A GENERALIZED FRAMEWORK FOR REPRESENTING COMPLEX NETWORKS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Viplove Arora

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Mario Ventresca, Chair

School of Industrial Engineering

Dr. Jennifer L. Neville

Department of Computer Science

Dr. Joaquín Goñi

School of Industrial Engineering

Dr. Shreyas Sundaram

School of Electrical and Computer Engineering

Approved by:

Dr. Abhijit Deshmukh

Head of the School of Industrial Engineering

Dedicated to my parents Shashi Arora and Vijay Kumar Arora, and my grandmother, Asha Rani, who wanted me to become a doctor some day.

ACKNOWLEDGMENTS

The journey through grad school is a long and arduous one, which I was able to successfully navigate only with the help and constant support of multiple people. First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Mario Ventresca, for his excellent guidance and support during the course of my PhD. He has been a great mentor and his guidance propelled me to do research as a Master's student and eventually pursue a PhD. He has always been very helpful, accommodating and understanding in various aspects of the PhD life, and the opportunities he provided to freely explore my research interests has made me into an independent researcher.

I would also like to thank my thesis committee members, Dr. Jennifer Neville, Dr. Joaquín Goñi, and Dr. Shreyas Sundaram for their guidance and inputs that helped me improve my dissertation. I would also like to thank them for their encouraging words, insightful comments, and hard questions. I am grateful to Dr. Susan Hunter for her guidance during my time at Purdue, and for providing invaluable writing experience during IE 690.

I thank my fellow labmates Bryan, Dali, and Dawei who never got tired of hearing about my work, and always found ways to help me improve my ability to explain this research. I would also like to acknowledge the support of the 'relatively new' members of the group - Monika, Yan, Xinqi, Mintao, Michael, Abhishek, Dawoon, as well as other visiting members who were a part of the research group at various points of time.

Friends play a role that cannot be described in words, they help create a home away from home and make the time spent during a PhD an enriching experience. These people helped create an environment that facilitated sharing personal experiences, and initiating never ending stimulating discussions that allowed me to grow as a

researcher. I would like to begin by thanking all the people with whom I had the opportunity to share a home. Ashutosh, who not only played a huge role in helping me make the decision to pursue a PhD, but also learned to cook with me, motivated me to stay fit and go to Corec regularly, and made the time spent at Grissom more fun. Rohit, who was a constant presence during the PhD journey, participated in various runs, watched movies and tennis, and gave the much needed driving lessons. Mayank, who through his antics, made the last two and a half years much more enjoyable than I anticipated, shared some quick-cooking tips/hacks, and taught me the importance of being Bayesian. Akash Patil, for introducing board game culture to our group and being an awesome pseudo-roommate. Vibhav and Rohil, for the long walks and discussions and the time spent (wasted?) playing Foosball and engaging in other fun activities.

I am indebted to Ankur-Nikhita and Dheeraj-Shikha for all their time and advice, delicious food, board games, long conversations, and teaching me about life of a postdoc. I would also like to thank the badminton group, which I could always count on to de-stress after a long working day. I thank Mayank Garg for driving from UIUC to help and support me on the day of the defense, and Monika for endless discussions about research, personal life, and an uncountable number of lunches in Grissom. Then there are other friends who kept me going during the PhD: Sunny, Kalpana, Sanmathi, Anamika, Mridul, Akash Kumar, and many more. I would also like to acknowledge other friends for constantly asking me questions like, “when are you finishing the PhD?”, “till how long will you continue studying?”.

Finally, I would like to thank family members, especially my parents, for believing in me and supporting me throughout this journey.

Viplove Arora

December 2019

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABBREVIATIONS

ABM	Action-based model
ABNG	Action-based network generator
BA	Barabási-Albert
DD	Degree distribution
ED	Euclidean distance
ERGM	Exponential random graph model
GED	Graph edit distance
GP	Genetic programming
HCP	Human Connectome Project
InvSD	Inverse shortest distance
kMMD	Kernel maximum mean discrepancy
KPGM	Kronecker product graph model
KS	Kolmogorov-Smirnov
LA	Local assortativity
LSM	Latent space model
LT	Local transitivity
MCMC	Markov chain Monte Carlo
MOPBnB	Multiple objective probabilistic branch and bound
NA	No action
NSGA	Non-dominated sorting genetic algorithm
PAB	Preferential attachment on betweenness
PAC	Preferential attachment on closeness
PAD	Preferential attachment on degree
PADD	Preferential attachment on degree difference

PAID	Preferential attachment on in-degree
PAND	Preferential attachment on neighbor degree
PAOD	Preferential attachment on out-degree
PAPR	Preferential attachment on PageRank
PSA	Pareto simulated annealing
SBM	Stochastic block model
SCN	Supply chain network
SGOF	Spectral goodness of fit
SJ	Jaccard similarity
SLW	Inverse log-weighted similarity
TA	Triadic closure

ABSTRACT

Arora, Viplove Ph.D., Purdue University, December 2019. A Generalized Framework for Representing Complex Networks. Major Professor: Mario Ventresca.

Complex systems are often characterized by a large collection of components interacting in nontrivial ways. Self-organization among these individual components often leads to emergence of a macroscopic structure that is neither completely regular nor completely random. In order to understand what we observe at a macroscopic scale, conceptual, mathematical, and computational tools are required for modeling and analyzing these interactions. A principled approach to understand these complex systems (and the processes that give rise to them) is to formulate generative models and infer their parameters from given data that is typically stored in the form of networks (or graphs). The increasing availability of network data from a wide variety of sources, such as the Internet, online social networks, collaboration networks, biological networks, etc., has fueled the rapid development of network science.

A variety of generative models have been designed to synthesize networks having specific properties (such as power law degree distributions, small-worldness, etc.), but the structural richness of real-world network data calls for researchers to posit new models that are capable of keeping pace with the empirical observations about the topological properties of real networks. The mechanistic approach to modeling networks aims to identify putative mechanisms that can explain the dependence, diversity, and heterogeneity in the interactions responsible for creating the topology of an observed network. A successful mechanistic model can highlight the principles by which a network is organized and potentially uncover the mechanisms by which it grows and develops. While it is difficult to intuit appropriate mechanisms for network formation, machine learning and evolutionary algorithms can be used to

automatically infer appropriate network generation mechanisms from the observed network structure.

Building on these philosophical foundations and a series of (not new) observations based on first principles, we extrapolate an action-based framework that creates a compact probabilistic model for synthesizing real-world networks. Our action-based perspective assumes that the generative process is composed of two main components: (1) a set of actions that expresses link formation potential using different strategies capturing the collective behavior of nodes, and (2) an algorithmic environment that provides opportunities for nodes to create links. Optimization and machine learning methods are used to learn an appropriate low-dimensional action-based representation for an observed network in the form of a row stochastic matrix, which can subsequently be used for simulating the system at various scales. We also show that in addition to being practically relevant, the proposed model is relatively exchangeable up to relabeling of the node-types.

Such a model can facilitate handling many of the challenges of understanding real data, including accounting for noise and missing values, and connecting theory with data by providing interpretable results. To demonstrate the practicality of the action-based model, we decided to utilize the model within domain-specific contexts. We used the model as a centralized approach for designing resilient supply chain networks while incorporating appropriate constraints, a rare feature of most network models. Similarly, a new variant of the action-based model was used for understanding the relationship between the structural organization of human brains and the cognitive ability of subjects. Finally, our analysis of the ability of state-of-the-art network models to replicate the expected topological variations in network populations highlighted the need for rethinking the way we evaluate the goodness-of-fit of new and existing network models, thus exposing significant gaps in the literature.

1. INTRODUCTION

Many real-world systems can be understood as complex systems[1] comprising of a number of individual components interacting in a nontrivial fashion [1, 2]. *Self-organization* among these individual components often leads to *emergence* of a macroscopic structure that is neither completely regular nor completely random. While the study of complex systems has multiple historical roots, there are two core concepts that are common across almost all subareas of complex systems: emergence and self-organization [3]. The action-based approach proposed in this dissertation employs the twin concepts of self-organization and emergence as the basis for interactions between nodes, thus leading to synthesis of realistic networks.

These natural and artificial systems can be described/represented as networks composed of sets of nodes and edges that represent system elements and their interactions, respectively. Networks have become a useful tool for studying complex systems because the network representation provides a way of consistently discarding some details, while still being able to see how the whole system hangs together. Research in network science has provided transformative perspectives, models and methods in diverse application domains such as computer science, sociology, chemistry, biology, anthropology, psychology, geography, history and engineering [1,2,4,5]. In particular, the increasing availability of high throughput network data from a wide variety of sources such as the Internet, online social networks, citation and collaboration networks, biological networks (brain connectivity, protein-protein interactions), etc. have fueled a great deal of interest in the analysis and modeling of networks.

The development of data-based mathematical models to study these networks has provided fresh perspectives towards our understanding of complex systems, thus giv-

[1]see `https://complexityexplained.github.io/` for an introduction to some key ideas about complex systems.

ing rise to *network science* [2, 6–8]. Research in network science can be partitioned into two broad categories: (i) finding and measuring key statistical features, such as degree distributions, clustering and path lengths, that can be used to analyze the structure and behavior of networked systems, and (ii) formulating statistical models that can answer questions related to emergence of these properties in real-world networks. The two research directions are closely related because to develop new models capable of explaining the structural features of real-world networks, we must first be able to say what those features are and hence empirical data are essential. But, adequate theoretical models are equally essential if the significance of any particular empirical finding is to be correctly understood. Just as in traditional science, where theory and experiment continually stimulate one another, the science of networks is being built on the twin foundations of empirical observation and modeling [9].

1.1 Statistical network modeling

The theme of this research falls in the second category, where the aim is to develop a generative network model capable of providing a mechanistic description of the interaction processes that can be algorithmically implemented to synthesize networks exhibiting the diverse structural features observed in real-world networks. Generative modeling provides a quantitative approach that allows researchers to infer complicated hidden structural patterns in existing data and generate synthetic data sets whose structure is statistically similar to real data. Repeated execution of the stochastic algorithms of generative network models can produce a set of networks that replicate some statistical features observed in empirical data, but are otherwise random [1]. The importance of network models can be highlighted by their utility [10, 11]:

simulation to evaluate sensitivity of network functionality to parameterization, **abnormality detection** for finding subnetworks that are unexpected, **extrapolation** to synthesize larger networks for predicting future network topology, **sampling** to synthesize smaller representative networks to decrease computation time,

compression to the network generator parameters and obtain an algorithmic description, **control** to influence nodes to achieve a desired outcome of the topology, **anonymization** of a private network to synthesize a similar network for public availability, **null-modeling** to assess whether certain network properties are expressed, and **structural analysis** to reveal characteristics of the system being studied. Consequently, researchers have invested resources into constructing generative network models that are representative of the real-world phenomena being studied [12, 13].

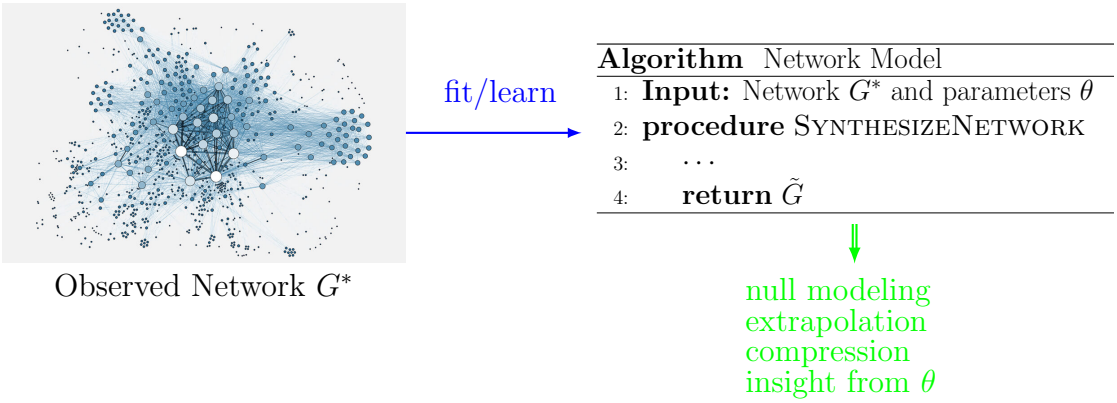


Figure 1.1. A network model consists of an algorithmic procedure that is parameterized using the observed network G^* as an input. The parameterized model can then be used to synthesize multiple networks, and subsequently contribute to our understanding of the network G^* in various ways.

These generative models attempt to identify a common set of laws and principles that can explain the structure and evolution of the networks, and the underlying system it represents [2, 14]. The set of laws and principles can therefore shed light on predicting implicit characteristics and future development of the system. For instance, a brain network model that explains the development of brain structure can help make early diagnosis of brain disease. Consequently, a goal of network modeling is to solve the problem of decoding how the observed structure of a network supports its perceived/desired function [15]. Due to this, a long-standing question in the network science community has concerned the *existence of a general model*

capable of generating synthetic networks that are statistically representative of real networks. Traditionally, generative models have been formulated to use a single empirical observation G^* of the true system as the input to an algorithmic procedure, whose parameters are best fit to synthesize networks statistically similar to G^* (see Figure 1.1 for a pictorial description). Note that it is far from guaranteed that best fit parameterization of the algorithm will yield a satisfactory generative model because the model itself might not be a good candidate for representing the observed data. Nevertheless, there are an infinite number of models that can be formulated for a given network observation [16].

