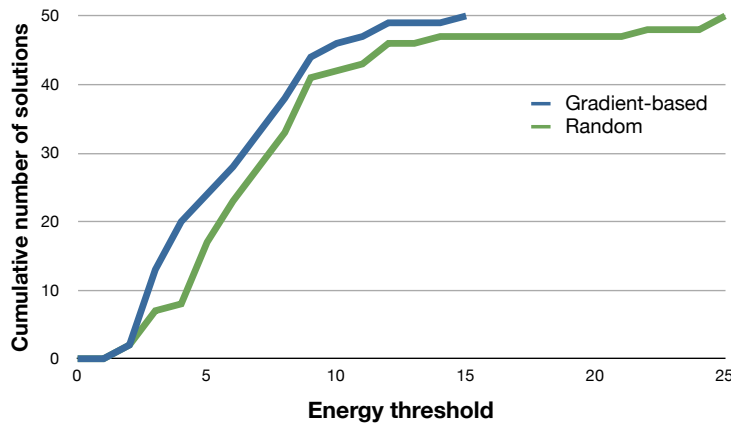


Figure 5.4: A cumulative plot of the number of solutions found with energy below certain thresholds using simulated annealing with approximate gradient-based decisions and simulated annealing with random decisions. The curves are terminated at the maximum data point in that series.

Figure 5.4 supports our belief that choosing the dimension based on approximate gradient information yields better results than choosing randomly. Our proposed decision making method improved the average energy obtained by 22%. To further quantify this improvement, we estimate the probability that approximate gradient-based decisions would benefit a simulation. We do this by constructing a binary square matrix $A$ with entry $A_{ij} = 1$ if the $i$-th simulation using gradient-based decisions found a better solution than the $j$-th simulation using random decisions. Using this estimator, we found that with probability $p = 0.59$ gradient-based decision making would improve the solution.

As expected, the gradient-based decision process reduced the running time, with the average running time for one simulation with random decision making 44% (37 minutes) longer than with gradient-based decisions. We believe this is due to the fact that gradient-based decisions cause the search process to leave high-energy regions more quickly and this reduces the running time since moving through high-energy regions, where the parameters causes the model to generate very dense networks, is very computationally expensive. Note that, with simulated annealing, the algorithm is expected to be more likely to go downhill, even with random decision making. The improvement obtained using this process is a result of the fact that the algorithm now chooses the 'steepest' slope to go down on.

Since gradient-based decisions making improves both the solutions found and the running time of the algorithm, we use gradient-based decision making in all simulations throughout this study.

## 5.5  Relationships to existing stochastic approximation techniques

Stochastic approximation methods are algorithms that attempt to optimize objective functions which can not be evaluated directly, but only estimated via noisy observations [103]. Since these methods do not use any direct gradient information, they are also said to perform *gradient-free* stochastic optimization. The classical method for gradient-free stochastic optimization is the Kiefer-Wolfowitz finite-difference stochastic approximation (FDSA) [104]. The FDSA method effectively performs a stochastic version of gradient-descent, where the gradient is directly approximated by estimating partial derivatives in each dimension.[14] The partial derivative in each dimension is estimated by linear interpolation between observa-

---

[14]This is the standard approach for approximating gradient vectors.

tions at two perturbed points[15], thus requiring $2d$ noisy evaluations of the objective function, if the search space is $d$-dimensional. An alternative is to use one-sided perturbation, which uses the current solution as the one observation in the linear interpolation, and reduces the number of required noisy evaluations to $(d+1)$.

The simultaneous perturbation stochastic approximation (SPSA) method aims at keeping the same general behavior, but reducing the number of observations. The SPSA method makes observations at only two symmetrically perturbed points - the perturbation obtained by perturbing each dimension by a random amount - and estimating the partial derivative in dimension $i$ by attributing the difference in the observations entirely to the change in the $i$-th component between the two perturbed points. This reduces the number of objective function evaluations, relative to classical FDSA, by a factor $d$. Surprisingly, in practice, SPSA and FDSA achieve comparable performance for a given number of iterations, and as such, SPSA is generally preferred over FDSA for computational reasons [104].

Note that both of these methods are, like gradient-descent, designed for local optimization. One way to try to find a global optimum, is to introduce random noise into the objective function evaluations, in order to reduce the accuracy of the estimated gradient [105]. As an alternative, both these methods could potentially be combined with simulated annealing, as we did with our approximate gradient-based decision approach. Our gradient estimation technique functions like FDSA, in which each partial derivative is calculated through a separate perturbation. In our method, however, only one partial derivative is updated per iteration, assuming that the previous estimates for the other partial derivatives are still satisfactory.[16] Also, we do not cycle through the dimensions when performing derivative updates, but rather update the derivative of the dimension chosen using proportional selection relative to the estimated partial derivatives.

An interesting consideration is whether our simulated annealing process would have benefited from simultaneous perturbation, like in the SPSA algorithm. Since the performance of SPSA is comparable to that of FDSA with regards to finding local optima, one would expect simultaneous perturbation to benefit our algorithm when the temperature is low and the algorithm is focused on local optimization. However, it is not clear how simultaneous perturbation will affect the simulated annealing algorithm's ability to move across local regions to avoid local minima. Since the SPSA (and FDSA) methods always try to find the *best* direction to move

---

[15]Two points at an identical offset from the current state, but in opposite directions.

[16]This assumption is not always valid: in particular, there is a risk of ignoring dimensions which happen to have a particularly small estimated partial derivative even when subsequent movement in the search space has made the actual partial derivative in this dimension quite large.

in, the difference in objective value between candidate solutions chosen along the approximate gradient will be large, and this might reduce the randomness of the simulated annealing algorithm, i.e. substantially reduce the probability of moving uphill at each iteration of simulated annealing.

As the relationship between our method and stochastic approximation only came to light at a very late stage in our study, it is not further examined in this work, but it certainly merits future investigation.

## 5.6   Conclusion

In this chapter, we reviewed the way parameters are usually fitted for social network models. We gave a general introduction to metaheuristics in Section 5.2 and presented the gradient-descent and simulated annealing algorithms in detail in Sections 5.2.1 and 5.2.2. In Section 5.3, we proposed a search algorithm, which can be seen as a hybrid of simulated annealing and gradient-descent using a proxy energy function. An evaluation of our proposed adaptations to simulated annealing was presented in Section 5.4. In Section 5.5, we related our model to existing stochastic approximation techniques.

# Chapter 6

# Results and discussion

In the previous two chapters, we proposed a random graph model and a method for fitting the model to real-world networks. In this chapter, we proceed to investigate the quality of this model's fit on two proprietary online social networks for which we could obtain complete historical records.[1] We compare the model's performance with two existing random graph models for social networks, focusing on a range of temporal and static characteristics.

Section 6.1 explains our evaluation methodology and motivates our choice of existing models for comparison purposes. In Section 6.2, we analyze the data sets and explain the methods we use to obtain optimal parameters for each of the models in fitting these data sets. We then proceed to analyze the relative quality of the various models' fits to the data sets with regards to average separation (Section 6.3), clustering coefficient (Section 6.4), transitivity (Section 6.5), degree distribution (Section 6.6), network densification (Section 6.7) and shrinking diameter (Section 6.8).

## 6.1  Method of evaluation

We use two real-world data sets to evaluate our model. To evaluate the performance of our model in generating networks structurally similar to real-world networks, we compare our model to two existing models. We chose the PA model (Section 3.1.3) as a benchmark because of its prominence in the literature. To investigate the relative performance our model provides over the existing top-down

---

[1]These records do not include nodes or connections that were removed from the network. Since this data is not usually available, we do not try to incorporate the removal of nodes/connections in our model.

models, we also chose the GL model (Section 3.2.1) to compare to. In the class of top-down models, the GL model and the model of Lattanzi and Sivakumar are the current state of the art, with the GL model being the only existing model that can produce networks which include isolated nodes. The model of Lattanzi and Sivakumar implies a restrictive lower bound on the degree, which was a deciding factor for us in choosing the GL model since both real-world networks contain a significant number of isolated nodes and nodes with lower degree. Another critical factor for us is the generation of complete temporal data in both the PA and GL models, which allows for a more rigorous analysis than with other state-of-the-art bottom-up models, such as Kronecker graphs (Section 3.1.9), where the number of nodes increases exponentially in the number of iterations. We base the comparison with the PA and GL models on the characteristics presented in Section 2.2.

In fitting the models to the networks, our main requirement was that the models should produce networks with the same number of nodes and approximately the same number of connections as in the real-world networks.

To obtain estimates for power-law parameters, we used the method and implementation of Clauset et al. [106] which uses maximum-likelihood fitting methods in combination with Kolmogorov-Smirnov goodness-of-fit tests.

Since all random graph models use a probabilistic process in generating networks, no two networks generated with the same model will be identical. Although the generated networks are unique, they show similar structural properties if evaluated based on the characteristics described in Chapter 2. In Table 6.4, at the end of the chapter, we present the mean value for all the characteristics of the networks generated by the various models, together with approximate 95% confidence intervals[2], calculated over 100 identical simulations for each model. The size of the confidence intervals indicate that the margin of error is negligible for the conclusions made in this chapter. As such, many of the charts presented in this chapter are based on a single run of every model, and we omit error bars for the purpose of visual clarity. In these cases, the networks we chose to represent each model were chosen in such a way that their characteristics closely resemble the mean values.

All experiments were performed on a Unix server with an Intel Xeon® 2.50GHz CPU and 8Gb of RAM.

---

[2]The confidence intervals are calculated using Student's t-distribution, which assumes that the samples are normally distributed.