Just like for any other type of structured data, two prominent paradigms exists for the modeling of network data, commonly referred to as the statistical (phenomenological) approach and the mechanistic approach [17]. The statistical approach hypothesizes relationships between the variables in the data set, where the relationship seeks only to best describe the data. This approach focuses on developing probabilistic models that specify the likelihood of observing a given network. The class of latent space models [18] and exponential random graphs [19] are prime examples of this approach. These models seek to exploit the statistical relationships and correlations within the data to make predictions about the structure. The mechanistic approach, on the other hand, relies on our scientific understanding of causal mechanisms and domain-specific microscopic mechanistic rules to grow or evolve the network over time. The small-world [13] and Barabási-Albert models [12] fall into this category. Such models are typically used for forward simulation, which can be achieved by constructing simplified mathematical formulations for the hypothesized mechanistic rules and processes governing the creation of observed data. Due to this, mechanistic models are better suited for incorporating domain knowledge, and to study effects of interventions (such as changes to specific mechanisms). Irrespective of the paradigm, a network model can be defined as follows:

Definition 1.1 (Network Model [20, 21]) *A network model is a collection*

$$\{\mathbb{P}_\theta(G), G \in \mathcal{G} : \theta \in \Theta\}, \tag{1.1}$$

where $\mathcal{G}$ is an ensemble of possible networks, $\mathbb{P}_\theta$ is a probability distribution on $\mathcal{G}$, and θ are parameters of the model ranging over possible values in Θ.

The pursuit for a unifying network model, one that can represent several previous (classes of) models found in the literature, has led to the discovery of many generative models capable of synthesizing networks with specific properties. The roots of network generators can be traced to random graphs [22–25], which assume a constant number of nodes and uniform probability on the existence of each link in the network. These generators are typically not capable of consistently reproducing phenomena observed in the real-world. Erdős and Rényi recognized this shortcoming of their model and stated "Of course, if one aims at describing a real situation, one should replace the hypothesis of equiprobability of all connection by some more realistic hypothesis. It seems plausible that by considering the random growth of more complicated structures one could obtain fairly reasonable models of more complex real growth processes" [23]. It is clear that no networks seen in nature or technology are completely random, that is, mechanisms beyond randomness shape their evolution [26].

The discovery of widely observed network topological characteristics, such as, scaling in degree distributions [12], high clustering [13, 27], degree correlations [28–30], motifs [31], and communities [32–34], have been used as a springboard to create a wide variety of network models. For example, researchers have proposed methods for synthesizing networks that exhibit a specific subset of (typically 1-3) characteristics, with parameters that could be adjusted to better reflect a given network. These approaches focus on controlling network growth or permitting non-uniform link existence and result in generators capable of reproducing behaviors such as small-worldness [13,35–39] (i.e. a node can reach other nodes in a small number of steps) and scale-free degree distributions [12, 40–42] (i.e., the probability $P(k) \approx k^{-\alpha}$, for degree k with usually $\alpha \in (2, 3)$). Some contemporary variants of random graphs are capable of exactly reproducing arbitrary degree distributions [43–48].

To achieve desirable network properties, most of the existing models either make assumptions biased by system-specific observations that are not plausible across do-

mains, or focus on replicating a few predefined topological features, such as degree distribution and clustering, at the expense of other potentially more important characteristics. Without any indication that they are either necessary or sufficient as descriptors for the actual network data, these summary quantities can often be highly misleading [5]. Further, even when a model is capable of consistently reproducing a set of target properties, it might fail to capture the naturally occurring stochasticity in those properties [49]. A fundamental challenge thus is to adequately incorporate our understanding of such workings into generative network models that are nevertheless still at least computationally, if not also analytically, tractable [20].

Motivated by these observations, a number of machine learning approaches have been proposed as well. The goal is to learn network generator parameterization from given network observation(s) by maximizing the probability of synthesizing networks with similar global characteristics. Some examples include exponential random graphs [50–52], latent space models [18], and stochastic Kronecker graphs [11, 53–55]. While successful, each approach is biased by beliefs their human designer had about the nature of observed networks and the manner that real-world networks evolve. That is, while the algorithms have some degree of freedom, they are inherently constrained by an underlying (rigid) algorithm and consequently are only capable of synthesizing networks exhibiting pre-defined topological features.

The aforementioned generators were devised by a strategy that focuses on developing algorithms capable of replicating a subset (of typically no more than three) topological network properties. Ideally, a small subset of topological properties would be sufficient to ensure the realism of synthesized networks with respect to the real-world phenomena being modeled. Unfortunately, this set of network properties is unknown and whether such a set even exists is also unknown. Additionally, designing mechanisms and models that lead to useful synthetic networks is further complicated by the stochastic local interactions and nonlinear behavior inherent in complex systems. Consequently, recent investigations have been proposed to automate the discovery of network generators for arbitrary global characteristics and phenomena [56–59]. Such

techniques hold significant promise due to their ability to circumvent much of the tedium and creative limitations faced by humans when designing a network generator.

In order to learn a particular generative model using network data, we must also choose a learning paradigm that defines the relationship between an observed network and the various parameters of the model. As the research on network modeling moves beyond simple models for network structure, the tools of statistical inference have played an important role in the development of more realistic network models. Statistical inference in the context of network models typically consists of estimation of model parameters θ from an observed network G^*. While most network models make philosophically different modeling choices based on their empirically-motivated goals, differences might even stem from their choices about how to learn from data. Sophisticated network models make hypothesis about processes that create varied structural patterns, and statistical inference can prove to be an effective tool for understanding network data and testing assumptions of these models. The increasing interest in such problems has motivated researchers to organize symposiums for the Statistical Inference of Network Models[2] to promote cross-pollination of ideas and interdisciplinary interactions.

In conclusion, despite ongoing efforts, network scientists have been unsuccessful in producing a coherent theoretical framework that can simultaneously account for discoveries like power law degree distributions, small world effect, heterogeneity and clustering in networks. Hence, new generators must be continuously developed in order to keep pace with the demand for network models exhibiting more and different local and global characteristics. Moreover, the process of scouring literature for potentially useful generators, properly configuring them and then deciding the one(s) that best represent the particular phenomena under study is a daunting task. Indeed, despite the plethora of generative models in literature, a robust framework for inferring plausible network generative models from an arbitrary network observation remains elusive [10, 11, 60–65].

[2]see `http://danlarremore.com/sinm2019/`

1.2 Problem statement

Network analysis can be seen as the language to quantify the interactions between the components of a networked system. These interactions can be hard to observe over time, which has lead to observation of these networks in the form of one-time snapshots, and consequently research on generative models for explaining the single underlying observation [66]. The goal of this research is to identify putative mechanisms that can be used to synthesize networks resembling the key topological properties of real networks and provide intuitive explanations behind the processes believed to have generated these properties. In order to maximize utility, the framework should be robust to the number and type of global network characteristics that are to be modeled, in addition to yielding easily interpretable generators. The computation time required to design the generator must also not be burdensome. To achieve these goals, we closely examine a few critical aspects related to network modeling, and the possibility of using simple mechanistic rules for link formation as a general principle determining the topology of complex networks. A framework motivated by existing observations and arguments of complex system formation is then extrapolated by utilizing a set of link formation decision process (actions) within a synthesis algorithm.

Before outlining the research questions, we would first like to define the scope of our research. Although we make some simplifying assumptions to define a restrictive scope, it does not imply that the utility of the proposed framework is restricted to the particular problems considered in this research.

- It is assumed that all of the target and synthesized networks are simple graphs, i.e., undirected with no self loops and multi-edges. The networks considered for experiments were also unweighted. We briefly consider the case of directed networks, but only in the context of supply chain networks in Chapter 5.

- The research problems we consider in this dissertation focus on parameterizing a network model based on a single network snapshot. For this reason, the pro-

posed approach can be categorized to be a 'static' (or pseudo-dynamic) model that aims to describe networks and their topology at a given time instant by sharing desirable properties with the network under consideration.

- Although most real-world networks provide additional information in the form of metadata associated with nodes or edges, the model does not utilize this information as a part of the generative process. This is not a limitation of the model as specific mechanistic rules that use this information can be added based on the application being considered. To demonstrate this, we consider the specific case of spatially embedded networks in Chapter 6 to develop models that utilize the spatial embedding in different ways.

- An important feature of most real-world networks is the existence of community structures [67]. In its current state, the model does not explicitly model communities in networks, making it an important direction for future research.

As highlighted in [68], a 'good' network model is one that is estimable from data and provides a reasonable representation of the underlying network generative process, while making theoretically plausible assumptions about the type of effects that might have produced the data. Such a model should also be amenable to examination of other competing effects that might provide better explanations of the data. With this definition and the previously stated assumptions in mind, the research questions considered in this dissertation are:

1. *Can a framework be extrapolated from existing observations and arguments of complex system formation that uses simple mechanistic rules for link formation in networks?* A mechanistic rule is a simplified mathematical formulation that emulates the process a node uses to evaluate its preference for linking with other nodes.

2. *Is it possible to combine a set of general-purpose mechanistic rules in an algorithmic framework, which under suitable parametrization can be used to model*

networks originating from a wide range of applications? If such a framework exists, assessment of the strength of association between the observed network topology and the mechanistic rules can provide parameter estimates for the proposed generative model.

3. *Can such a model strike a balance between computational tractability, empirical properties and theoretical considerations?* One common way of overcoming these trade-offs is to find an invariant for the data generating process. Exchangeability in network models can help us in addressing these issues.

4. A distinctive feature of mechanistic models is that they lend explainability to the data being modeled. Thus, one would expect that *parameters of the fitted model can provide insights about the topology and the processes that might have created the observed network.* Further, *despite the diverse application domains where these networks arise, can the same mechanistic rules be used to explain the similarities and differences between the underlying systems?*

5. In most applications, the observed network can be regarded as a sample from a set of possible networks originating from some (unknown) stochastic process. A natural question is *if a network model can learn the topological variability of the (unknown) stochastic process using a single sample network?* This can have huge implications on the development and evaluation of current and future generative network models.

1.3 Research philosophy

René Descartes and Issac Newton are often credited for laying the scientific foundations for a collection of beliefs now called the **Mechanistic Philosophy**. They believed that the workings of a system can be determined by the mechanical interaction of inanimate objects obeying universal mathematical laws of cause and effect. These beliefs led to the notion of mechanistic modeling, where hypothetical processes

are described based on our beliefs about what is happening in the system, even though we cannot directly observe these processes. Developing a mechanistic understanding of complex systems forms the cornerstone of my research philosophy. I feel data-driven mechanistic models could provide a unique perspective towards our understanding of the underlying, shared patterns observed across different complex systems as the inferences drawn from such models are based on causality of input-output relationships instead of correlations.

A mechanistic model, if successful, can provide algorithmic explanations that can help us better understand the organizing principles and processes that drive the formation of an observed system. Such models can serve several different purposes en route to establishing a mechanistic explanation, thus furthering our current understanding of the systems they represent. This makes the mechanistic modeling approach an ideal choice for understanding network formation as a single snapshot of the network provides us with limited information about the system being observed.

Most fundamental ideas in network science are based on data and meticulous observations by simultaneously looking at the World Wide Web and genetic networks, Internet and social systems. This makes us wonder if we can use these observations to take apart a complex system and try to intuit network formation principles that translate into simple yet successful generative models. While it is difficult to intuit appropriate mechanisms for network formation, machine learning and evolutionary algorithms can be used to automatically infer appropriate network generation mechanisms from the observed network structure. Eventually, the choices we make while modeling, that is, how we 'look at' and 'think about' data, will be critical to determining the usefulness of inferences drawn from the model. The essence of our research philosophy is captured by the Boxian trope: "All models are wrong, but some are useful" [69], and we firmly believe that developing a mechanistic understanding can lead to 'useful' models.

1.4 Overview of the dissertation

After defining the scope of this dissertation in Chapter 1, we provide an introduction to various network models in Chapter 2, while making important observations about link formation mechanism and processes, which subsequently leads to the extrapolation of our action-based framework. Additionally, we also discuss literature on some key topological features observed in real-world networks and briefly introduce the problem of comparing networks in Chapter 2 as they are crucial aspects that need to be examined for the development of a network model. In Chapter 3, we begin by introducing the main contribution of this dissertation, the action-based framework. We also explore some fundamental theoretical features that underline our framework. The action-based approach employs the twin concepts of self-organization and emergence as the basis for interactions between nodes, thus leading to synthesis of realistic networks.

Following the general description of the action-based framework, we show how it can be used to learn a compact probabilistic model of network formation using a mixture of link creation mechanisms in Chapter 4. Statistical comparison to existing network generators is performed and results show that the performance of our approach is comparable to the current state-of-the-art methods on a variety of network measures, while also yielding easily interpretable generators. Our experimental evaluations provide evidence that the action-based model is equally applicable to biological, technological, and social systems.

An advantage of mechanistic models is the ease with which one can incorporate domain knowledge. Since the modeler is in control of the mechanisms to include, one can encode relevant domain knowledge of known or hypothesized interaction processes between actors in the system as mechanistic rules. Along these lines, Chapter 5 considers the application of the action-based model to directed networks, particularly for the case of supply chain networks. The application is motivated by the need for a centralized approach for designing realistic supply chains that may quickly recover

from disruptions. The ability to adapt and recover from adverse circumstances is another important feature of complex systems. In Chapter 5, we test the ability of the action-based approach to synthesize robust and resilient supply chain networks, while specifically focusing on the aspect of topological resilience and capturing the heterogeneous roles of different firms in a supply chain by incorporating domain specific constraints.

In Chapters 3 and 4, we assumed that a node can use an action to interact with any node in the network. In most real-world networks, a node can only observe a subset of nodes that it can interact with. To incorporate this feature in our model, we proposed the concept of node visibility, wherein the probability of interaction between nodes depends on their attributes. In Chapter 6 we consider the specific case of spatially embedded structural brain networks to test the idea of node visibility in a continuous space. This led to a generalized version of the action-based model that utilizes spatial embedding of networks to learn better models. We found that the model sheds light on our understanding of the relationship between the structural organization of human brains and the cognitive ability of subjects.

Building on our findings in the previous chapters, Chapter 7 explores an alternative approach for evaluating generative network models and highlights that most network models fail to capture the naturally occurring variability observed in network populations. Our findings highlight the need for rethinking the way we evaluate the goodness-of-fit of new and existing network models, thus exposing significant gaps in the generative network modeling literature. We also use this evaluation technique to test if the action-based model can reproduce the topological variability of a network sampled from a known population. Chapter 8 concludes the dissertation and provides some directions for future research.

2. BACKGROUND

A number of well-known network models are essentially static in nature, i.e., the aim is to explain the existence of the observed network structure based on a single snapshot of the network. The statistical/algorithmic procedure of these generators focuses on synthesizing networks exhibiting certain local and global network statistics, with the goal of recreating a small set of the important properties observed in real-world networks. Some of these models also have a generative interpretation that allows us to think about their use in a dynamic, evolutionary setting [5].

We begin this Chapter with an introduction of well-known link formation mechanisms in Section 2.1 and also discuss relevant network properties that can be used to inspire potential link formation mechanisms. In Section 2.2, we review the literature on network models, specifically focusing on generators that consider the problem of modeling the generative process based on a single network observation. While discussing these link formation mechanisms and network models, we also make a series of (not new) observations based on first principles, leading to the extrapolation of an action-based perspective for modeling and synthesizing complex networks. Finally, we also briefly introduce the problem of evaluating network models (Section 2.3) as it is directly related to the network modeling problem.

2.1 Mechanisms for link formation

Early investigations of potential mechanisms for generating and evolving networks were limited to building models that reproduce stylized facts (for example heterogeneous node degrees, small-worldness, etc.) of real-world networks [70]. Networks are generally seen as structural representations of systems that emerge from local interactions of simpler components (nodes) [71]. Thus, modeling networks by observing

the dynamic interactions among nodes that leads to the creation of links in a network may provide a useful perspective for understanding the processes that create a network. Identifying the link creation mechanisms that lead to observed network structures is a fundamental question that is still not well understood [72]. A link creation mechanism can be some pre-defined notion that a node uses to evaluate its preference for interacting with other nodes[1]. Understanding drivers for link creation can potentially reveal the local phenomena responsible for the emergence of network properties observed at the global level.

Observation 2.1 *Stochastic local interactions give rise to emergent global structure.*

In this section, we review models and mechanisms that can be used to describe processes capable of synthesizing networks emulating some of the well-known properties observed in real-world network. The networks synthesized by these mechanisms can potentially give insights into the emergence of a variety of topological properties observed in real-world networks.

2.1.1 Erdős-Rényi random graph

The most basic graph generator is the random graph model, popularly known as the Erdős-Rényi-Gilbert model [22–25]. Erdős and Rényi defined two different random graph ensembles $G_{n,p}$ and $G_{n,m}$. In $G_{n,p}$, a graph with n nodes is generated by independently adding an edge between each pair of nodes with probability p, while $G_{n,m}$ uniformly at random picks m distinct pairs of nodes to be connected by an edge. The random graph model has been subjected to significant theoretical analysis leading to definitions of giant component, percolation, phase transitions, etc. [23, 73].

Many properties of the random graph model are exactly solvable in the asymptotic limit when $n \to \infty$. The degrees of node can be shown to follow a Poisson distribution

[1]It should be noted that such mechanisms can also be used to for emulating higher-order interactions such as subgraphs, hyperedges, etc.

with parameter $z = p(n-1)$, where n is the number of nodes and z is the mean degree. The random graph model synthesizes networks that show the small-world effect [74], but other properties of real-world networks, such as high clustering coefficients and community structure, are not observed. Also, the Poisson degree distribution is not observed in real-world networks. See [75, Chapter 3] for a more comprehensive description of properties of real-world networks that cannot be explained by the random graph model.

Observation 2.2 *Although the random graph model cannot describe real-world observations, it serves as a good baseline to test hypothesis such as: are the networks synthesized by a model better than a random graph?*

2.1.2 Small-world effect and model

High clustering and small-worldness [13,35–37,39] are properties that are observed in most real-world networks. To synthesize networks with high clustering coefficient, Watts and Strogatz developed the small-world model [13] as an alternative to the random graph. The basic idea is to create shortcuts in a regular lattice of n nodes by rewiring existing links. The rewiring procedure involves iterating through each edge in turn and, with probability p, moving one end of that edge to a new location chosen uniformly at random from the lattice, while avoiding double or self-edges. Changing the value of p from 0 to 1 allows the small-world model to interpolate between a regular lattice and a random graph (see Figure 2.1). In early 2000s, the model had deep implications for our understanding of dynamic behavior and phase transitions in real-world phenomena ranging from contagion processes to information diffusion [76]. Though the paper by Watts and Strogatz served as a pedestal for network science as a multidisciplinary field [76], the model has a basic drawback that it synthesizes networks whose degree distribution does not match most real-world networks.

Observation 2.3 *Real-world networks lie somewhere between completely random and regular lattice-like structures.*

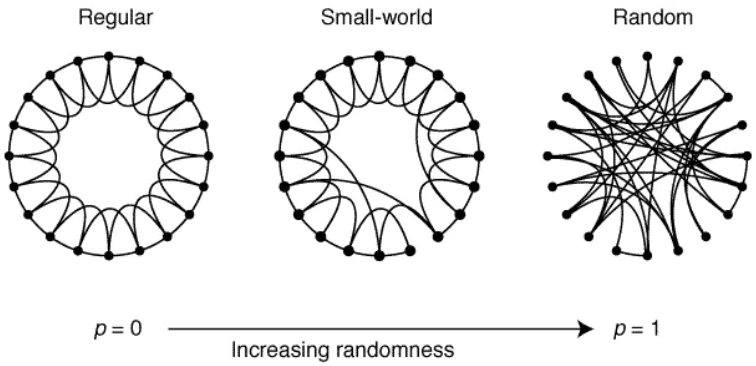


Figure 2.1. The random rewiring procedure of the Watts-Strogatz model interpolates between a regular ring lattice and a random network, without altering the number of vertices or edges in the graph. Figure from [13].

2.1.3 Heavy-tailed degree distributions

Most real-world networks are known to exhibit heavy-tailed degree distributions, that is there are few nodes that are connected to a lot of other nodes and most nodes have low degrees. Network scientists have often attributed the existence of heavy tailed degree distributions to the scale-freeness of the vertex connectivities[2] [12,40–42] (i.e., the probability $P(k) \approx k^{-\alpha}$, for degree k with usually $\alpha \in (2,3)$). Barabási and Albert [12] proposed a preferential attachment mechanism to explain the scale-freeness observed in real-world networks. The network generation algorithm begins with n_0 (connected or unconnected) nodes at time $t = 0$, and at each subsequent time step, a new node is added with $m \leq n_0$ edges. The probability that the new node is connected to an existing node is proportional to the degree of the latter. In

[2]although there have been some recent interesting discussions about the observation of scale-freeness or power law degree distributions in real-world networks, see [77–79].

other words, the new node picks m nodes from the existing network according to the multinomial distribution:

$$p_i = \frac{k_i}{\sum_j k_j}$$

where k_i denotes the degree of node i. The Barabási-Albert (BA) model results in a network with a power-law degree distribution whose exponent is empirically determined to be $\gamma = 2.9 \pm 0.1$ [5].

Observation 2.4 *Rich-gets-richer attachment is among the most powerful drivers for link formation in networks [26], hence serving as the baseline for many network models.*

The trio of network models introduced above can be seen as the basis for much of the future work on generative network modeling. There have been multiple extensions and generalizations of these models with varying goals such as synthesizing networks with scale-free degree distribution and adjustable clustering coefficients [80] or scale-free networks with clustering and communities [81]. The interested reader is directed to [1, 4, 5, 10, 20, 75, 82] for more details.

2.1.4 Homophily

Nodes in a network, especially in the context of social interactions, are typically associated with a set of sociodemographic, behavioral and interpersonal characteristics. These nodes tend to form connections with other nodes possessing similar characteristics [27]. This phenomena is explained by the principle of homophily (or homogeneity), which states that a contact between similar elements (nodes) occurs at a higher rate than among dissimilar ones [27]. Using the random graph as the null model, a network is said to be homophilic when the number of interactions between nodes with different characteristics are significantly lower than the baseline level that would be expected under a uniform random assortment reflecting groups' population share [83] (see Figure 2.2).

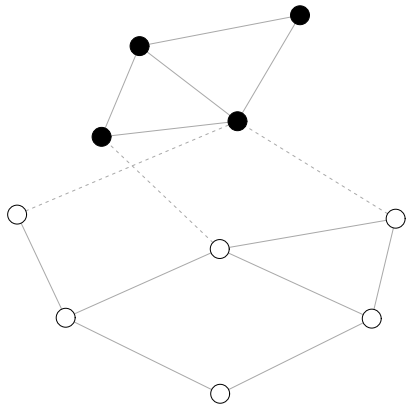


Figure 2.2. Proportion of cross (dotted) edges in the network is 0.2 (3 out of 15), while under a random assignment the expected proportion is $2pq = 0.48$, where p and q are the proportion of white and black nodes respectively. This suggests that homophily may be present in the network.

Homophily can also be observed as a structural effect, where nodes with more common neighbors are more likely to connect with each other. Thus, the source of homophily in a network can generally be attributed to the interplay between selection and social influence [84, Chapter-4], where selection accounts for the tendency of people to form friendships based on attributes determined at birth, and social influence refers to behaviors that people modify to align their interests with friends.

The correlations between characteristics (node degree, attributes, geographic location, etc.) of adjacent nodes in a network is known as assortative mixing [28]. The level of assortativity (or disassortativity) can have profound effects on the topological structure of a network, particularly when considered at the local level [85].

The prevalence of homophily in social networks has been well studied in the econometric literature on strategic network formation [83,86,87], which necessitates the development of models that can help us understand the processes that lead to homophily. Consequently, [88] proposed a network model that uses the idea of homophily to explain the data patterns observed in social networks, such as transitivity (a friend of

a friend is a friend), balance (the enemy of my friend is an enemy) and the existence of cohesive subgroups of nodes (communities).

Homophily can also be observed as a structural effect, where nodes with more common neighbors are more likely to connect with each other. Structural diversity [89] on the other hand emphasizes the importance of connectivity between common neighbors. Observations using structural diversity for link prediction show that it may lead to violation of the principle of structural homophily i.e., not only does the number of common neighbors but also the sub-graph of the common neighbors plays a role in link formation [89]. This implies the possibility of using structural diversity as a mechanism driving the formation of local structures.

Observation 2.5 *Using homophily in network structure together with node-attribute information can potentially enhance the capabilities of a network model by devising a link formation mechanism that optimizes the contribution of different mechanisms towards the formation of the observed network topology.*

2.1.5 Hierarchy

A property commonly observed in real-world networks is the existence of hierarchical organization among nodes, such that small groups of nodes organize in a hierarchical manner into increasingly larger groups [90]. Herbert Simon [91] proposed the concept of hierarchy, stating that most living and artificial complex systems are organized at multiple levels, creating a hierarchy of systems and subsystems. Hierarchy in a complex system can thus be represented as a tree of relationships, where closely related pairs of vertices have lowest common ancestors that are lower in the tree than those of more distantly related pairs [92, 93]. Hierarchical organization in real-world complex networks was first observed in the internet [94] and metabolic networks [95].

The relevance of hierarchy for synthesizing realistic networks was shown by [90], where properties like scale-freeness and high clustering emerge as a consequence of

hierarchical organization within small groups of nodes. This led to the notion that the presence of hierarchy in networks can be identified by checking if clustering coefficient C of a node with degree k is inversely proportional to its degree, i.e. $C(k) \propto k^{-\beta}$, where β is referred to as the hierarchical exponent [90]. Multiple measures or algorithms capable of characterizing the hierarchical structure of complex networks have been proposed in the literature [93, 96–99]. Hierarchy has also been seen as a central organizing principle of complex networks, capable of offering insight into many network phenomena [93]. An example highlighting the importance of hierarchy is provided by [100], where it was shown that topological abnormalities in people with schizophrenia led to a reduction in the hierarchical organization of the structural brain networks.

Due to the pervasive nature of hierarchical organization in real-world networks, network models have been proposed to exploit this feature of complex networks. Hierarchical network models are usually derived in an iterative way by replicating the initial cluster of the network according to a certain rule. Applying this rule repetitively produces a network yielding a similar structure at several different orders of magnitude.