## 6.2  Model parameters

Table 6.1 is an extract from Table 6.4, showing a summary of the characteristics of the real-world social networks employed in this study. Our first data set, the Corporate Network (CN), is from a proprietary corporate social network owned by a multi-national holding company. It is a closed network in which employees can connect with colleagues in other companies owned by the umbrella company. Our second network, the Friendship Network (FN), is a South African social network attracting young people through a local presence in entertainment venues.

|    | $N$ | $M$ | $\alpha(N)$ | $CC(N)$ | $AS(N)$ | $ED(N)$ | $D(N)$ | DPL |
|----|------|-------|------|--------|------|------|----|------|
| CN | 1265 | 4753 | 1.63 | 0.29 | 2.94 | 3.62 | 7 | 1.71 |
| FN | 13295 | 40679 | 1.87 | 0.021 | 4.11 | 4.89 | 11 | 1.36 |

Table 6.1: Information about the networks, showing the total number of nodes ($N$) and connections ($M$), with the estimated power-law parameter of the degree distribution ($\alpha$), the clustering coefficient ($CC$), average separation ($AS$), effective diameter ($ED$), network diameter ($D$), and densification power law (DPL) exponent.

Although both networks strongly exhibit all the properties of social networks discussed in Section 2.2, they are structurally very different. The separate companies in the CN form dense communities, whereas the FN shows a much lower degree of clustering: the CC of the CN is 14 times that of the FN. Despite this, the average degree of nodes in the FN is only 20% less than those in the CN. Modeling both of these networks accurately poses an interesting challenge, since it requires the model to show a great deal of flexibility.

In the PA model, there is only one parameter, $m_0$: the number of connections a node creates upon its entry. We use maximum-likelihood to fit the PA model to these two networks, choosing $m_0$ as half the desired average degree $k$. Because $m_0$ must be an integer, we generate networks using $m_0 = \lceil k/2 \rceil$. This causes the resulting networks to have at least the number of connections the original network has.

For the GL model, we initially used the method from the original paper, described in Section 3.2.1, to obtain parameters for the model. Initial experiments showed that these parameter values for the GL model are far from the optimal values for fitting the two real-world data sets.[3] This is due to the failure of the GL model

---

[3]Based on a comparison of average degree, CC and power-law exponent of the degree distribution.

to incorporate *bipartite clustering*, noted by the authors in the original paper. This means that the parameter estimation technique does not account for the amount of overlap between the cliques. Note that what Guillaume and Latapy call the *overlap parameter*, $\lambda$, is merely a measure of the average number of cliques a node belongs to and is *not* a measure of the overlap between these cliques. Thus, $\lambda$ measures the average degree in their bipartite graph, not bipartite clustering.

Since there is a significant amount of neighborhood overlap between the cliques in the original networks, and this is not reflected in the parameters of the GL model, the cliques in the GL-generated network are more disjoint, rarely having more than one node in common. This means that the cliques a node belongs to are more disjoint, which results in a larger number of dense communities in the social network and a bigger set of neighbors for each node. Consequently, this yields much denser networks with higher CC and average degree than the original network.

In order to find parameters for the GL model that fit the real-world data sets better, we divided the real valued domains of the parameters $\lambda \in [0, 1]$ and $\mu \in [2, 5]$ into 50 discrete values,[4] and performed a grid search for the optimal parameters. In evaluating the quality of the fit, our main criteria were producing the same number of connections as the data set, and reducing the CC of the network.[5] These were natural choices, since the failure of the GL model to reflect bipartite clustering results directly in an over-estimation of the average degree and CC. We found that for both networks, we obtain a better fit if both $\lambda$ and $\mu$ are smaller than the values estimated using the technique from the original paper. We present the original parameters, $\mu$ and $\lambda$, together with the parameters obtained through the grid search, $\mu'$ and $\lambda'$ in Table 6.2. Note that, although the maximal clique-decomposition required to calculate $\mu$ and $\lambda$ is computationally expensive, our grid-search takes even longer, and as such, provides no improvements in terms of computational requirements. We refer to the model using parameters $\lambda'$ and $\mu'$ as the GLG model.[6]

|    | $\mu$ | $\lambda$ | $\mu'$ | $\lambda'$ |
|----|-------|-----------|--------|------------|
| CN | 3.89  | 0.89      | 2.50   | 0.80       |
| FN | 2.09  | 0.83      | 1.93   | 0.79       |

Table 6.2: The two sets of parameters for the GL model as obtained by the maximal clique decomposition and grid search respectively.

---

[4]Technically, $\mu \in (0, \infty)$ but the interval $\mu \in [2, 5]$ contains both sparser and denser networks than the real-world networks, and as such we did not consider values of $\mu$ outside this interval.

[5]This corresponds to minimizing the energy function (5.1) with $w_{CC} = 2w_{deg}$ and $w_{PL}, w_{DPL}, w_{iso} = 0$.

[6]GLG is short for **GL** with **g**rid-based initialization.

For detailed results of the simulations using the different parameter sets, refer to Table 6.4. The GL and GLG models generate networks with very high clustering compared to the real-world networks, but, in both cases, the GL model produced networks with higher average degree and CC than the GLG model. To further understand why the GLG model outperforms the GL model, we performed a maximal clique-decomposition on the original networks and the GL-generated networks and plotted the maximal clique size distribution in Figure 6.1. It is clear that the average clique size in the real-world networks is much smaller than in the generated networks. In the case of the CN, the GLG model's maximal clique size distribution is much closer to the real-world network's, with a lower average clique size than the GL model. In the case of the FN, the maximal clique size distributions of both models have much longer tails than the real-world network, although the GLG's is slightly shorter. This could explain the lower degree of connectivity and clustering observed in the GLG-generated networks. We will use the GLG model for the rest of this study, since it models the real-world networks more accurately.



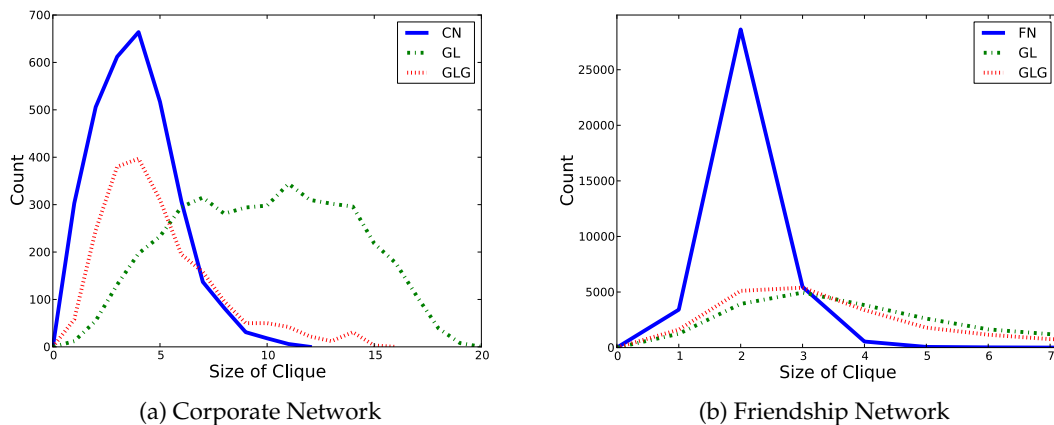(a) Corporate Network          (b) Friendship Network

Figure 6.1: The maximal clique size distributions of the real-world networks together with the GL- and GLG-generated networks.

For our model, we used our modified simulated annealing method described in Chapter 5 to find suitable parameters for generating networks that are similar to the data sets. The resulting parameters are presented in Table 6.3.

|  | $\beta$ | $\gamma$ | $\alpha_1$ | $\alpha_2$ | $w$ |
|---|---|---|---|---|---|
| CN | 0.13 | 0.83 | 1.74 | 1.85 | 1.45 |
| FN | 0.03 | 0.43 | 2.32 | 1.9 | 10.8 |

Table 6.3: Our model's parameter estimates for the CN and the FN. We sample the user activities and the commitments from power-law distributions with parameters $\alpha_1$ and $\alpha_2$ respectively.

## 6.3 Average separation

Recall from Section 2.2.1 that the average separation is defined as the mean shortest path length between a randomly chosen pair of distinct nodes in the network. In social network literature, the path length between pairs of nodes which are not connected is defined as zero, which causes this measure to be easily influenced by isolated nodes.[7] To make the measure more robust, we calculated the average separation only over the giant components of the networks. Figure 6.2 plots the average separation of the nodes in the two real-world networks and the average separation of the nodes in the generated networks. The sizes of the giant components of all the networks are also indicated in Figure 6.2.

The PA model produced a much higher average separation in both cases. With the PA model, all of the nodes are always part of the giant component since a node is always inserted with connections to already existing nodes. This results in an increase in size of the giant component, which, combined with the low clustering in the network (discussed in Section 6.4) causes the nodes to be more separated on average.

The GLG model produced lower average separations than the real-world networks. The fact that the GLG model is constructed from fully connected communities makes it intuitive that the giant component should contain a high number of nodes. The random way in which these fully connected communities overlap also causes the average distance between the nodes to be lower. This is because the loss in overlap structure creates short paths between most communities since users are more likely to form 'bridges' between communities. This was observed in both networks.

Our model matched the average separation in both networks well, with deviations of 2% and 11% respectively, although the PA model provided a slightly better match to the FN network (7% deviation). The sizes of the giant components of the

---

[7]If the path length between unconnected nodes was defined as $\infty$, networks with more than one component would not have a finite average path length.

networks generated with our model also matched the sizes of the giant components in the real-world networks closely.



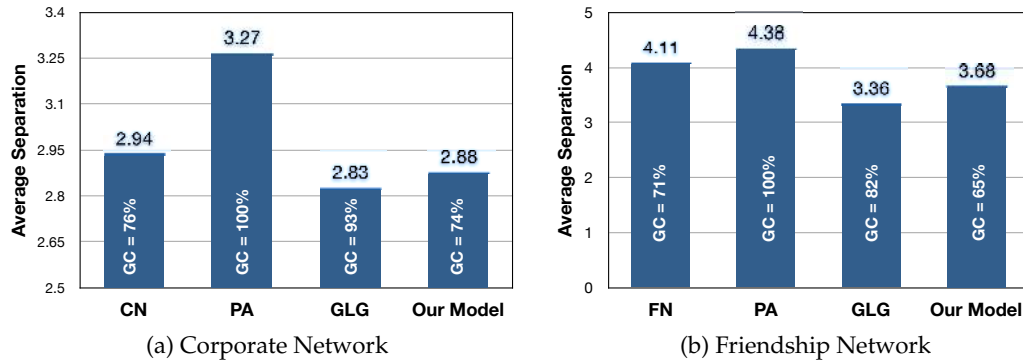(a) Corporate Network                    (b) Friendship Network

Figure 6.2: The average separation in the real-world and generated networks, together with the proportion of nodes that form part of the giant component (GC).

In examining the small-world effect, it is important to not only look at the average separation, a measure that doesn't incorparate variance, but to also compare the distribution of the path lengths. A histogram of the path lengths is plotted in Figure 6.3. Note that this plot is a non-cumulative version of the *hop-plot* used by, amongst others, Leskovec et al. [90]. We prefer this vizualization to the hop-plot, since this emphasizes the number of longer paths present in the network and also clearly shows the diameter.

Both the PA and GLG models tend to overproduce shorter paths, although the GLG model clearly outperforms the PA model. Interestingly, despite the fact that the PA model produces networks with the highest average separation, we note that the PA model is extremely unlikely to produce networks with very long paths, since each node is connected to at least $m_0$ other nodes. This causes medium length paths to exist between most pairs of nodes with high probability, which causes the path-length distribution to not be sufficiently right-skewed (refer to Figure 6.3). Our model matches the networks noticeably better than the other two models, but slightly overproduces longer paths.

To further understand the why we observe these different path length distributions for the different models, we performed a maximal clique decomposition, and plotted the maximal clique size distribution in Figure 6.4. This figure suggests that the PA model fails to include shorter paths because of its inability to construct larger cliques. The GLG model, on the other hand, tends to construct many larger cliques, which explains the short average path lengths observed. Our model lies

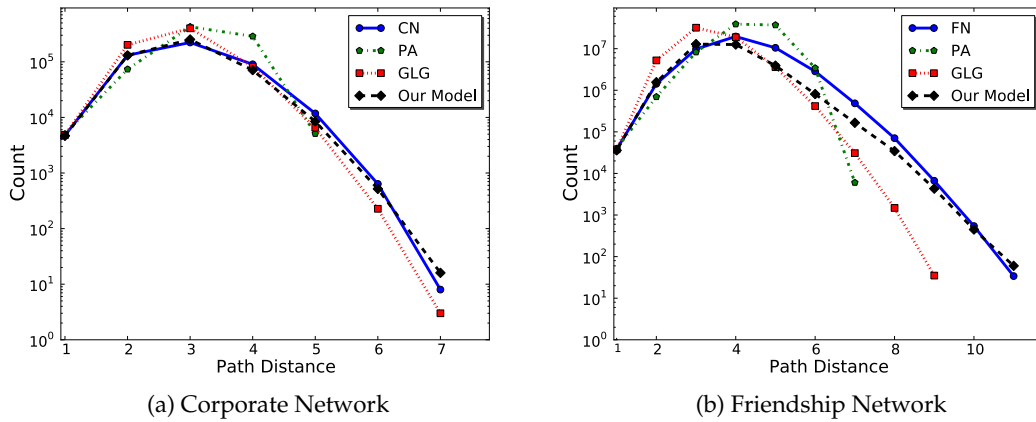(a) Corporate Network                        (b) Friendship Network

Figure 6.3: Pairwise distance histograms of the real-world and generated networks.

between these two extremes, and matches the real-world distributions better, although our model still overproduces larger cliques which explains why, in both cases, our model's average separation was lower than desired.



(a) Corporate Network                        (b) Friendship Network
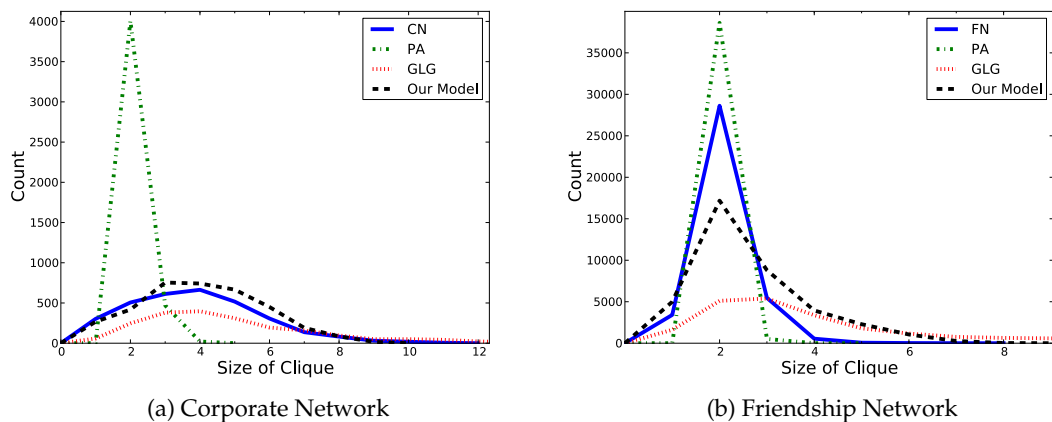
Figure 6.4: The maximal clique size distributions of the real-world and generated networks. The tail for the GLG model's distribution is not fully shown. The largest maximal cliques obtained with the GLG model was 15 and 22 for the CN and the FN respectively.

## 6.4  Clustering coefficient

Figure 6.5 shows the evolution of the CCs of the two networks and the fitted models. The CN shows an initial increase in clustering and starts to stabilize for $n > 400$.

For the FN, the increasing trend still seems to be continuing. Our model is the only model to capture this initial growth period of the networks in which the clustering increases with network size.



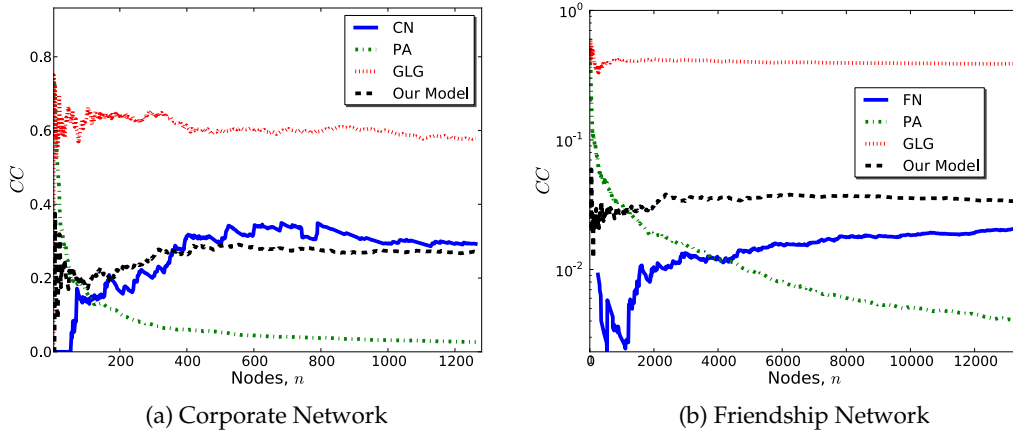(a) Corporate Network  (b) Friendship Network

Figure 6.5: Evolution of the CCs of the real-world and generated networks. Note that due to the high CC of the GLG compared to the low CC of the FN, we are forced to use a logarithmic y-axis on the right.

The PA model yields a CC with an almost exponential decrease, a trend that is clearly not observed in the true data. For the highly clustered CN, the final CC yielded by the PA model is about ten times lower than desired. This is due to the fact that the average degree in the PA model is constant during the development of the network, which restricts the formation of triangles. This is further discussed in Section 6.5.

The GLG model, on the other hand, generates a more clustered network with a CC double that of the CN. Our model fits the CC well throughout the evolution of the network.[8]

The FN shows an extremely low degree of clustering compared to the CN, although its CC is still about 60 times that of a purely random Erdős-Rényi graph. Once again, the PA model shows an exponential decrease with a final CC that is a fraction of the real network's. On average, the GLG model exceeds the real network's CC by a factor of 20. Our model matches the evolution of the CC more accurately, with about double the amount of clustering than present in the real network.

The results of the GLG model on the FN exhibit the incapability of top-down models using the deterministic flattening rule to generate social networks with low CCs

---

[8]Apart from the initial noisy phase ($n < 340$), our model is consistently within 20% of the true value.

and high average degree. The reduced flexibility imposed by the deterministic flattening rule seems to be a serious limitation. In the case of the FN, which shows a CC one fourteenth that of the CN, the GLG model's CC only decreased by a factor of 0.25. This demonstrates that the existing top-down models are more suited to modeling highly clustered networks. Our model, on the other hand, composes the network of quasi-cliques which allows the model to produce communities with lower density. Our model showed much greater flexibility, with the CC in the case of the FN being a factor 8 smaller than in the case of the CN. This being said, our model still produces networks which are too clustered, although the fact that the CC is the same order of magnitude as the real-world networks' suggests that our probabilistic flattening rule is a step in the right direction.

## 6.5   Transitivity

Transitivity refers to the tendency of triangles to form in a network. The most popular measure of transitivity is, indeed, the clustering coefficient analyzed in the previous section, but in this section, we do a more in-depth analysis of the degree of triangle participation in the network. A popular visualization, recently used by Leskovec et al. [41], is the node triangle participation graph, discussed in Section 2.2.4. We present the node triangle participation graphs for the two networks in Figure 6.6.