Observation 2.6 *Self-similarity across scales in a network can be efficaciously generated using a hierarchical network generation procedure.*

2.1.6 Other link formation mechanisms

A well known observation from real-world networks states that nodes in a network might be more likely to connect with important or popular nodes, and the perception of popularity can be a consequence of different underlying processes like fitness, centrality, optimization, etc. [101]. This principle is known as preferential attachment [12], and is commonly described as the reason behind emergence of scaling in networks. Another related observation is the existence of fractal structures in networks, which has been attributed to the repulsion between hubs (nodes with high

degrees) [102]. These mechanisms can be potential candidates for link creation, and can be coupled with other mechanisms like similarity [101], or triadic closure [103] and degree correlations [104], etc. for creating more realistic network models.

It is worthwhile to note that linking mechanisms based on inverse versions of the properties described in Section 2.1 are also possible. For example food webs, technological and biological networks show disassortative mixing patterns [28]. Many unique properties of complex networks are due to heterogeneity, making its measurement and analysis important to understand the topology of complex networks [105].

Observation 2.7 *There exist multiple, potentially domain specific, motivational drivers governing link formation and most network generators are constructed to exploit only one driver.*

2.2 Network generation models

This section discusses models for synthesizing networks with a given set of nodes, and some algorithm for sampling from the distribution $\mathbb{P}_\theta(G)$ (see Definition 1.1), depending on the desired properties of a given (single) network observation. We are particularly interested in empirically grounded network models that can be parameterized to synthesize networks exhibiting different properties. It has been recently pointed out that, contrary to previous claims, the empirical laws that generative models aim to emulate are not always supported by real data [5]. Ideally we would like:

1. A network generation model that can be parameterized to synthesize networks where many properties that are also found in real networks naturally emerge.

2. The model parameter estimation should be fast and scalable, so that we can accurately generate extremely large networks.

3. The resulting set of parameters should generate realistic-looking networks that match the statistical properties of the target, real networks.

4. The learnt or fitted model should provide some statistical explanation regarding the topology of the observed network.

5. The observations of the model based on data can be projected to the unseen structure.

2.2.1 Inhomogeneous random graphs

As discussed earlier, the Erdős-Rényi model assumes nodes are homogeneous with respect to how they connect to other nodes. While that assumption cannot reproduce empirically observed properties, it motivated a general class of models of inhomogeneous random graphs that can be visualized as attempts at making the random graph model more realistic. Fitting the Erdős-Rényi model to a given network observation essentially fixes the parameter p and thus the average degree of the network. The obvious next step is to define models that sample networks from an ensemble consisting of networks with the same degree distribution as the given network observation [43–48, 106, 107].

Configuration model

The configuration model [44, 106] fixes the number of nodes with a specific degree (the degree distribution of the observed network or node degrees sampled from an arbitrary distribution can be used as input), and uniformly samples networks with the given degree sequence. The aim of the configuration model is to construct a network with n nodes, where node i has degree k_i. One way of doing this is by creating k_i outgoing stubs from each node i, followed by a uniformly random matching of two stubs to create an edge (see Figure 2.3 for a visual description). It must be noted that the procedure described here can lead to creation of networks with self-loops and multi-edges. Alternative procedures have been proposed that can sample networks without self-loops and multi-edges [108].

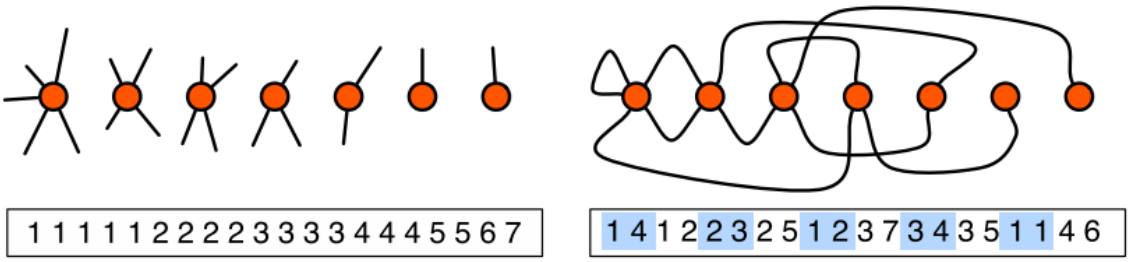


Figure 2.3. The figure shows an example of the stub matching procedure of the configuration model. The picture on the left has vertices with stubs providing a graphical description of the degree sequence. On the right, a random matching of the stubs creates the network.

Chung-Lu model

The Chung-Lu model [47, 48] is a generalization of the Configuration Model. In this model a vertex i is assigned a degree d_i from the given degree distribution and an edge is placed between the vertex pair (i, j) with probability proportional to $d_i d_j$, i.e. the probability that an edge exists between nodes i and j is given by:

$$P_{ij} = \frac{d_i d_j}{\sum_k d_k}. \tag{2.1}$$

It should be noted that $\max_i d_i^2 \leq \sum_j d_j$ to ensure that $P_{ij} \leq 1 \quad \forall i, j$. This model has the disadvantage that the final degree sequence is not precisely equal to the desired degree sequence (it matches the degree sequence in expectation), but it has some significant calculational advantages that make the derivation of rigorous results easier [4]. The Chung-Lu model is often used as the baseline for comparison owing to its simplicity and ability to synthesize fairly realistic networks [109]. Unfortunately, the Chung-Lu model synthesizes networks with low clustering coefficients making it unsuitable for most real-world applications

dk-random graphs

In [110] it was observed that fixing some structural properties in a network model to those observed in the given network can lead to the appearance of other statistical properties as a direct consequence. These observations follow from earlier research on the dk-series [111], which is a converging series of basic interdependent degree and subgraph-based properties that characterize the local network structure at an increasing level of detail, and define a corresponding series of null models or random graph ensembles [110]. Consequently, dk-graphs [110] model networks as random ensembles, where ensemble size is controlled using dk-distributions. dk-random graphs for $d = 0, 1, 2$ correspond to the random graph model [22], configuration model [44, 106] and random graphs with a given joint degree distribution [112], respectively.

In [110], dk-random graphs rely on ergodic edge-swapping operations to sample networks from the ensemble defined using the chosen dk-distributions. The lack of an edge-swapping operation that is ergodic for $3k$-distributions leads to the creation of $2.1k-$ and $2.5k-$targeting rewiring, where the moves preserve the $2k$-distribution, but each move is accepted with probability p designed to drive the graph closer to a target value of average clustering $\bar{c}(2.1k)$ or degree-dependent clustering $\bar{c}(k)(2.5k)$. Experimental results [110, 113, 114] have shown that the networks synthesized by dk-random graphs have very low dissimilarity to most real-world networks. Despite this fact, the limited inferential capabilities and inability to perform tasks such as compression, extrapolation, etc. limit the utility of dk-random graphs. We also show in Chapter 7 that dk-random graphs tend to overfit the given network observation instead of capturing the variability of the "true" generative process. We also note that there have been some promising recent developments in algorithms for sampling undirected networks with exact joint degree distribution along with other properties such as node attributes, clustering and number of connected components [115].

2.2.2 Exponential random graphs

One of the most popular statistical network models in the social science literature are the exponential random graph models (ERGM) [50–52]. ERGMs are a family of statistical models for network data that assume the existence of links in networks follows exponential distribution parameterized by some network statistics of interest [19,116]. These models are built on the notion of conditional dependence, which states that the existence of links in a network is shaped by the presence or absence of other links (and possibly node-level attributes) [19]. These dependence assumptions claim that the probability of an edge is conditionally dependent on the local structure, and influential configurations can help decode the structure of networks. ERGMs represent probability distributions over networks with an exponential linear model that uses feature counts of local graph properties ,for example, edges, triangles, paths, etc., considered relevant by the modeler:

$$\mathbb{P}(\mathbf{Y} = G^* | \boldsymbol{\theta}) = \frac{1}{Z} \exp(\boldsymbol{\theta}^T \boldsymbol{\phi}(G^*)), \tag{2.2}$$

where (i) $\boldsymbol{\phi}(G^*)$ are feature counts of G^* of the target/observed network; (ii) $\boldsymbol{\theta}$ are parameters to be learned; and (iii) Z is a normalizing constant. The generality of the exponential distribution makes it an ideal candidate for representing the conditional probability distribution between different local graph properties $\boldsymbol{\phi}(G^*)$. Further, there is theoretical evidence supporting the exponential form as it can be derived from first principles using maximum entropy arguments [117].

Generating an exponential random graph consists of the following steps [19]: (i) assume that the existence of each edge is a random variable; (ii) a dependence hypothesis is proposed that embodies the local processes assumed to generate the network; (iii) network configurations (e.g. triangles, 2-stars) get parameter values based on the dependency hypothesis; (iv) use homogeneity or other constraints to reduce number of parameters; (v) model parameters are estimated and interpreted from the observed network data to get a statistical model for the network. Though ERGMs are the

most widely used models for social networks, they are plagued with the degeneracy problem [118] (i.e., the probability distribution is biased towards empty and complete networks), whereas real-world networks are sparse. Another issue is that they may not always be consistent under sampling [119].

Observation 2.8 *The dependence assumptions and the inherent flexibility of ERGMs makes them a plausible technique for network modeling, but the absence of a consistent parameter estimation framework along with problems related to choice of features inhibits potential applications.*

2.2.3 Hierarchical network models

As discussed in Section 2.1.5, the hierarchical organization in real-world networks has inspired a number of hierarchical network models, such as Kronecker product models [11, 120], generalized graph products models [121], multiscale network generation [49], recursive matrix model [122], and Corona graphs [123, 124]. Hierarchical network models are usually derived in an iterative way by replicating the initial cluster of the network according to a certain rule, which may be deterministic or stochastic.

Here, we describe the recursive algorithm of one such model, namely Kronecker product graph models (KPGMs) [11, 53–55] and its variants [120, 125]. The main intuition behind Kronecker graphs is to create self-similar graphs by taking Kronecker product of a $b \times b$ initiator adjacency matrix, where typically $b = 2$ or 3. The Kronecker product of two matrices is given by:

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \cdots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \cdots & a_{2,m}\mathbf{B} \\ \vdots & & \ddots & \vdots \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \cdots & a_{n,m}\mathbf{B} \end{pmatrix}$$

Given a $b \times b$ initiator matrix $\mathbf{\Theta}$, where each entry of the matrix is a probability, taking $K - 1$ Kronecker products of $\mathbf{\Theta}$ with itself will result in a $b^K \times b^K$ matrix $\mathbf{P}$.

Networks can then be synthesized by treating an edge between nodes i and j as a Bernoulli random variable with $p = P_{i,j}$ (see Figure 2.4 for a pictorial representation). This process of synthesizing networks can be seen as a realization of $\mathbf{P}$, and is denoted as $R(\cdot)$. The recursion of Kronecker graphs can be written as:

$$\mathbf{P} = P_{\mathcal{K}}(\mathbf{\Theta}, K) = \begin{cases} P_{\mathcal{K}}(\mathbf{\Theta}, K-1) \otimes \mathbf{\Theta} & K > 1 \\ \mathbf{\Theta} & K = 1 \end{cases} \tag{2.3}$$

where $\mathcal{K}$ refers to KPGM. For a given real-world network G^*, the initiator matrix $\mathbf{\Theta}$ can be estimated by maximizing the likelihood of synthesizing G^* using $\mathbf{\Theta}$. Although Kronecker graphs are mathematically tractable and synthesize networks with heavy-tailed degree distributions, the resulting networks tend to have a lot of isolated nodes [126]. Also, KPGMs synthesize networks with clustering coefficients that are much smaller than what is produced in real data [127, 128].