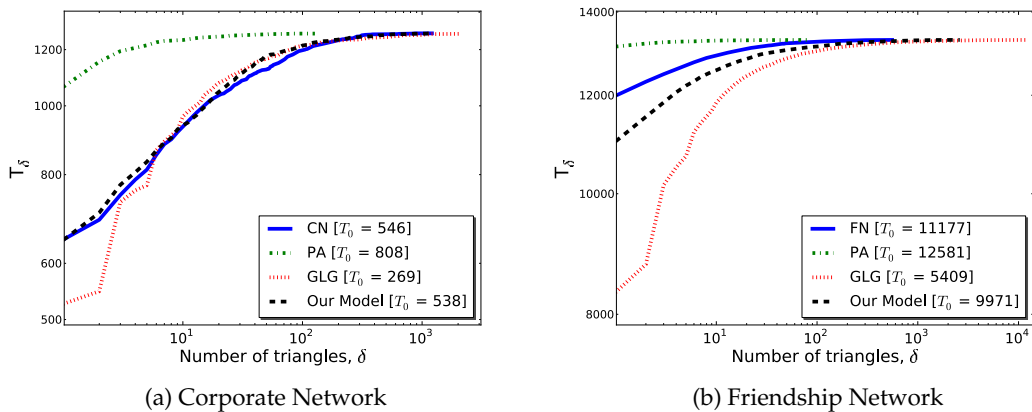(a) Corporate Network          (b) Friendship Network

Figure 6.6: Node triangle participation plots for the real-world and generated networks. Since we use a log-scale, $T_0$ can not be included in the graph, so these values are shown in the legend. Note that these plots are based on the static, fully evolved networks.

The PA model failed to produce the long tail observed in the real-world networks, with almost no nodes participating in more than 10 triangles and a large percentage of nodes not participating in any triangles. This very small number of triangles explains the low CC observed in Section 6.4. Whenever a new node is inserted, it connects to $m_0$ other nodes, resulting in a constant average degree. This causes the transitivity in the network to decrease as the network grows. This is due to the fact that a triangle can only form if a new node chooses two already connected nodes to connect to. The probability of this happening decreases as the network size increases, since nodes are more likely to connect to older nodes (due to the age-degree correlation), regardless of the social distance between these older nodes. Furthermore, since the probability of a node being part of a triangle decreases as nodes are added to the network, the probability of a node forming part of multiple triangles decays exponentially.

The GLG model exhibited opposite behavior, failing to include enough nodes that have low triangle-involvement. This directly correlates with the observed high CC in Section 6.4. This is very intuitive, since a node's participation in a community of size $k$ under this model means that it will be part of a $k$-clique in the resulting social network, guaranteeing that it will be part of at least $\binom{k-1}{2}$ triangles in that single community. The result is that there is a direct correlation between the average degree and the transitivity in the resulting social network, which explains the limited flexibility of the model.

Our model provided the most accurate match to the real-world networks in both cases, with a slightly lower level of transitivity than the CN and a slightly higher level of transitivity than the FN. It is evident that our model breaks the correlation between degree and transitivity which the GLG model is subject to. Our model also closely matches the number of nodes not involved in any triangles, with deviations of 2% and 11% from the true values of the CN and FN respectively. This is a major improvement over the GLG, which deviates by more than 50% from the true value in both cases.

## 6.6 Degree distribution

Figure 6.7 shows the degree distributions of the fully evolved real-world and generated networks. Recall from Section 2.2.5 that a truncated power-law distribution takes the form

$$P(k) \propto k^{-\alpha} \quad \text{for } k > k_{min} .$$

A log-log plot of such a distribution should thus be roughly linear with slope $-\alpha$. All three models produce power-law degree distributions. However, the minimum degree $m_0$ of the PA model is very high compared to the real-world networks and the maximum degree is very low, i.e. the distribution has a much shorter tail than that of the real-world networks. This can be observed by the steeper downward trend for the PA model than for the real-world networks.[9]

Our model yields a close match for both networks, with the GLG model also providing a better fit than the PA model, but with more deviation from the real-world networks than our model. From Figure 6.7, it can be seen that the GLG model overproduces nodes with degree in the range $d < 10$ and underproduces nodes with degree in the range $d > 10$. This can be viewed as a consequence of the correlation of nodes' degrees within local communities. For the GL model to produce a node with higher degree, this node has to be a part of a larger community which, when translated into a clique, will result into higher degrees for all other nodes in that community. This causes an increase in average degree, and since we only considered graphs with an accurate average degree, the chosen graph can not include such a large number of nodes with high degree.

Once again, this proves the value of our quasi-clique approach. Within local communities, some nodes are able to obtain high degrees, based on their high commitments to those communities, and this has no direct influence on the degrees of other nodes in those communities.

Since we are interested in the development of the network, we also want to examine the evolution of the power-law parameter, $\alpha$, as the network develops. We used the method of Clauset et al. [106] to analyze the degree distributions. This method estimates the truncation threshold, $k_{min}$, using a maximum-likelihood estimator $\widehat{k}_{min}$. For both the real-world networks, their method yields $\widehat{k}_{min} = 1$. Since the value of $\alpha$, estimated from a network, depends heavily on the value of $k_{min}$ used in the calculation, we decided to perform an initial experiment in which we use the same value of $k_{min} = 1$ for all the generated and real-world networks. The evolution of $\alpha$ is presented in Figure 6.8.

In both real-world networks, $\alpha$ decreases before stabilizing, although in the CN, $\alpha$ shows a slight increase towards the end. Thus, both networks start out with a degree distribution with a shorter tail, and this tail gradually grows before $\alpha$ stabilizes. None of the models replicate this behavior. Our model also shows a slight initial decrease in $\alpha$, although this decrease is not as noticeable as in the real-world

---

[9]A known issue with the PA model is that it can only produce networks for which the degree distribution has a power-law parameter close to 3 [59].

(a) Corporate Network



(b) Friendship Network
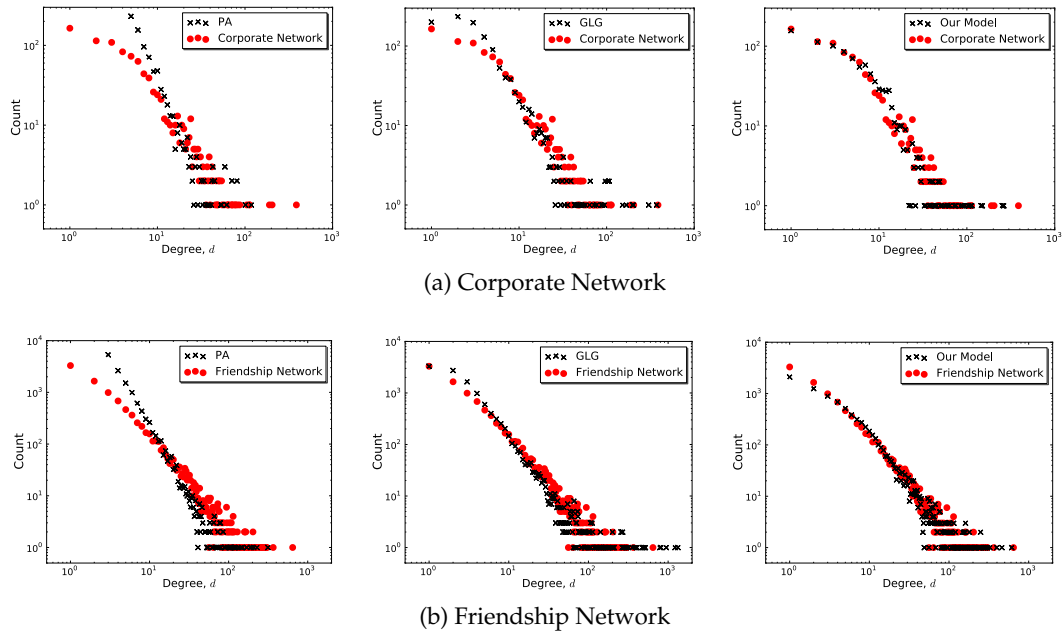
Figure 6.7: Log-log plots of the degree distribution of the real-world and generated networks. The plots are based on the static, fully evolved networks.



(a) Corporate Network          (b) Friendship Network
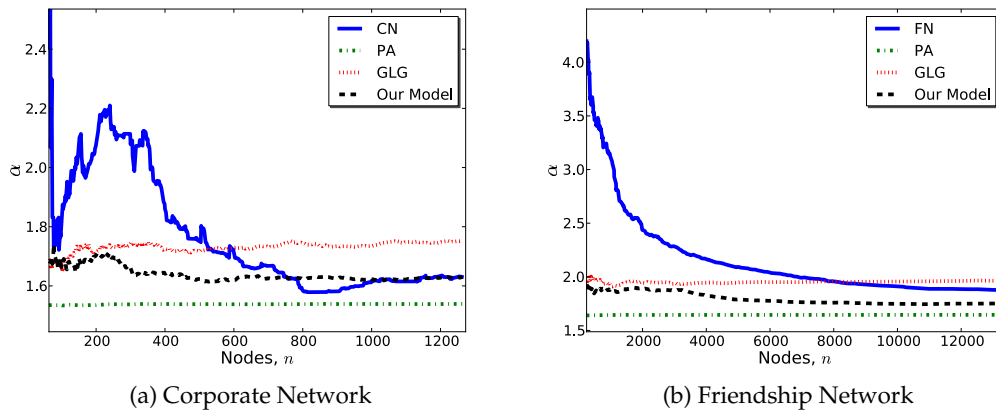
Figure 6.8: Evolution of the power-law parameters of the real-world networks compared to those of the three models, using the fixed value $k_{min} = 1$ to estimate the power-law exponent for each network.

networks. The GLG model does not show this trend, yielding a power-law parameter which increases somewhat, although the model's final degree distribution closely matches that of the FN network. Our model deviates from the evolution of the power-law parameter of the CN for $n < 600$, and thereafter provides the best fit to the data.

It is important to note that $\widehat{k}_{min} = 1$ is not the optimal estimate for all of the models. In Figure 6.9, we plot the evolution of the power-law coefficient using the estimated value, $\widehat{k}_{min}$ for each model.[10] Our model is the only model for which $\widehat{k}_{min} = 1$, as in the real-world networks. This means that both the PA and the GLG models deviate from power-law behavior for the lower degrees, as was seen in the static snapshot in Figure 6.7. With the optimal estimates of $\widehat{k}_{min}$, both the PA and GLG models fit power-law distributions with shorter tails than the real-world networks.



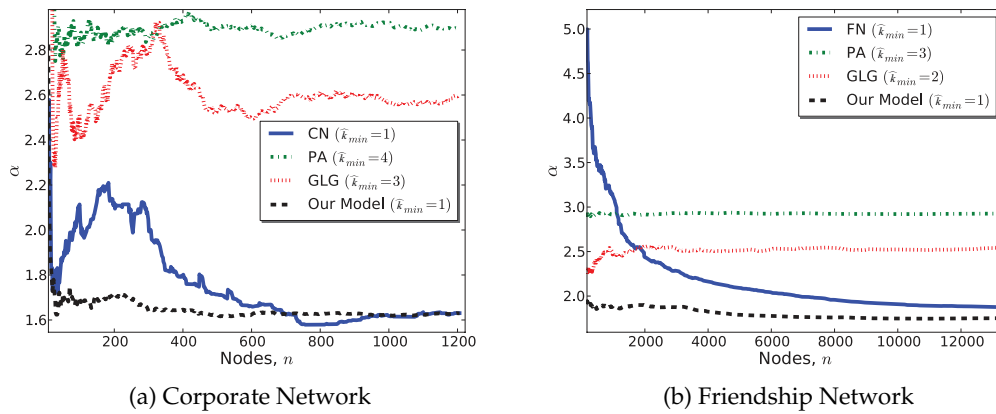|  |  |
|:--:|:--:|
| (a) Corporate Network | (b) Friendship Network |

Figure 6.9: Evolution of the power-law parameters of the real-world and generated networks.

## 6.7 Network densification

In Section 2.2.6, we discussed the DPL exponent, a measure based on the observation that the number of connections in a social network typically scale superlinearly in the number of nodes, i.e. for some $1 < \rho \leq 2$, we have $m_t \propto n_t^\rho$. In Figure 6.10, we plot the number of connections against the number of nodes in the real-world networks and the fitted models. For each network, we also indicate the network's densification exponent, $\rho$, which we estimate using a least-squares fit.[11] In the CN, we see strong superlinear growth for $n < 800$, after which the curve suddenly flattens. This could be due to external factors (e.g. marketing) and it causes the estimate of $\rho$ to be higher than we expect. Although much lower, our model provides

---

[10]Since the value of $\widehat{k}_{min}$ changes as the network evolves, we choose the median value as our estimate. In all cases, the median occurred with high frequency.

[11]We use linear regression to fit a straight line to the log-log plot of the data. Since $m_t = kn_t^\rho \implies \log m_t = \log k + \rho \log n_t$, $\rho$ can be estimated by calculating the slope of the line.

the strongest superlinear growth of the three models, with the PA model yielding linear growth by its design.

The DPL exponent of the FN is $\rho = 1.36$. The GLG model shows an almost linear trend on the FN, so that our model is the only one to replicate noticeable densification in this case, although at a slower rate than the real-world network. This could be explained by the observation made in Section 6.6 that our model is the only model that does not underproduce nodes with degree in the range $d \in [10, 100]$. As the network grows, the average degree in the network needs to rise, and if the model underproduces nodes with degree in this range, the average degree will increase at a slower rate.



(a) Corporate Network



(b) Friendship Network
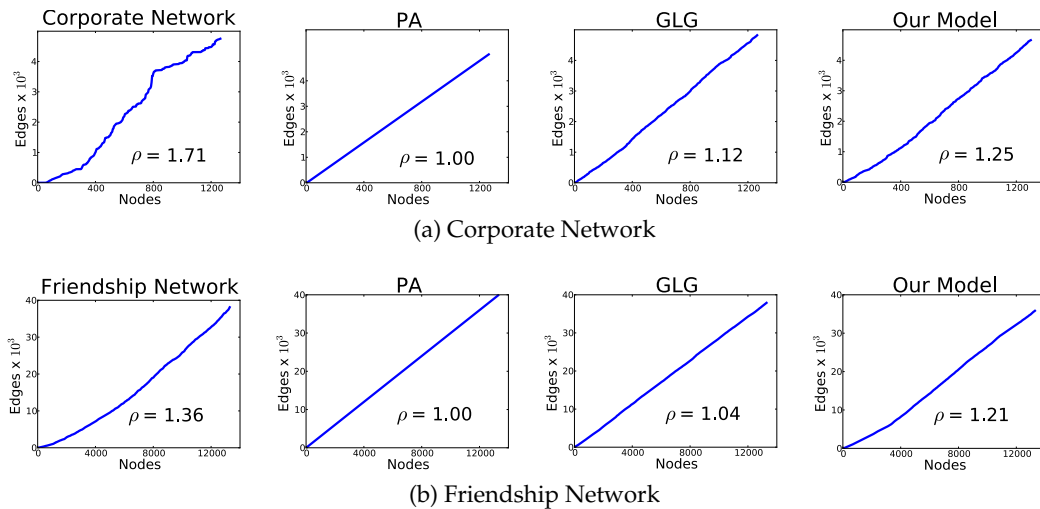
Figure 6.10: Log-log plots of the number of connections vs the number of nodes, together with the densification exponent for the real-world and generated networks.

## 6.8  Shrinking diameter

Section 6.3 discussed the average separation of the real-world and generated networks. Recently, Leskovec et al. [41] also analyzed the change in network diameter over time and found that a number of current real-world networks show a decreasing trend in effective diameter (see Section 2.2.2). We show the evolution of the diameters and effective diameters of the real-world and generated networks in Figure 6.11.

The diameters of the two real-world networks behave differently, the CN showing an increase over time and the FN a decrease. However, with the effective diameter, the trend is similar for both networks: they show an early decrease in effective

Figure 6.11: The full diameters and effective diameters of the real-world and generated networks. The exact values of the diameter and effective diameter for each network is shown in Table 6.4.
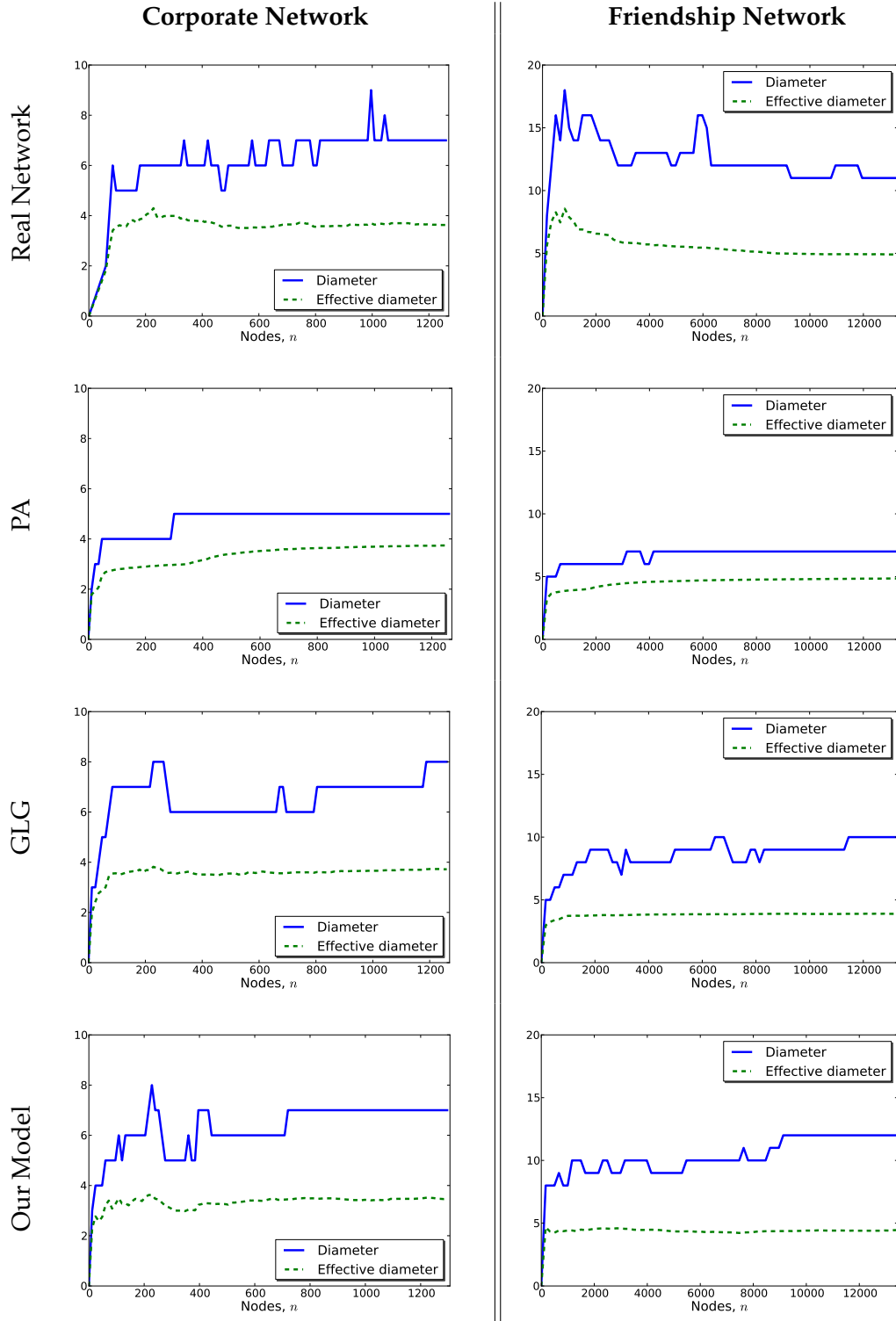
diameter, and then seem to stabilize for $n > 600$ in the CN and for $n > 8000$ in the FN. All three models provide a reasonably good match for the effective diameter of the CN but in the case of the FN both our model and the GLG model produce a lower effective diameter than the real-world network. The decreasing effective diameter of the FN for $700 < n < 3000$ is not observed in any of the models, with all three models showing a more stable initial effective diameter.