An extension of KPGMs was proposed in [120, 125] with the goal of tackling these shortcomings and increasing variability in the synthesized networks. The proposed model ties the parameters by sampling the probability matrix before each Kronecker multiplication. Tied KPGM have the following recursive form:

$$\mathbf{P} = P_{\mathcal{T}}(\mathbf{\Theta}, K) = \begin{cases} R(P_{\mathcal{T}}(\mathbf{\Theta}, K-1)) \otimes \mathbf{\Theta} & K > 1 \\ \mathbf{\Theta} & K = 1 \end{cases} \tag{2.4}$$

where $\mathcal{T}$ refers to tied KPGM. The model can be generalized by adding an additional parameter $l \leq K$ such that the first l Kronecker multiplications are independent, and ties are introduced when $K > l$. Mixed KPGMs ($\mathcal{M}$) can be defined recursively as

$$\mathbf{P} = P_{\mathcal{M}}(\mathbf{\Theta}, K, l) = \begin{cases} R(P_{\mathcal{M}}(\mathbf{\Theta}, K-1, l)) \otimes \mathbf{\Theta} & K > l \\ P_{\mathcal{K}}(\mathbf{\Theta}, l) & K = l \end{cases} \tag{2.5}$$

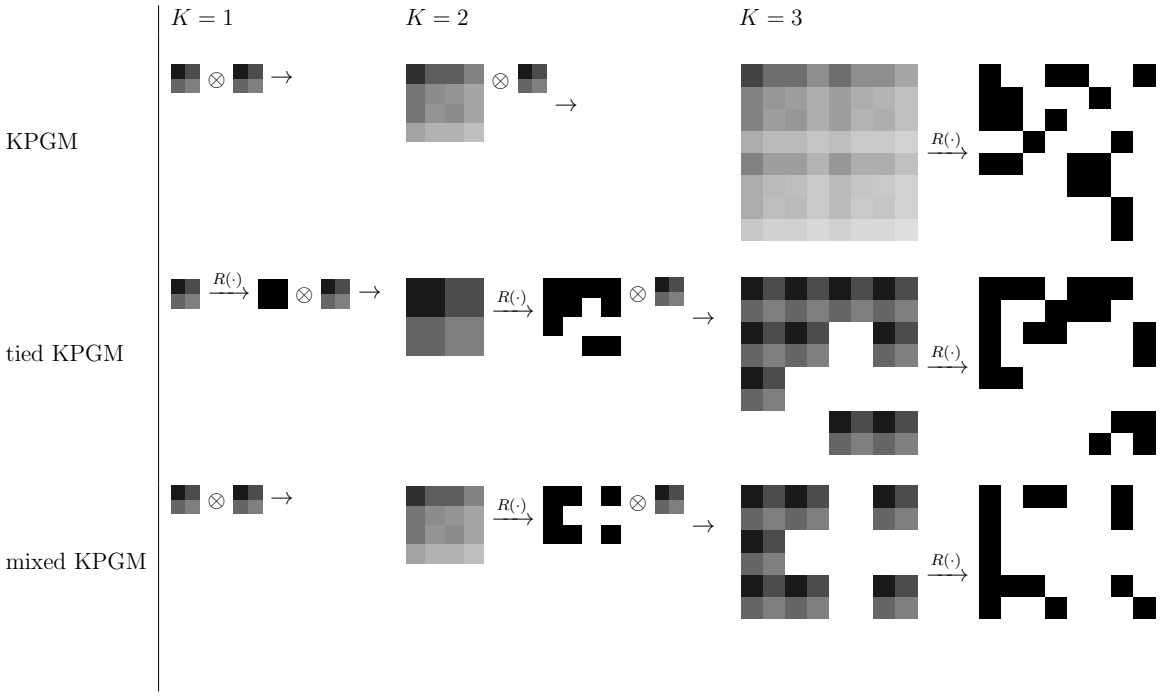


Figure 2.4. Kronecker product models for $K = 3$ and $b = 2$: For each model we show the recursive construction of $\mathbf{P}$. For tied and mixed KPGM, a realization of the probability matrix $R(\mathbf{P})$ is also shown at appropriate steps. $l = 2$ is used for mixed KPGMs.

2.2.4 Latent space models

For many real-world systems such as transportation, power grids, brains, human contact, etc., space is relevant and topology alone does not contain all the information. An important consequence when nodes are embedded in a space is that there is a cost associated with the length of edges, which in turn has dramatic effects on the topological structure of these networks [129]. This motivated the creation of latent space models [18], where each node $i \in V$ can be represented as a point $\mathbf{z}_i$ in a "low dimensional" space, say $\mathbb{R}^k$. The probability of existence of an edge is determined by the distance among the corresponding pair of nodes in the low-dimensional space, $d(\mathbf{z}_i, \mathbf{z}_j)$, and by the values of a number of covariates measured on each node individually.

Formally, let $x_{ij} = 1$ if there is an edge from node i to node j in the observed network, and 0 otherwise. The probability of there being an edge from node i to node j in a latent space model is then $\mathbb{P}(x_{ij}|\mathbf{z}_i, \mathbf{z}_j, \mathbf{y}_{ij}, \theta)$, where $\mathbf{z}_k$ is the position of node k in the latent space for $k \in V$, $\mathbf{y}_{ij}$ is some covariate information, and θ represents additional parameters. We assume conditional independence between edge probabilities, so that

$$\mathbb{P}(\mathbf{X}|\mathbf{Z}, \mathbf{Y}, \theta) = \prod_{i \neq j} \mathbb{P}(y_{ij}|\mathbf{z}_i, \mathbf{z}_j, \mathbf{x}_{ij}, \theta), \tag{2.6}$$

where $\mathbf{Y} = \{\mathbf{y}_{ij}\}$ are observed characteristics which are potentially pair-specific and vector-valued and θ and $\mathbf{Z}$ are parameters and positions to be estimated. See [130] for a review of statistical inference for the class of random dot product graphs [131], which is a more tractable version of the latent space models.

Fitting a latent space model to a given network observation involves solving the inverse problem of inferring the latent positions in a Euclidean space and the effects of observed covariates, which is typically solved using MCMC sampling [18]. [64] states that by relaxing the continuous space assumption and allowing edge probabilities to be determined by a generic function of those attributes, most generative network models can be seen as special cases of the latent space models. Spatial models have been extended in a number of directions to include treatment of transitivity, homophily on node-specific attributes, clustering, and heterogeneity of nodes, see [129, Section 4] for a review.

Other variants of latent space models embed networks in a hyperbolic space [132], where nodes are placed in a hidden hyperbolic metric and links are established according to their hyperbolic distance [133]. One of the most famous models using the hyperbolic space is the popularity versus similarity model [101]. Recent research on hyperbolic random graphs [133–135] have shown the potential utility of these methods to synthesize realistic network topologies and existence of efficient algorithms to infer the hyperbolic embedding of nodes.

Despite wide-spread usage, these models have two main drawbacks: (i) synthesized networks tend to be assortative, which might not be a reasonable assumption as many biological, technological, and economic networks show disassortativity [64], and (ii) scalability issues need to be addressed as inferring the coordinates to map a real network to its latent geometry remains a challenging inverse problem [5].

2.2.5 Stochastic block models

An important artifact of real-world networks is the existence of meso-scale structures (commonly referred to as communities or blocks) [136–139], where nodes are grouped based on their distinctive interaction patterns. Consequently, methods for finding [67,140] and modeling these structures have been a topic of significant interest in network science. One of the most popular models for modeling these structures are the Stochastic Block Models (SBM), which can be seen as a categorical/discretized version of the latent space models described in Section 2.2.4. In its simplest form, a blockmodel [141–143] is a model of network data that relies on the intuitive notion of structural equivalence: two nodes are defined to be structurally equivalent if their connectivity with similar nodes is similar [5].

The generative process of the SBM consists of two parts: (i) each node i is assigned a community label b_i, (ii) edge is inserted between nodes i and j with probability $w_{b_i b_j}$, where $\mathbf{W} = \{w_{b_i b_j}\}$ is a matrix representing the probability of an edge between nodes i and j belonging to community b_i and b_j, respectively. In this form, SBMs can be treated as a mixture of random graphs. To incorporate more features, such as degree distributions [144], multiple community memberships [145], etc. in the SBM, extensions and variants have been proposed, see [146, Section 2] for a recent review.

2.2.6 Automatic discovery of generators

Apart from the aforementioned models, alternative computation models have also been explored. For instance, cellular automata can model network evolution by ap-

plying local agent rules derived by observing collective real-world behavior [147–149]. While robust, these local rules can be very tedious and difficult to derive from an observation of network evolution, especially for nontrivial systems. Others have focused on automating the discovery of network generators, and are briefly discussed below.

Genetic programming

Recent work on network modeling has focused on automating the discovery of network generators for arbitrary global characteristics and phenomena [56,150]. Such techniques hold significant promise due to their ability to circumvent much of the tedium and creative limitations faced by humans when designing a network generator. In particular, the framework of [56] utilizes genetic programming (GP), which is a technique that uses evolutionary algorithms to evolve computer programs capable of performing a predefined task, to evolve a plausible algorithmic description of a user-defined target network. The GP based model proposed in [56] creates trees that consist of three branches at the root such that each branch is evaluated once per program execution: (i) initialize the set of nodes, (ii) grow network by adding edges among nodes initialized in step (i), and (iii) finalize network by removing and/or rewiring edges created in step (ii).

The GP based approach suffers from some drawbacks: (i) the evolutionary search is computationally expensive and scales poorly with network size, (ii) it can discover complicated generators, which can be difficult to analyze mathematically. [58] proposed to relax some restrictions on basic algorithmic structure of the GP framework that helped alleviate some of the aforementioned drawbacks, but was not tested for real-world networks.

Symbolic regression

Following the footsteps of the genetic programming based approach described above, an approach based on symbolic regression was recently developed with the

goal of automatically detecting realistic network models from empirical data [57, 59]. The basic idea is to learn a function $w(i, j)$ that assigns a weight w_{ij} to a set of edges defined using a random sample S[3], which in turn is used to obtain the probability of an edge between nodes i and j (see Figure 2.5 for a pictorial representation). The computational problem of learning the function $w(i, j)$ is solved using symbolic regression, where a network model is represented as tree-based computer programs using mathematical expressions that best fit a given network observation.

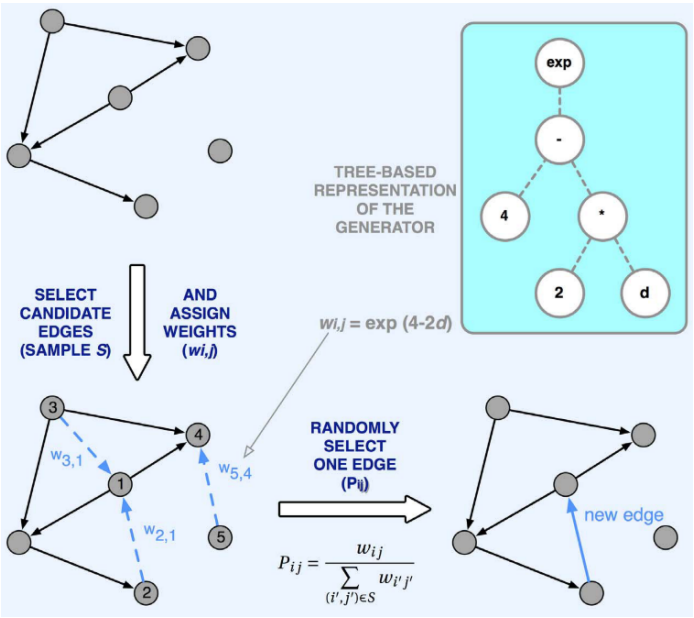


Figure 2.5. Synthesis of a network using a given tree-based representation of the generator obtained using symbolic regression. Figure from [57]. In the top right corner, we see a tree-based representation of the function $w(i, j) = 4 - \exp(4 - 2d)$, where d is the undirected distance between two nodes. Dotted edges in the bottom left graph correspond to the random sample S. This process of adding new edges is repeated until we reach a termination criteria, such as the desired number of edges.

Observation 2.9 *Most existing network models are rigidly defined, which limits their ability to synthesize realistic topologies.*

[3]A random sample is used instead of the complete set of edges to reduce computation.

In summary, most existing models either make assumptions biased by system-specific observations that are not plausible across domains, or focus on replicating a few predefined topological features, such as degree distribution and clustering, at the expense of other potentially more important characteristics. Even when a model is capable of consistently reproducing a set of target properties, it might fail to capture naturally occurring stochasticity in those properties [49]. Hence, new generators must be continuously developed manually in order to keep pace with the demand for network models exhibiting more and different local and global characteristics. Moreover, the process of scouring literature for potentially useful generators, properly configuring them and then deciding the one(s) that best represent the particular phenomena under study is a daunting task. Consequently, there is a high priority need for robust network generators [10, 11, 61–63, 151].

2.3 Evaluating network models

The problem of evaluating different generators can be approached using two principled criteria: (i) selecting the most plausible model in terms of its posterior probability; (ii) selecting the model with the highest predictive performance. While we expect both these methodologies to choose the same model, recent research [16] has shown that this is not the case. Thus, an important aspect of network modeling is to evaluate the suitability of a given model by comparing the synthesized networks with the observed network.

The roots of network comparison can be traced to the graph isomorphism problem [152], leading to the implicit usage of the notion of network dissimilarity. The standard approach to model selection is using 1-3 user defined network topological characteristics that might reflect the general structure of real-world networks. Recent observations have highlighted the need to consider multiple global characteristics when comparing networks [61–63, 110, 113, 151, 153–156]. Ideally, a subset of network properties would be sufficient to capture this dissimilarity, but unfortunately, this set

of network properties is unknown (and may not exist). The development of methods for the comparison of networks is an active area of research and in recent years many new methods have been introduced (see [157–159] for reviews and `netrd` for implementation of methods to compare two networks), which generally take the following form:

Definition 2.1 (Dissimilarity Measure) *Given two networks $G_1 \in \mathcal{G}_1$ and $G_2 \in \mathcal{G}_2$, a (bivariate) network dissimilarity measure $d(G_1, G_2)$ is a mapping $d : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathbb{R}$ from sets of graphs to a real number.*

These approaches can also be utilized within a network model for parameter estimation. Hence, robust and well-founded network dissimilarity measures are vital for the development of generative network models. It is important that a dissimilarity measure captures and adequately quantifies topological differences among networks. A good dissimilarity measure should have the ability to recognize the different roles of links and nodes, considering disconnections and other structural conditions [113]. Here, we provide a short introduction to some of these network comparison techniques.

The network morphospace [160] provides a coarse-grained approach for classifying and mapping network architectures according to a set of network-level structural characteristics. The network morphospace can be transformed to a network dissimilarity space ($\mathfrak{D}_{G^*} \subset \mathbb{R}^d$), where networks are placed based on their dissimilarity to the single observed network $G^* \in \mathcal{G}$ with respect to a variety of dissimilarity measures (see Definition 2.1), as illustrated in [161]. The utility of such a network dissimilarity space relies heavily on the choice of dissimilarity measures used for network comparison. Network science provides numerous quantitative tools to measure and classify different patterns of local and global network architectures across disparate types of systems. Any dissimilarity measure that defines a real-valued distance akin to the one in Definition 2.1, which goes to zero for a pair of isomorphic networks, can be used in the dissimilarity space. A set of node-level measures that could prove particularly useful for the network dissimilarity space is provided by the dk-series [110,111], which

is a systematic series of properties $(Y_0, Y_1, \dots)$ of network structure defined in a way such that each Y_i provides more detailed information about the network structure and Y_n fully characterizes a network with n nodes. The first three terms in the dk-series (Y = degrees + correlations + clustering) have been shown to be capable of almost fully defining local and global organization of most real-world networks that do not exhibit community structure [110].

Machine learning has also been used to find the best set of network properties for such a network dissimilarity space [156, 162]. A genetic programming based technique was also used within a meta-analysis framework [155, 163] for the same purpose. Similarly, [153] investigated six spectral graph metrics with the aim of evaluating their suitability as summary statistics for network data. NetSimile [164] provides a way of extracting a small number of descriptive, numerical features from a network. Though these methods can help us in choosing an appropriate set of properties and generally rely on computationally inexpensive local features, the findings can be biased towards the networks that were used during their empirical validation. Thus, we need more theoretically grounded ways such as the dk-series to choose the set of properties.

Efforts have also been made at using information-theoretic measures for evaluating the dissimilarity of networks [165, 166]. These measures have also been used for parameter estimation of simple network generators. Other metrics utilizing the Laplacian [167] or its spectrum [168] have also been proposed to measure the goodness-of-fit for network models. Measures using subgraph counts for comparing networks have also been proposed, for example [169] generalized the idea of degree distributions to a set of small, connected, non-isomorphic subgraphs called graphlets, which lead to the creation of graphlet degree distribution agreement [61, 170] as a measure for network comparison. Other graphlet-based measures include relative graphlet frequency distribution [169] and graphlet correlation distance [63]. Netdis [171] is another approach that creates two-step-ego-networks and compares the counts of these graphlets. It should be noted that the graphlet-based measures were developed for biological applications and have high computational costs.

A popular technique for measuring the similarity between a pair of networks is the graph edit distance (GED) [172]. The principle idea of GED is to define graph edit operations such as insertions or deletions of edges/nodes etc., along with certain edit costs associated with these operations. Based on these operations, the graph edit distance of two given graphs is the minimum cost associated with a series of edit operations. GED is a flexible error-tolerant measure that has been applied to many practical problems, but it has several problems: (i) few robust algorithms exist that can efficiently and accurately compute GED for all kinds of graphs, and (ii) the user needs to define the cost of edit operations.

The D-measure of [113] quantifies network dissimilarity by evaluating the difference between distance probability distributions of networks. The authors perform thorough empirical validation of the measure and show applications in multiple contexts while pointing out that it has computational problems when dealing with sparse networks. Geometric network comparison [173] is a statistical approach to network comparison that approximates networks as probability distributions on negatively curved manifolds. The approach is non-parametric and model-based, but fails when networks being compared are not hyperbolic.

Empirical experiments have also been performed to rank various measures [157]. They found that there is correlation between different network similarity methods and some complex network similarity methods can be closely approximated by much simpler methods. In general, most of these network measures are validated by performing classification of networks coming from different domains (or generators), which does not immediately imply that they are a sound choice for evaluating network models. To be useful in the context of evaluating the suitability of a network model, these methods need to: (i) account for the inherent stochasticity of the generators, (ii) devise a way of selecting the simplest model that also has high predictive power, and (iii) be valid for networks with or without communities. Consequently, developing model-based approaches for hypothesis testing in networks such as [174] might prove to be a fruitful endeavor. Similarly, there has been recent work on mapping networks

into natural Euclidean spaces for conventional hypothesis testing [175], which can help us determine if two groups of networks are significantly different in statistics. Bayesian methods can also be used for selecting a model $\mathcal{M}$ that is most likely to have generated a network G^* by computing the posterior probability as follows [16]:

$$\mathbb{P}(\mathcal{M}|G^*) = \frac{\mathbb{P}(G^*|\mathcal{M})\mathbb{P}(\mathcal{M})}{\mathbb{P}(G^*)}. \tag{2.7}$$

3. ACTION-BASED NETWORK MODEL

As described in Chapters 1 and 2, generative modeling of networks has been a topic attracting researchers from across disciplines such as physics, engineering, computer science, statistics, and social sciences. The pursuit for a unifying network model has lead to the discovery of a lot of generative models that adopt either the statistical (phenomenological) or mechanistic philosophy [17] for describing the observed network data. The statistical approach focuses on developing probabilistic models that specify the likelihood of observing a given network. The class of latent space models [18] and exponential random graphs [19] are prime examples of this approach. These models seek to exploit the statistical relationships and correlations within the data to make predictions about the structure. The mechanistic approach, on the other hand, relies on our scientific understanding of causal mechanisms and domain-specific microscopic mechanistic rules to grow or evolve the network over time. The small-world [13] and preferential attachment models [12] fall into this category. Such models are typically used for forward simulation, which can be achieved by constructing simplified mathematical formulations for the hypothesized mechanistic rules and processes governing the creation of observed data. Due to this, mechanistic models are better suited for incorporating domain knowledge, and to study effects of interventions (such as changes to specific mechanisms).

The importance of developing a mechanistic understanding has been a topic of central interest in our quest to comprehend the intricacies of social [176] and biological [177] systems. The driving force behind the popularity of mechanistic frameworks is that they are devised to explain the significant processes driving a phenomenon, such as spread of information, interactions between individuals, etc. While it is well-known that mechanistic models are difficult to estimate, the understanding provided

by the underlying processes enables us to answer questions about "how possibly, how plausibly, or how actually things work" [178, 21]. The ability of mechanisms to uncover underlying processes of a system can thus facilitate: (i) development of proof-of-concept models [178], (ii) extrapolating beyond the observed conditions [179], and (iii) studying the behavior of a system under various interventions [180].

Building on these philosophical foundations, we approached the problem of modeling complex systems (represented in the form of a single-layer network) through a mechanistic network model. The fundamental idea is to use the observations outlined in Chapter 2, such as small-worldness [13], preferential attachment [12], homophily [27], etc., as mechanisms/process responsible for interactions between various nodes in a network. These processes when combined with the observation that complex networks naturally form through stochastic local node interactions [71], led to the creation of the action-based framework for modeling networks [114]. Motivated by existing observations and arguments of complex system formation, a framework is then extrapolated by utilizing a set of link formation decision process (actions) within an algorithmic environment to synthesize networks. The proposed framework led to the development of a novel *action-based model* (ABM) that uses a generative algorithm extrapolated from first principles to synthesize networks that are statistically similar to the observed network(s) [114,181]. The action-based model provides a compact probabilistic description of network formation using a mixture of link creation mechanisms (actions).

Developing a computational solution using the ABM presented three significant challenges: (i) intuiting general-purpose mechanisms/actions that can depict interaction processes in real-world systems, (ii) developing a theoretically sound forward operator that provides nodes with the opportunities to create links, thus simulating emergence of a macroscopic structure from individual interactions, and (iii) solving the inverse problem of learning model parameters using given (often only one) network observation(s). We discuss the first two challenges in this Chapter, while also

reviewing some theoretical aspects that are relevant for the development of a sound network model in Sections 3.1–3.3.

3.1 Statistical units for network data

The term *statistical unit* refers to the unit of observation or measurement for which data are collected or derived. It is therefore the basic element considered when tabulating statistical data. For example, the number of nodes in a network (as a unit of measurement) or the number of connections per node (as a unit of analysis). Notice that a critical feature of statistical units is their relationship to the outcome of a statistical process because the units serve as the building blocks that a model uses to generate/synthesize data. A good statistical unit will be (i) unambiguously defined, (ii) easy to ascertain, (iii) suitable to the question being investigated, and (iv) have stable value.

In the context of network modeling, the given network is typically partially observed, although often assumed otherwise for simplicity. The network data G can be defined as a function $G\colon \mathcal{U} \mapsto \mathcal{R}$ mapping a set of statistical units $\mathcal{U}$ (e.g., nodes, edges, paths) into a response space $\mathcal{R}$ (e.g., the union of the neighborhoods of each node, a list of all connected node pairs, sequence of vertices in each path) [60]. In network modeling, the implicit units are the basic entities from which network structure is constructed, and they are determined by the theoretical context of the model. Therefore, a network modeling framework can be defined as a combination of the data generating model $\mathcal{M}_\theta$ with parameters θ, and a sampling mechanism or synthesis process Π_n used to generate a network with n units [182].

Observation 3.1 *The choice of statistical unit reflects assumptions concerning how the network model represents network data and synthesizes new instances.*

The specification of statistical units is crucial for drawing meaningful inferences from a network model as it has direct implications on the theoretical properties of the

modeling process [182, 183]. We discuss these properties, projectivity and exchangeability, in the following sections.

3.2 Projectivity

A typical setting in modeling networks involves learning models for the unobserved (population) using the observed (data) network. The data in this case consists of a sampled sub-network, which is used to estimate parameters for the population. This innately assumes that the model is consistent under sampling [119], that is, it defines a projective family. In context of a network models, the sampling mechanism or synthesis process Π_n should provide a way to project the predictions from the observed network data to a larger sample [184]. *Projectivity* is the property of a model that provides us with a context in which inferences from the model can be interpreted beyond the observed data. A statistical network model is deemed to be projective when the same parameters can be used for both the whole network and any of its sub-networks [119].

Definition 3.1 (Projective Network Model [119, 185]) *A network model $\mathcal{M}_\theta$ is projective when $A \subset B \in \mathcal{U}$ implies that $\mathbb{P}_{A,\theta}$ can be recovered by marginalization over $\mathbb{P}_{B,\theta}$, for all $\theta \in \Theta$*

$$\Pi_{B,A}(\mathbb{P}_{A,\theta}) = \mathbb{P}_{B,\theta}, \tag{3.1}$$

where $\mathbb{P}_{A,\theta}$ represents the probability distribution of the model with observed units $A \in \mathcal{U}$ and parameters $\theta \in \Theta$.

That is, the synthesis process $\Pi_{B,A}$ can be used to project the model from a smaller set of units to a larger one using the same parameter setting. Thus, the choice of the process used to synthesize or sample networks from a model can have significant ramifications on the kind of inferences that can be drawn from the model. For a generative model, projectivity in the synthesis process can ensure that the generating mechanism provides a reasonable justification for the observed network and how the network is expected to evolve if more units were observed [186].

3.3 Exchangeability

Another approach for establishing the connection between observed (data) and unobserved (population) is through the study of invariance principles for structured data [186]. Investigating symmetry through invariance principles has been a common topic in classical statistics, for example, the i.i.d. assumption in sequences of random variables provides a way of linking the sample with the population, whereas stationarity in time-series analysis can help us establish relationship between observations across time. For network models, the principle of *exchangeability* plays a central role in establishing this symmetry. Exchangeability implies that the probability of observing a particular sequence of random variables does not depend on the order of the elements in the sequence, making it an assumption on the data source rather than the data [187]. We will begin with an introduction to exchangeability for an infinite sequence of random variables, and thereafter focus on data in the form of graphs.

Theorem 3.1 (de Finetti [188]) *An infinite sequence* $(X_1, X_2, \dots)$ *of random variables is exchangeable if the joint distribution is invariant under any (finite) permutation of indices* $\sigma : \mathbb{N} \to \mathbb{N}$

$$(X_1, X_2, \dots) \stackrel{d}{=} (X_{\sigma(1)}, X_{\sigma(2)}, \dots), \tag{3.2}$$

where $\stackrel{d}{=}$ *signifies equality in distribution.*

Such representation theorems for exchangeable random structures can help us choose the appropriate class of statistical models for a given type of structured data [187]. Further, these theorems can have huge implications on statistical inference for data, because the elements X_i can be regarded as (conditionally) independent samples originating from an unknown distribution, and thus the data can be used to extract information about the value of the parameters of the population.

Before thinking about exchangeability for data in the form of networks, we need to ponder on the following question: *what is the corresponding sequence of random variables for data in the form of networks?* As described in Section 3.1, choosing

the sequence leads to an implicit choice of a statistical unit for network data [60], one that is typically overlooked by most network models. Once a unit is chosen, a network model is exchangeable if it assigns equal probability to any two networks that are equivalent up to the relabeling of the units [186]. In terms of network models, exchangeability means that the generative process of synthesizing networks does not depend on the order in which we observe data [64]. Overall, exchangeability in network models implicitly provides a way of asserting that the statistical units observed in the data are representative of the population.

Exchangeability of relational data has been the subject of numerous research studies [187, 189–196] and is not the focus of this Chapter. In the following sections, we briefly describe some of the recent notions of exchangeability that have been investigated in the statistical network analysis literature.

3.3.1 Vertex exchangeability

The famous Adlous-Hoover theorem for random arrays [189, 190] leads to the notion of vertex exchageability, a characteristic property of the network models that assume nodes as statistical units. In what follows, we assume that $\mathbf{X}$ is a symmetric binary matrix representation of an undirected network G, where $X_{ij} = 1$ implies that there is an edge between nodes i and j.

Definition 3.2 (Vertex Exchangeability) *A random array (X_{ij}) is (jointly) exchangeable if*

$$(X_{ij}) \stackrel{d}{=} (X_{\sigma(i)\sigma(j)}) \tag{3.3}$$

for simultaneous permutations σ of rows and columns of $\mathbf{X}$.