Interestingly, the PA model shows very few fluctuations in diameter with a slight increasing trend. Bollobás and Riordan [107] showed analytically that in the case of the PA model, the diameter increases almost logarithmically:

$$D(n) \approx \log n / \log \log n.$$

There is also little difference between the actual and the effective diameter of the PA model. This is a consequence of the high minimum degree. Every node can be reached through at least $m_0$ different neighbors, making it highly unlikely that a degenerate configuration (like a chain of nodes) will cause the diameter to greatly exceed the effective diameter. However, it is clear that both real-world networks are subject to such fluctuations in the diameter, and our model and the GLG model better incorporate this behavior.

## 6.9   Conclusion

In this chapter we compared our model to two existing random graph models by analyzing the capability of the models to duplicate various characteristics of two real-world social network data sets. In Section 6.3 we found that our model is the only one to accurately match the larger separation and longer path lengths present in the real-world networks. In Sections 6.4 and 6.5 we saw that our model improves on the GLG model by relaxing the dependency between average degree and clustering, enabling the generation of networks with high average degree and low levels of clustering and transitivity. A detailed static and temporal analysis of the degree distribution of the models was given in Section 6.6, finding that our model provides a natural match to the degree distributions of the real-world networks. In Section 6.7 we considered network densification and found that our model better replicates the superlinear growth in connections that is observed in the real-world networks. Section 6.8 showed that the initial decreasing trend in effective diameter of both real-world networks is not observed in any of the models.

One has to keep in mind that our model is more complex than the other two models and takes five parameters compared to the two parameters of the GLG model

and the single parameter of the PA model. Because of this, one would expect our model to provide a better fit to the real-world networks. Throughout this analysis, it has been evident that the two real-world networks show very different structural properties, and we are convinced that the amount of flexibility required to model both of these networks accurately justifies the use of an expanded parameter set. This is evident by the fact that the PA model can only produce networks with very similar, short-tailed degree distributions, and the GLG model can only produce highly clustered, highly transitive networks. By introducing a few extra parameters, we are able to overcome both these restrictions by increasing the flexibility of the model, specifically with regard to the connection density within communities. Regretfully, the exact extent of the required flexibility of such a model is still not clear to us, since we only had access to two complete data sets.

|          | **CN**  | PA             | GL              | GLG             | Our Model        |
|----------|---------|----------------|-----------------|-----------------|------------------|
| $N$      | **1265**| 1265           | 1265            | 1265            | 1265             |
| $M$      | **4753**| 5040           | $9099 \pm 670$  | $4826 \pm 322$  | $4516 \pm 811$   |
| $CC(N)$  | **0.29**| $0.03 \pm 0.005$| $0.78 \pm 0.02$| $0.58 \pm 0.033$| $0.27 \pm 0.025$ |
| $\alpha(N)$ | **1.63**| $2.92 \pm 0.045$| $1.53 \pm 0.02$| $1.75 \pm 0.034$| $1.63 \pm 0.07$ |
| $AS(N)$  | **2.94**| $3.27 \pm 0.043$| $2.33 \pm 0.05$| $2.83 \pm 0.09$| $2.88 \pm 0.12$ |
| $ED(N)$  | **3.62**| $3.73 \pm 0.03$| $2.75 \pm 0.03$ | $3.26 \pm 0.2$  | $3.47 \pm 0.25$  |
| $D(N)$   | **7**   | $5.02 \pm 0.28$| $5.02 \pm 0.6$  | $6.73 \pm 1.43$ | $7.5 \pm 1.44$   |
| $DPL$    | **1.71**| 1.0            | $1.12 \pm 0.04$ | $1.12 \pm 0.055$| $1.25 \pm 0.15$  |

(a) Corporate Network

|          | **FN**   | PA     | GL     | GLG   | Our Model |
|----------|----------|--------|--------|-------|-----------|
| $N$      | **13295**| 13295  | 13295  | 13295 | 13295     |
| $M$      | **38112**| 39873  | 46559  | 37895 | 36845     |
| $CC(N)$  | **0.021**| 0.004  | 0.47   | 0.39  | 0.034     |
| $\alpha(N)$ | **1.87**| 2.92  | 1.85   | 1.96  | 1.75      |
| $AS(N)$  | **4.11** | 4.38   | 3.04   | 3.36  | 3.68      |
| $ED(N)$  | **4.89** | 4.85   | 3.56   | 3.90  | 4.44      |
| $D(N)$   | **11**   | 7      | 8      | 9     | 12        |
| $DPL$    | **1.36** | 1.0    | 1.048  | 1.04  | 1.21      |

(b) Friendship Network

Table 6.4: A detailed comparison of the real-world and generated networks, based on the total number of nodes ($N$) and connections ($M$), with the power-law parameter ($\alpha$), clustering coefficient ($CC$), average separation ($AS$), effective diameter ($ED$), network diameter ($D$), and DPL exponent ($DPL$). For the CN, we show 95% confidence intervals for the characteristics calculated over 100 identical simulations for each model. Since the scale of these intervals are insignificant in terms of the conclusions made, and the variance observed in the FN is smaller than the CN, these are omitted for the FN.

Chapter *7*

# Conclusion

In this study, we presented a dynamic top-down model that uses a probabilistic process to convert the bipartite network into a social network. This is the first such top-down model, with the other models all using a deterministic flattening rule. We compared our model to two existing models and found that the probabilistic flattening rule of our model provides more flexibility and was more accurate at modeling two real-world data sets. We present a summary of the results in Section 7.1. In Section 7.2 we summarize the contributions made through the course of the study and in Section 7.3 we discuss possible extensions of this work.

## 7.1  Summary of investigation and results

Recently, a new family of random graph models for social networks, called top-down models, have become popular. All of the existing top-down models, however, show limited flexibility and assume that each community will result in a clique over its members in the social network. We investigated a probabilistic flattening rule, allowing for variable connection density within communities in the social network. In order to do so, we introduced a commitment value for each user-community pair. When determining the probability of two users connecting, these commitment values for mutual communities are used. Our model also explicitly incorporates bipartite clustering, or the tendency of communities to overlap, by choosing users' new communities from the set of communities that overlap with their existing communities. No existing models include bipartite clustering to this extent.

An important aspect of a social network model is its effectiveness in reproduc-

ing real-world characteristics. To fit our model to real-world networks we used a stochastic optimization metaheuristic that can be seen as a hybrid of simulated annealing and gradient-descent using approximate partial derivatives. We compared our model's performance with both the bottom-up PA model of Barabási and Albert [34] and the top-down GL model of Guillaume and Latapy [27].

Our experimental results suggest that the PA model is unable to produce realistically high levels of clustering and transitivity. Furthermore, because of its high minimum degree and low flexibility, the PA model can only produce networks with very similar degree distributions and proved to be unable to model our real-world networks.

We found that the GL model's failure to include bipartite clustering results in a much lower average separation and diameter in the generated networks than in the real-world networks. The random overlap structure between communities causes nodes to be more likely to form bridges between communities, and this, combined with the number of large cliques present, shortens the paths between distant nodes. We observed that the GL model's deterministic flattening rule can only generate highly clustered and highly transitive networks, with shorter effective and actual diameters than the real networks.

Our model's inclusion of bipartite clustering allows for the creation of longer paths and larger diameters. Our model is the only model able to generate networks with low average separation and networks with high average separation. Through our model's ability to include sparser communities in the network, it showed major improvements over the GL model, accurately modeling the clustering and transitivity of both the real-world networks, even though the clustering in the one real-world network was orders of magnitudes higher than in the other. The introduction of commitment values between users and communities in our model allow some nodes in a community to have high degrees based on their high commitments to the community, without directly influencing the degree of any other nodes in the community. This allows for more accurate reproduction of real-world degree distributions.

Although our model is more complex than the existing top-down models, we believe that this additional complexity is justified by our results: it is the first top-down model that can accurately reproduce a variety of important social network characteristics, namely average separation, clustering, degree distribution, transitivity and network densification, simultaneously.

## 7.2   Contributions

In pursuing the objectives of this study, the following contributions have been made:

- We proposed a probabilistic flattening rule that makes use of users' commitments to mutual communities to determine the probability that they will be connected. This flattening rule is the first non-deterministic flattening rule that we are aware of for top-down models.

- In order to formulate the probabilistic flattening rule of our model, we introduced the first use of a weighted bipartite network to model affiliations between users and communities. We refer to the weights of the connections between users and communities as *commitments*. Each user is assigned a commitment value for every community it belongs to and these values allow for variation of the degree of community members.

- We introduced a method for building a bipartite community structure in such a way that bipartite clustering is modeled: users choose new communities to connect to from communities that overlap with their existing communities.

- We presented an analysis of two temporal data sets of current online social networks, comparing our model to two existing social network models.

## 7.3   Future work

In our model, each user is assigned an activity value and for each community they are a part of, they are assigned a commitment value. All these activity and commitment values are constant for the duration of the network generation. This is not true to real-world behavior where people change their behavior over time, going through periods of higher and lower activity on social networks. This behavior could be incorporated in the model by treating activity and commitments as values that change over time.

Our flattening rule, based on a combination of the two users' commitments to their mutual communities is the first probabilistic flattening rule, and although this flattening rule experimentally improved upon the deterministic flattening rule used by existing models, many different forms of this function could be examined.