The Aldous–Hoover theorem is often referred to as the analogue of de Finetti's theorem (Theorem 3.1) for exchangeable arrays.

Theorem 3.2 (Aldous–Hoover [189, 190]) *A random array (X_{ij}) is vertex exchangeable iff there exists a measurable function $F : [0,1]^3 \to \mathbf{X}$ such that*

$$(X_{ij}) \stackrel{d}{=} (F(U_i, U_j, U_{ij})), \tag{3.4}$$

where the U's are i.i.d. uniform random variables.

For undirected random graphs, the representation in Equation 3.4 can be simplified by assuming that the function F is symmetric in its first two arguments. This leads to the notion of a graphon process, which has traditionally been used to study the limits of graph sequences in the statistics literature [192, 197].

Definition 3.3 (Graphon process [197]) *A random symmetric function $W : [0,1]^2 \to [0,1]$ that by construction satisfies*

$$F(U_i, U_j, U_{ij}) = \begin{cases} 1 & \text{if} \quad U_{ij} < W(U_i, U_j) \\ 0 & \text{otherwise,} \end{cases} \tag{3.5}$$

can be used to sample vertex exchangeable graphs.

The Aldous–Hoover theorem brings out an important relationship between graphons and vertex exchangeability. Graphons characterize the subclass of vertex exchangeable networks where any two non-overlapping subgraphs are independent. This limits the graphon model (or vertex exchangeable random graphs) to synthesize networks that are either dense or empty with probability 1. This contradicts empirical observations that most real-world networks are sparse. Sparsity means that the number of edges in a network grow sub-quadratically as a function of the number of vertices. Another implication of the graphon model is the assumption that the observed vertices are representative of the population of all vertices and thus the population is homogeneous, which again is untrue for most real-world data. Based on this, [60] points out that much of the confusion in network modeling can be attributed to the fact that most generative models innocuously assume nodes as fundamental units for network datasets.

Various attempts have been made to circumvent the limitations of the graphon model [65, 195, 198, 199]. A particularly interesting approach is the one adopted by Caron and Fox [195] (later generalized as a graphex in [65]), where networks are treated as exchangeable point processes in $[0, \infty)^2$. The model introduces heterogeneity in the network generation process by using a sociability parameter $\mathbf{w} = \{w_i\}$ for each node. This modification allows the model to synthesize networks that are sparse and follow power law degree distributions. The graphs are exchangeable in the sense that every observation of the network over a time period of length $t \geq 0$ is representative of every other observation of the network over a period of length t.

3.3.2 Relational exchangeability

Networks can also be constructed by sampling interactions among sets of individuals in the population. Under such a setting, interactions or relations between nodes (for example, edges, hyperedges, paths, etc.) act as statistical units for the network model. Such a model assumes that the sampled relations are representative of the population of relations. This leads to the notion of relational exchangeability [194], which is a generalized version of edge exchangeability [183, 200, 201]. Here we focus on the simple case of edge exchangeability, where the sequence of random variables $Y_i = (s_i, t_i)$ is the edge between nodes s_i and t_i. Network creation is modeled as arrival of edges, and edge exchangeability implies that the distribution of a random edge-labeled graph is invariant under arbitrary relabeling of its edges. For $S \subset \mathbb{N}$, let $\mathfrak{E}_S$ denote the set of all edge-labeled networks with edges labeled in S.

Definition 3.4 (Edge Exchangeability [183]) *A random edge-labeled graph* $\mathbf{X} \in \mathfrak{E}_S$ *is edge exchangeable if* $\mathbf{X}^\sigma \stackrel{d}{=} \mathbf{X}$ *for all permutations* $\sigma : S \to S$.

Three different papers have provided alternative methods for sampling edge exchangeable graphs using graph frequency models [200], interaction propensity processes [183], and a mixture of Dirichlet network distributions [201]. In each case, the models are theoretically capable of synthesizing sparse networks, but their ability

to synthesize real-world networks remains questionable and needs to be subjected to empirical validation.

3.3.3 Relative exchangeability

Network data observed from real-world applications typically contains information in the form of metadata that is more than just the structural interactions between the nodes in the population. This metadata typically exists as node attributes, and is seen as the primary source of inhomogeneity in the connection patterns of the nodes [27]. Vertex exchangeability assumes that this heterogeneity is inconsequential to the structure of the network. But, a random permutation of nodes without accounting for the node attributes would result in a network whose probability of being synthesized by a stochastic process might not be the same as that of the observed network [184, 187]. Given that the interaction between nodes is highly dependent on node attributes, it is crucial to account for this source of heterogeneity in our exchangeable models for networks.

The simplest example of a network model that utilizes this heterogeneity is the stochastic block model [141–143] (see Section 2.2.5 for details), where the nodes are partitioned into non-overlapping communities. The SBMs are a special case of the more general class of relatively exchangeable network models [182, 202]. Relative exchangeability refines vertex exchangeability by expressing the distributional symmetries of a network in terms of another (fixed) structure Z that is meant to capture the heterogeneity in the population [186]. The basic assumption behind relative exchangeability is that the structure of the data is sufficient for describing the heterogeneity in the population.

Definition 3.5 (Relative Exchangeability [182, 202]) *A random network* $\mathbf{X}$ *is relatively exchangeable with respect to a (fixed) structure* Z *if*

$$\mathbf{X}|_S \stackrel{d}{=} \mathbf{X}|_S^\sigma \quad \textit{for all permutations } \sigma : S \to S \textit{ such that } Z|_S^\sigma = Z|_S, \tag{3.6}$$

where $\mathbf{X}|_S^\sigma$ *and* $Z|_S^\sigma$ *are relabeling of* $\mathbf{X}|_S$ *and* $Z|_S$ *according to* σ *with the domain restricted to the set* $S \subset \mathbb{N}$.

The discussions above necessitates the careful choice of statistical units for network models as it is the foundational assumption about the data modeling process. Various undiscovered components of a network can be selected as units depending on the application at hand, but the revelation of a unit that can be utilized to model a wide variety of network data can have widespread implications on the subject of network analysis. Further, the choice of statistical unit affects the inferential capabilities of a network model as it serves as the starting point for the representation of a network and the corresponding theoretical properties of the model.

3.4 The action-based framework

Building on the observation that networks naturally form through stochastic processes of node interactions, we propose an action-based model (ABM) that uses a generative algorithm extrapolated from first principles to synthesize networks that are statistically similar to the observed network(s). In the action-based model, we assume that the macroscopic structure of a network emerges from structured microscopic interactions between individual nodes. That is, we assume that the structure of the network emerges from local mechanisms of node interactions, while the nodes themselves are oblivious of the global network topology, resulting in the synthesis of non-trivial network structures. The action-based approach draws inspiration from cellular automata, where simple mechanisms can combine to recreate properties of a living system, such as self-replication, adaptability, robustness and evolution [203].

The action-based approach for efficiently inferring accurate and compact models of complex systems builds on the two core concepts of complex systems: emergence and self-organization. Emergence is a phenomena observed in complex systems, where the macroscopic properties of the system cannot be completely explained by simple microscopic properties of the individuals, that is, "the whole is greater that the sum

of its parts" [3]. The action-based approach uses diverse mechanisms of interactions between components to facilitate the emergence of non-trivial network structures and behaviours at large scale. In a complex system, individual components interact and self-organize in multiple ways over time without the existence of any central authority [204]. In the action-based approach, the synthesis algorithm provides nodes with opportunities to interact and create edges as the structure evolves over time.

Real-world networks exhibit a wide variety of intricate non-trivial topological features that do not occur in completely regular or completely random networks. These topological features can be attributed to the heterogeneity, diversity and dependence in the structural connectivities of the nodes. It seems unlikely that such complex interactions can be captured by a simple or single process of interaction among nodes, but one could potentially list a finite number of decision processes or mechanistic rules that work in conjunction to create the resulting structure [84]. The question then is *how do we emulate these decision processes*, and *how these decision processes can be combined to synthesize non-trivial network structures.*

One way could be to define a probability distribution on a finite set of distinct decision processes (hereafter referred to as actions) and use them to model the interactions between nodes. For example, the exact reason for interaction between two specific nodes may not be known, but potential reasons for the interaction can be enumerated and assigned a corresponding probability. Given an appropriate probabilistic model it seems reasonable to presume that it should then be possible to synthesize a variety of topologies. The fundamental idea behind the action-based framework is to define a unifying network generative process, which follows from observations by [205] who note that there must exist an assembling algorithm to combine various actions to synthesize a variety of network structures. Identifying such an assembling algorithm can help us distinguish between network properties that are responsible for network growth from those that emerge as byproducts of the network generation process.

Our action-based perspective assumes that the generative process is composed of two main components: (i) general-purpose mechanisms/actions that can depict

interaction processes in real-world systems, and (ii) a theoretically sound forward operator that provides nodes with the opportunities to create links, thus simulating emergence of a macroscopic structure from individual interactions. We discuss these two components in Sections 3.4.1 and 3.4.2, respectively.

3.4.1 Actions

Consider a network $G = (V, E)$ containing n nodes that may or may not have any links between them. Let there exist a finite action set $A = \{a_1, \ldots, a_k\}$ containing k node actions, each representing a single decision process for node $v_i \in V$ to create a link (v_i, v_j) to any node $v_j \in V$. Without domain specific knowledge actions for choosing v_j can be based on network topological characteristics, for example a set of four actions could hypothetically be:

$$A = \left\{ \begin{array}{l} a_1 = \text{probabilistically select } v_j \text{ based on its degree,} \\ a_2 = \text{select a second neighbor } v_j \text{ uniformly at random,} \\ a_3 = \text{select } v_j \text{ as node having highest Jaccard similarity to } v_i, \\ a_4 = \text{do not make a link.} \end{array} \right\}$$

The novel concept presented herein builds upon the assumption that nodes create, rewire or delete edges by probabilistically choosing from a set of actions, thus giving rise to a global network structure. In this context, actions are similar to updating rules in cellular automata, whereby simple spatial neighborhood rules are used to evaluate the next state of a cell. Cellular automata are capable of universal computation using these rules, and very simple deterministic systems can create unpredictable complex behavior [204]. Similarly, it is conjectured that by combining simple actions and carefully choosing corresponding probabilities, we can evolve networks by simulating their stochastic local interactions.

Actions for removing, rewiring, or adding multiple edges can be easily added to the framework while following the guidelines given below:

- Adding, removing, rewiring edges should be based on some mechanistic explanation of an observed phenomena. Actions can be inspired from some real-world phenomena for forming connections or based on some common criteria based on an application to particular domains.
- A node should only have the ability to change its local structure.
- An action should attempt to provide insights into the topological patterns that exist in the target network.

Node-type as statistical unit

Let $\mathbb{P}(a = a_l)$ be the probability that action $a_l \in A$ is chosen by a node, implying that nodes are homogeneous with respect to their probability of choosing actions to create links. However, actors in real-world networks are heterogeneous with respect to their strategies for creating links, thereby rendering this assumption unlikely to be true. Hence, $\mathbb{P}(a = a_l)$ must be conditioned on the type of node, where the probability distribution over actions sufficiently differs between any two node-types, and every node is classified as being one and only one of the types. That is, let $\mathbb{P}(T = t)$ be a column vector storing the probability of a node being of type $t = 1, \dots, d \ll |V|$. An action matrix $\mathbf{M} = [\mathbb{P}(A|T)]$ can be used to define an action-based network generative process for a given set of nodes V. The action matrix $\mathbf{M}$ could be user-defined, learned from multiple network observations, or optimized/estimated from even a single network observation [114].

Definition 3.6 (Action Space) *Given action set $A = \{a_1, \dots, a_k\}$, the action space $\mathcal{A}$ is the set $\{\mathbb{P}_1(A|T), \mathbb{P}_2(A|T), \dots\}$ of all probability distributions over A.*

Then, a node-type $t \colon \{1, 2, \dots\} \mapsto \mathcal{A}$ can be represented as a map from a positive integer to the action space $\mathcal{A}$. If the action set A has k actions, then $\mathcal{A}$ is a $k - 1$ dimensional hyper-plane in $\mathbb{R}^k$ such that for $\mathbf{M} \in \mathcal{A}$

$$\sum_{i=1}^{k} M_i = 1, \quad M_i \geq 0 \quad \forall i = 1, \dots, k. \tag{3.7}$$

Definition 3.7 (node-type) *Given action set* $A = \{a_1, \ldots, a_k\}$ *and distributions* $\mathbb{P}_1(A)$ *and* $\mathbb{P}_2(A)$. *If* $d(\mathbb{P}_1(A), \mathbb{P}_2(A)) \geq \epsilon$ *then the node-type* $t(\mathbb{P}_1(A)) \neq t(\mathbb{P}_2(A))$, *for some appropriate statistical distance measure* d *(e.g., Kullback-Leibler divergence, Kolmogorov-Smirnov statistic, etc.).*

In other words, node-type corresponds to a probability distribution over a predefined finite action set A, consisting of decision process for link formation. The choice of actions can be based on the network being modeled, which is discussed in Chapters 4–6. Definition 3.7 ensures an injective map of a node-type to the action space $\mathcal{A}$, hence making it a "good" statistical unit based on the characteristics described in Section 3.1.