During the generation of a social network using our model, no nodes or connections are ever removed from the network. In real-world networks, both nodes and connections can be removed. Due to the unavailability of data for removed entities,

we did not attempt to include this in the model.  Should such data become available, removal of nodes and/or connections could be incorporated in the model.

The method we use for parameter estimation uses an energy function (defined in equation 5.2) that is not symmetric around the target value.  Using this energy function, overestimates are able to achieve higher energies than underestimates.  Experimentally, this energy function worked better than some symmetrical versions we tried, since overestimates usually correspond to denser networks, which take longer to process.  Further investigation could be done in order to find a symmetrical form of energy function which still avoids denser regions in the search space.

Our stochastic optimization metaheuristic shows very strong similarities to methods from the field of stochastic approximation, in particular the FDSA and SPSA methods.  A better theoretical understanding of our approach and a more thorough comparison of our approach to these models, could help identify an alternative method for parameter estimation, which could reduce the computational requirements for selecting parameters or even increase the accuracy of the model when fitted to real-world social networks.

As noted, the experimental success of our model is qualified by the fact that we only had two real-world data sets to evaluate it on. The model could also be tested more extensively on other data sets, should they become available.

To date, top-down models have only been able to model highly clustered networks. Our model enables networks with lower levels of clustering to also be modeled using top-down models.  However, as observed on the FN, our model could still do better at generating unclustered networks with high degrees. Any future work on the model should aim towards this property, which will further broaden the general applicability of the model.

# Bibliography

[1]     Facebook. Retrieved September 2011.
         Available at: `http://www.facebook.com`

[2]     Twitter. Retrieved September 2011.
         Available at: `http://www.twitter.com`

[3]     Twitter co-founder Jack Dorsey rejoins company. Retrieved September 2011.
         Available at: `http://bbc.co.uk/news/business-12889048`

[4]     Facebook says membership has grown to 750 million.     Retrieved
         September 2011.  Available at:     `http://usatoday.com/tech/news/`
         `2011-07-06-facebook-skype-growth_n.htm`.

[5]     UniversalMcCann: Wave 5 - the socialisation of brands.   2009.   Re-
         trieved September 2011. Available at: `http://universalmedia.nl/files/`
         `Wave_5-The_Socialisation_Of_Brands-Report.pdf`.

[6]     Kasper, C. and Voelkl, B.: A social network analysis of primate groups. *Primates*, 2009.
         Available at: `http://dx.doi.org/10.1007/s10329-009-0153-2`

[7]     Freeman, L.C.: *The Development of Social Network Analysis: A Study in the Sociology of Science.* Empirical Press, 2004. ISBN 9781594577147.

[8]     Granovetter, M.S.: The strength of weak ties. *The American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, Aug 1973.

[9]     Eubank, S., Anil Kumar, V.S. and Marathe, M.: Epidemiology and wireless communication: Tight analogy or loose metaphor?  *Bio-Inspired Computing and Communication: First Workshop on Bio-Inspired Design of Networks,*

*BIOWIRE 2007 Cambridge, UK, April 2-5, 2007 Revised Selected Papers*, pp. 91–104, 2008.

[10] Vélez, M., Ospina, J. and Hincapié, D.: Tutte polynomials and topological quantum algorithms in social network analysis for epidemiology, biosurveillance and bio-security. In: *BioSecure '08: Proceedings of the 2008 International Workshop on Biosurveillance and Biosecurity*, pp. 74–84. Springer-Verlag, Berlin, Heidelberg, 2008. ISBN 978-3-540-89745-3.

[11] Mayer, A.: Online social networks in economics. *Decision Support Systems*, vol. 47, no. 3, pp. 169–184, 2009. ISSN 0167-9236.

[12] Granovetter, M.: Economic action and social structure: The problem of embeddedness. *The American Journal of Sociology*, vol. 91, pp. 481–510, Dec 1985.

[13] Ioannides, Y.: Random graphs and social networks: An economics perspective. *Unpublished manuscript, Tufts University*, 2006. Retrieved September 2011.
Available at: `http://tufts.edu/~yioannid/IoannidesRandom_Graph_Soc_Net3_MIT.pdf`

[14] Chorley, R.J. and Haggett, P.: *Network Analysis in Geography*. London, 1969.

[15] Erman, N.: Citation analysis for e-government research. In: *dg.o '09: Proceedings of the 10th Annual International Conference on Digital Government Research*, pp. 244–253. Digital Government Society of North America, 2009. ISBN 978-1-60558-535-2.

[16] Otte, E. and Rousseau, R.: Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, vol. 28, no. 6, pp. 441–453, May 2006.

[17] Yang, C.C. and Sageman, M.: Analysis of terrorist social networks with fractal views. *Journal of Information Science*, vol. 35, no. 3, pp. 299–320, 2009. ISSN 0165-5515.

[18] Milgram, S.: The small world problem. *Psychology Today*, vol. 2, pp. 60–67, 1967.

[19] Google plus. Retrieved September 2011.
Available at: `http://plus.google.com`

[20] Erdös, P. and Rényi, A.: On random graphs. *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.
Available at: `http://renyi.hu/~p_erdos/1959-11.pdf`

[21] Newman, M.E., Watts, D.J. and Strogatz, S.H.: Random graph models of social networks. *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 2566–2572, February 2002. ISSN 0027-8424.
Available at: `http://dx.doi.org/10.1073/pnas.012582999`

[22] Watts, D. and Strogatz, S.: Collective dynamics of 'small-world' networks. *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[23] Hallinan, M.T. and Smith, S.S.: Classroom characteristics and student friendship cliques. *Social Forces*, vol. 67, no. 4, pp. 898–919, 1989. ISSN 00377732.

[24] Pang, S., Chen, C. and Wei, T.: A realtime clique detection algorithm: time-based incremental label propagation. In: *IITA'09: Proceedings of the 3rd International Conference on Intelligent Information Technology Application*, pp. 459–462. IEEE Press, Piscataway, NJ, USA, 2009. ISBN 978-1-4244-5212-5.

[25] Bianconi, G. and Marsili, M.: Emergence of large cliques in random scale-free networks. *Europhysics Letters*, vol. 74, no. 4, pp. 740–746, 2006.

[26] Guillaume, J. and Latapy, M.: Bipartite structure of all complex networks. *Information Processing Letters*, vol. 90, no. 5, pp. 215–221, 2004.

[27] Guillaume, J. and Latapy, M.: Bipartite graphs as models of complex networks. *Physica A: Statistical Mechanics and its Applications*, vol. 371, no. 2, pp. 795–813, 2006.

[28] Orlin, J.: Contentment in graph theory: covering graphs with cliques. In: *Indigationes Mathematicae*, pp. 468–474. ACM, 1977.

[29] Srihari, S., Ng, H., Ning, K. and Leong, H.: Detecting hubs and quasi-cliques in scale-free networks. *Proceedings of the 19th International Conference on Pattern Recognition*, pp. 1–4, 2008.

[30] Abello, J. and Resende, M.G.C.: Massive quasi-clique detection. *Proceedings of the 5th Latin American Symposium on Theoretical Informatics*, Mar 2003.
Available at: `http://citeseer.ist.psu.edu/572004`

[31] Newman, M.E.J.: Random graphs with clustering. *Physical Review Letters*, vol. 103, p. 058701, Jul 2009.

[32] Pool, I. and Kochen, M.: Contacts and influence. *Social Networks*, vol. 1, pp. 1–48, 1978.

[33] Gaure, J.: Six degrees of separation: A play. *New York: Random House*, vol. 1, 1990.

[34] Barabási, A. and Albert, R.: Emergence of scaling in random networks. *Science*, vol. 286, no. 5439, p. 509, 1999.

[35] Amaral, L., Scala, A., Barthelemy, M. and Stanley, H.: Classes of small-world networks. *Proceedings of the National Academy of Sciences*, vol. 97, no. 21, p. 11149, 2000.

[36] Leskovec, J. and Horvitz, E.: Planetary-scale views on a large instant-messaging network. *Proceedings of the 17th International Conference on World Wide Web*, pp. 915–924, 2008.

[37] Fronczak, A., Fronczak, P. and Hołyst, J.: Average path length in random networks. *Physical Review E*, vol. 70, no. 5, p. 56110, 2004.

[38] Chung, F. and Lu, L.: The diameter of sparse random graphs. *Advances in Applied Mathematics*, vol. 26, no. 4, pp. 257–279, 2001.

[39] Albert, R., Jeong, H. and Barabási, A.: The diameter of the world wide web. *Nature*, vol. 401, p. 130, 1999.

[40] Newman, M.E.J., Strogatz, S.H. and Watts, D.J.: Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, vol. 64, no. 2, pp. 1–17, Jul 2001.

[41] Leskovec, J., Kleinberg, J. and Faloutsos, C.: Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 2, 2007.

[42] Albert, R. and Barabási, A.: Statistical mechanics of complex networks. *Reviews of Modern Physics*, vol. 74, pp. 47–97, June 2002.

[43] Mislove, A., Marcon, M., Gummadi, K., Druschel, P. and Bhattacharjee, B.: Measurement and analysis of online social networks. *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pp. 29–42, 2007.

[44] Flickr. Retrieved September 2011.
Available at: `http://www.flickr.com`

[45] Livejournal. Retrieved September 2011.
Available at: `http://www.livejournal.com`

[46] Orkut. Retrieved September 2011.
Available at: `http://www.orkut.com`

[47] Youtube. Retrieved September 2011.
Available at: `http://www.youtube.com`

[48] Newman, M.: The structure and function of complex networks. *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[49] Newman, M.: Power laws, pareto distributions and zipf's law. *Contemporary Physics*, vol. 46, no. 5, pp. 323–351, 1996.

[50] Dorogovtsev, S.N. and Mendes, J.F.F.: Scaling behaviour of developing and decaying networks. *Europhysics Letters*, vol. 52, p. 33, 2000.

[51] Lee, S.H., Kim, P.-J. and Jeong, H.: Statistical properties of sampled networks. *Physical Review E*, vol. 73, no. 1, pp. 1–7, Jan 2006.

[52] Leskovec, J. and Faloutsos, C.: Sampling from large graphs. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 636, 2006.

[53] Breiger, R.: The duality of persons and groups. *Social Forces*, vol. 53, p. 181, 1974.

[54] Gui, C. and Dutton, R.D.: Distribution of in-degree in random digraphs. *Computer Science. University of Central Florida*, Feb 2001. Retrieved July 2010. Available at: `http://csif.cs.ucdavis.edu/~guic/random-di.pdf`

[55] Barrat, A. and Weigt, M.: On the properties of small-world network models. *The European Physical Journal B*, vol. 13, no. 3, pp. 547–560, 2000.

[56] Newman, M. and Watts, D.: Renormalization group analysis of the small-world network model. *Physics Letters A*, vol. 263, no. 4-6, pp. 341–346, 1999.

[57] Newman, M. and Watts, D.: Scaling and percolation in the small-world network model. *Physical Review Series E*, vol. 60, no. 6; PART B, pp. 7332–7342, 1999.

[58] de Solla Price, D.: A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science*, vol. 27, no. 4, pp. 292–306, 1976.

[59] Durrett, R.: *Random Graph Dynamics*. Cambridge University Press, Nov 2006. ISBN 9780521866569.

[60] Dorogovtsev, S.N., Goltsev, A.V. and Mendes, J.F.F.: Pseudofractal scale-free web. *Physical Review E*, vol. 65, p. 066122, 2002.

[61] Klemm, K. and Eguíluz, V.M.: Highly clustered scale-free networks. *Physical Review E*, vol. 65, p. 036123, Feb 2002.

[62] Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., Upfal, E., Center, I. and Jose, C.S.: Stochastic models for the web graph. *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pp. 57–65, 2000.

[63] Dorogovtsev, S.N., Mendes, J.F.F. and Samukhin, A.N.: Structure of growing networks: Exact solution of the Barabási-Albert's model. *Physics Letters*, vol. 85, p. 4633, 2000.

[64] Dorogovtsev, S.N., Mendes, J.F.F. and Samukhin, A.N.: WWW and internet models from 1955 till our days and the "popularity is attractive" principle. *Unpublished Manuscript*, Sep 2000. Retrieved January 2010.
Available at: `http://arxiv.org/abs/cond-mat/0009090v1`

[65] Dorogovtsev, S.N. and Mendes, J.F.F.: Effect of the accelerating growth of communications networks on their structure. *Physical Review E*, vol. 63, p. 25101, 2001.

[66] Dorogovtsev, S. and Mendes, J.: Evolution of networks with aging of sites. *Physical Review E*, vol. 62, no. 2, pp. 1842–1845, 2000.

[67] Krapivsky, P., Redner, S. and Leyvraz, F.: Connectivity of growing random networks. *Physical Review Letters*, vol. 85, no. 21, pp. 4629–4632, 2000.

[68] Bianconi, G. and Barabási, A.: Bose-Einstein condensation in complex networks. *Physical Review Letters*, vol. 86, pp. 5632–5635, 2001.

[69] Bianconi, G. and Barabási, A.: Competition and multiscaling in evolving networks. *Europhysics Letters*, vol. 54, pp. 436–442, 2001.
Available at: `http://dx.doi.org/10.1209/epl/i2001-00260-6`

[70] Ergun, G. and Rodgers, G.J.: Growing random networks with fitness. *Physica A*, vol. 303, p. 261, 2002.

[71] Klemm, K. and Eguiluz, V.M.: Growing scale-free networks with small world behavior. *Physical Review E*, vol. 65, no. 5, pp. 57102–57103, 2001.

[72] Dangalchev, C.: Generation models for scale-free networks. *Physica A: Statistical Mechanics and its Applications*, vol. 338, no. 3-4, pp. 659–671, 2004.

[73] Davidsen, J., Ebel, H. and Bornholdt, S.: Emergence of a small world from local interactions: Modeling acquaintance networks. *Physical Review Letters*, vol. 88, no. 12, p. 128701, 2002.

[74] Ebel, H., Davidsen, J. and Bornholdt, S.: Dynamics of social networks. *Complexity*, vol. 8, pp. 24–27, 2002.

[75] Bansal, S., Khandelwal, S. and Meyers, L.: Evolving clustered random networks. *BMC BioInformatics*, vol. 10, p. 405, 2009.

[76] Serrano, M. and Boguna, M.: Tuning clustering in random networks with arbitrary degree distributions. *Physical Review E*, vol. 72, no. 3, 2005. ISSN 15393755.

[77] Holme, P. and Kim, B.J.: Growing scale-free networks with tunable clustering. *Physical Review E*, vol. 65, no. 2, pp. 1–4, Jan 2002.

[78] Jin, E., Girvan, M. and Newman, M.: Structure of growing social networks. *Physical Review E*, vol. 64, no. 4, pp. 46132–46132, 2001.

[79] Robins, G., Pattison, P., Kalish, Y. and Lusher, D.: An introduction to exponential random graph (p*) models for social networks. *Social Networks*, vol. 29, no. 2, pp. 173–191, 2007.

[80] Frank, O. and Strauss, D.: Markov graphs. *Journal of the American Statistical Association*, vol. 81, no. 395, pp. 832–842, 1986.

[81] Robins, G., Snijders, T., Wang, P., Handcock, M. and Pattison, P.: Recent developments in exponential random graph (p*) models for social networks. *Social Networks*, vol. 29, no. 2, pp. 192–215, 2007.

[82] Butts, C.: Cycle census statistics for exponential random graph models. *IMBS Technical Report*, vol. MBS 06-05, p. 50, 2006.

[83] Snijders, T., Pattison, P., Robins, G. and Handcock, M.: New specifications for exponential random graph models. *Sociological Methodology*, vol. 36, no. 1, pp. 99–153, 2006.

[84] Pattison, P. and Wasserman, S.: Logit models and logistic regressions for social networks: II. multivariate relations. *British Journal of Mathematical and Statistical Psychology*, vol. 52, pp. 169–193, 1999.

[85] Besag, J.: Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 32, no. 2, pp. 192–236, 1974.

[86] Pattison, P. and Robins, G.: Neighborhood-based models for social networks. *Sociological Methodology*, vol. 32, pp. 301–337, 2002.

[87] Hunter, D.: Curved exponential family models for social networks. *Social Networks*, vol. 29, no. 2, pp. 216–230, 2007.

[88] Snijders, T.: Markov chain Monte Carlo estimation of exponential random graph models. *Journal of Social Structure*, vol. 3, no. 2, pp. 1–40, 2002.

[89] Robins, G., Pattison, P. and Woolcock, J.: Small and other worlds: Global network structures from local processes. *American Journal of Sociology*, vol. 110, no. 4, pp. 894–936, 2005.

[90] Leskovec, J., Chakrabarti, D., Kleinberg, J. and Faloutsos, C.: Realistic, mathematically tractable graph generation and evolution, using Kronecker multiplication. *KDD 2005: Knowledge Discovery in Databases*, pp. 133–145, 2005.

[91] Song, C., Havlin, S. and Makse, H.: Self-similarity of complex networks. *Nature*, vol. 433, no. 7024, pp. 392–395, Jan 2005.

[92] Fortunato, S.: Community detection in graphs. *Physics Reports*, vol. 486, pp. 75–174, 2009.

[93] Backstrom, L., Huttenlocher, D., Kleinberg, J. and Lan, X.: Group formation in large social networks: membership, growth, and evolution. *Proceedings of the 12th ACM SIGKDD International Conference*, p. 54, 2006.

[94] Latapy, M., Magnien, C. and Vecchio, N.D.: Basic notions for the analysis of large affiliation networks / bipartite graphs. *Unpublished Manuscript*, Jan 2006. Retrieved January 2010, `cond-mat/0611631v1`.
     Available at: `http://arxiv.org/abs/cond-mat/0611631v1`

[95] Blondel, V., Guillaume, J., Lambiotte, R. and Lefebvre, E.: Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, p. 10008, 2008.

[96] Birmele, E.: A scale-free graph model based on bipartite graphs. *Discrete Applied Mathematics*, vol. 157, pp. 1–23, Jun 2009.

[97] 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Retrieved January 2010. Homepage: `http://sigkdd.org/kdd2010/`.

[98] Luke, S.: *Essentials of Metaheuristics*. Lulu, 2009. Available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/.

[99] Kirkpatrick, S., Gelatt, C. and Vecchi, M.: Optimization by simulated annealing. *Science*, vol. 220, no. 4598, p. 671, 1983.

[100] Kirkpatrick, S.: Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, vol. 34, no. 5, pp. 975–986, 1984.

[101] Lam, J. and Delosme, J.-M.: An efficient simulated annealing schedule: Implementation and evaluation. *Yale University Technical Report*, pp. 1–28, Jul 2007.

[102] Toivonen, R., Kovanen, L., Kivelä, M., Onnela, J., Saramäki, J. and Kaski, K.: A comparative study of social network models: Network evolution models and nodal attribute models. *Social Networks*, vol. 31, no. 4, pp. 240–254, 2009.

[103] Maryak, J. and Chin, D.: Global random optimization by simultaneous perturbation stochastic approximation. *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 780–783, 2008.

[104] Spall, J.: An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, vol. 19, no. 4, pp. 482–492, 1998.

[105] Chin, D.: A more efficient global optimization algorithm based on Styblinski and Tang. *Neural Networks*, vol. 7, no. 3, pp. 573–574, 1994.

[106] Clauset, A., Shalizi, C.R. and Newman, M.E.J.: Power-law distributions in empirical data. *SIAM Reviews*, June 2007.
Available at: `http://arxiv.org/abs/0706.1062`

[107] Bollobás, B. and Riordan, O.: The diameter of a scale-free random graph. *Combinatorica*, vol. 24, no. 1, pp. 5–34, 2004.