When compared with node as a statistical unit, node-type contains much more information about the data modeling process, while also providing an intuitive representation for networks data. This enhances the potential inferential capabilities of the action-based model. These benefits come at the cost of the difficulty of identifying the number of node-types that are sufficient to model observed data and finding the corresponding map in the action space. The inverse problem of learning model parameters using given (often only one) network observation(s) is the subject of discussion for the next Chapter.

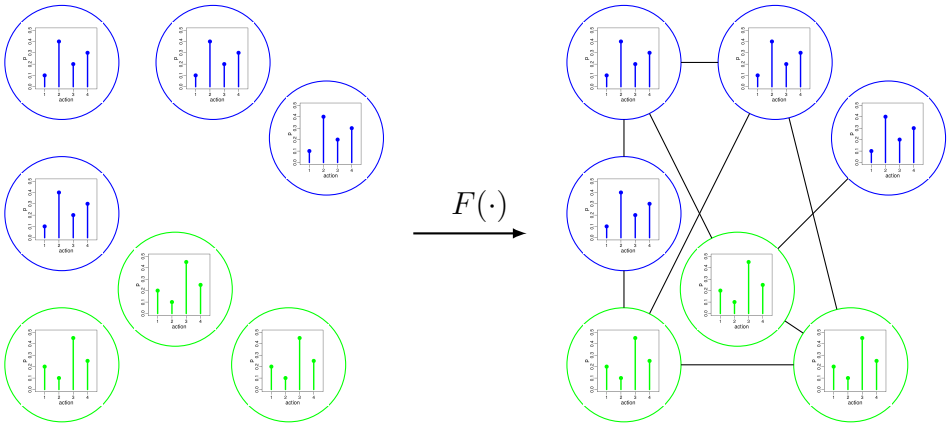
3.4.2 Synthesis algorithm

An algorithm that takes action matrix $\mathbf{M}$ as input is required to synthesize networks using the action-based model. A forward operator $F(\cdot)$ in case of the action-based model is a generative algorithm $F(\mathbf{M}, V, m, \xi)$ that can be used to synthesize networks containing n nodes and m edges using an action matrix $\mathbf{M}$, where ξ signifies the stochasticity in the generative process. As demonstrated in Figure 3.1, the generative algorithm synthesizes networks by first creating the set of desired nodes, and then probabilistically assigning a node-type t to each node according to $\mathbb{P}(T = t)$. Monte Carlo simulation is then performed using the probabilities of node actions from

$\mathbb{P}(a = a_l|t)$. For example, a model of a simple preferential attachment network will contain a single node-type that has a sufficiently high probability for the action of creating connections to nodes based on degree. Repeated sampling from $\mathbb{P}(a = a_l)$ will yield a preferential attachment network, as desired. Increasingly complicated topological properties will emerge with increased number of node-types and variety of actions.

a_1	a_2	a_3	a_4	$\bar{P}$
0.1	0.4	0.2	0.3	0.6
0.2	0.1	0.45	0.25	0.4

(a) Example action matrix with two node-types, blue and green.



(b) Pictorial description of the action-based approach using the action matrix shown above.

Figure 3.1. The action matrix shown in Figure 3.1a consists of two node-types, blue and green. On the left side of Figure 3.1b, each node in the network is assigned a node-type according to $\bar{P} = \mathbb{P}(T = t)$, following which the synthesis algorithm $F(\cdot)$ is used to provide an opportunity for nodes to create links, and thus synthesize a network.

Importantly, the synthesis algorithm allows the network modeler to easily integrate domain specific rules or constraints by implementing a problem specific set of

node actions. We consider the applications of our action-based approach to supply chain networks in Chapter 5 and structural brain networks in Chapter 6, showing how the approach can incorporate domain specific rules to provide useful insights by synthesizing networks that closely resemble the structural organization of these systems. Moreover, the modeler may wish to ensure a specific network backbone, which can be easily accommodated by defining the initial topology before executing the Monte Carlo simulation. Termination conditions for the synthesis algorithm are user defined, e.g., certain number of edges created or topological characteristics have satisfactorily emerged.

As discussed in Sections 3.2 and 3.3, the theoretical properties of a network model depend on the sampling mechanism or synthesis process used to generate the network. The synthesis algorithm for ABM produces a sequence of networks $(G_0, G_1, G_2, \ldots, G_K)$, where G_0 is the base starting network with the entire set of nodes and G_K is the network obtained upon termination of $F(\cdot)$. The choice of an appropriate forward operator $F(\cdot)$ will ensure the projectivity of the network model. To ensure this, we assume that the growth of the network G_k at iteration $0 < k < K$ during the construction depends only on the the network G_{k-1}. Further, node-types are independent and identically distributed (i.i.d.) according to $\mathbb{P}(T = t)$, and do not depend on the number of the nodes in the network. This allows easy projection of a model trained on a smaller network to a larger network (see Appendix A for related experiments).

In context of exchangeability, using node-type as the statistical unit introduces heterogeneity in the network synthesis process of ABM. While it might not be possible to write a closed-form expression for the probability $\mathbb{P}(\mathbf{X} = \mathbf{x})$ of a random network $\mathbf{X}$, we can use the forward operator $F(\cdot)$ to show that under appropriate assumptions for the synthesis algorithm, a relatively exchangeable assignment of the node-types will synthesize isomorphic networks. The forward operator $F(\mathbf{M}, V, m, \xi)$ allows us to sample networks from $\mathbb{P}(\mathbf{X} = \mathbf{x}|\mathbf{M}, V, m)$, and fixing the sequence of random variates (or sample path) $\xi = \xi_1$ will lead to synthesis of the exact same network upon repeated

sampling. Showing that $F(\mathbf{M}, V^\sigma, m, \xi_1)$ will produce that exact same network for a relatively exchangeable permutation $\sigma : V \to V$ of the node-types will imply that $\mathbb{P}(\mathbf{X} = \mathbf{x}|\mathbf{M}, V, m) = \mathbb{P}(\mathbf{X} = \mathbf{x}^\sigma|\mathbf{M}, V^\sigma, m)$, where $\mathbf{x}^\sigma$ is the network obtained after the permutation of node-types.

Let $C : V \to t$ assign each node in V to a unique node-type $t = 1, \ldots, d$. For a random permutation $\sigma : V \to V$, if two nodes i and j with $C(i) \neq C(j)$ are interchanged by σ (assuming that there are more than one node-types), the networks synthesized by $F(\mathbf{M}, V, m, \xi_1)$ and $F(\mathbf{M}, V^\sigma, m, \xi_1)$ will not be isomorphic as the nodes i and j will choose different actions (because two node-types will be sufficiently distinct, see Definition 3.7), hence creating different edges. This means that ABM with more than one node-type is not vertex exchangeable. For a random permutation σ that preserves the node-type assignment, that is, $\sigma : V \to V$ for which $C(\sigma(i)) = C(i)\ \forall i = 1, \ldots n$, the networks synthesized using $F(\mathbf{M}, V, m, \xi_1)$ and $F(\mathbf{M}, V^\sigma, m, \xi_1)$ will be isomorphic. This leads to the following theorem:

Theorem 3.3 *A random network* $\mathbf{X}$ *synthesized using the action-based model using forward operator* $F(\mathbf{M}, V, m, \xi)$*, action matrix* $\mathbf{M}$*, and node-type assignment* C *is relatively exchangeable with respect to the node-type assignment* C*.*

Proof Assume that fixing the sample path ξ_1 in the forward operator $F(\mathbf{M}, V, m, \xi_1)$ produces a network $\mathbf{x}$ using node-type assignment C. Let V^σ be some arbitrary relabeling of the nodes leading to a node-type assignment satisfying

$$C(\sigma(i)) = C(i) \quad \forall i = 1, \ldots n \text{ for some random permutation } \sigma : V \to V, \tag{3.8}$$

that is, the permutation σ preserves the node-type assignments. Using the forward operator $F(\mathbf{M}, V^\sigma, m, \xi_1)$ on the relabelled set of nodes with the same sample path ξ_1 produces network $\mathbf{x}^\sigma$. By fixing the sample path ξ_1, the probability of existence of an edge X_{ij} in a random network synthesized using $F(\cdot, \xi_1)$ depends only on the node-type assignment C. The restriction on σ specified in Equation 3.8 ensures that the node-type assignment for $\mathbf{x}$ and $\mathbf{x}^\sigma$ are identical, and so is the probability of every

edge. Thus, controlling all sources of randomness by fixing the sample paths, along with the equivalence in node-type assignments ensures that the networks $\mathbf{x}$ and $\mathbf{x}^{\sigma}$ are isomorphic, and hence $\mathbf{X}$ is relatively exchangeable with respect to the node-type assignment C. ■

It should be noted that if an action a_i is based on some node attribute ϕ of the initial network G_0, then the model will be relatively exchangeable with respect to a fixed structure $Z = (C, \phi)$ that contains both the node-type assignments and the node attributes. Thus, to ensure exchangeability for a model with actions based on topological network characteristics such as the ones listed in Section 3.4.1, G_0 needs to be an empty network. To summarize, the synthesis algorithm: (i) provides an algorithmic environment for nodes to self-organize and grow the network over time, (ii) specifies a termination criteria, and (iii) ensures exchangeability of the model.

Finally, we would like to point out that the action-based model can be seen as an interaction propensity process [186, p. 161], where the probability of interaction between two nodes i and j, given by p_{ij}, depends on two separate terms. It is widely accepted that the interactions in networks originate from some sort of dependence among the nodes, and thus there is a need to model interaction propensity using a conditional probability. This is a feature of the action-based model, where the synthesis algorithm first chooses the node i with probability p_i for creating an edge, following which i chooses to interact with j with probability $p_{j|i}$ that depends on the chosen action. Actions provide a natural way of defining $p_{j|i}$ that can help us incorporate dependence between vertices on either end of an edge[1].

[1]this has been listed as an open research problem in a recent book on Statistical Network Analysis [186, p. 169].

4. EMPIRICAL EVALUATION OF THE ACTION-BASED APPROACH

Complex networks can model a wide range of complex systems in nature and society, and many algorithms (network generators) capable of synthesizing networks with few and very specific structural characteristics (degree distribution, average path length, etc.) have been developed, see Section 2.2 for a review. However, there remains a significant lack of generators capable of synthesizing networks with strong resemblance to those observed in the real-world, which can subsequently be used as a null model, or to perform tasks such as extrapolation, compression and control. In this Chapter, we show that the action-based framework described in Chapter 3 can be used to learn a compact probabilistic model for a given target/observed network G^*, which can then be used to synthesize networks of arbitrary size. This is achieved by solving the inverse problem of learning model parameters using given (often only one) network observation(s) [206].

The goal in this Chapter is to devise a robust algorithmic framework for learning a compressed model of a given target network, and to show that the resulting generator is capable of synthesizing, with high probability, statistically similar networks to the given network. In order to maximize utility, the framework should be robust to the number and type of global network characteristics that are to be modeled, in addition to yielding easily interpretable generators. The computation time required to design the generator must also not be burdensome. We begin with an introduction of Statistical inference of generative network models, following which we provide specific details about our implementation of the action-based model. Statistical comparison to existing network generators is performed and results show that the performance of our approach is comparable to the current state-of-the-art methods on a variety of network

measures, while also yielding easily interpretable generators [114, 181]. Additionally, the action-based approach described herein allows the user to consider an arbitrarily large set of structural characteristics during the generator design process. Using experimental evaluations, we provide evidence that the proposed approach is equally applicable to biological, technological, and social systems.

4.1 Statistical inference of generative network models

Statistical inference of generative network models involves estimating parameters of the model that best fit the given data. The most widely used approach for estimating model parameters is to maximize the likelihood of the observed data. For network models such as latent space models, exponential random graphs, Kronecker graphs, etc., the goal is to tune parameters $\boldsymbol{\theta}$ such that the likelihood of generating the observed network G^* is maximized. Ultimately, the goal is to estimate network model parameters that maximize the algorithm's ability to synthesize networks that are statistically representative of G^*. As previously described in Chapter 3, it is not possible to compute the likelihoods for mechanistic models as we need to consider all the possible paths to generate any one particular network realization, which leads to a combinatorial explosion save for the most trivial settings [17]. Fortunately, mechanistic models are easy to forward simulate, and the algorithmic procedure can be used to draw samples of networks using the model. Thus we can rely on likelihood-free approaches, where we first need to define a set of network properties ($Y = \{Y_1, \cdots, Y_k\}$) to be matched, then define a quality-of-fit measure $Q(G|G^*, Y, X)$ (see Section 2.3 for a brief review) to quantify (dis)similarity and finally optimize the generator parameters $\boldsymbol{\theta}$ over the feasible domain D. Assuming that Q is a measure of network dissimilarity, the problem of estimating model parameters can be written into the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad \mathbb{E}\left[Q(G|G^*, Y, \boldsymbol{\theta})\right] \\ &\text{subject to} \quad \boldsymbol{\theta} \in D, \end{aligned} \tag{4.1}$$

Here, we assume that minimizing the expectation $\mathbb{E}\left[Q(G|G^*, Y, \boldsymbol{\theta})\right]$ (taken over the set of synthesized networks to account for stochastic variations in the network model) can be used as a proxy for maximizing the likelihood.

A network comparison technique evaluating $Q(G|G^*, Y, \boldsymbol{\theta})$ should ensure that the difference between G and G^* should not exceed what is expected from mere population variability or stochastic fluctuations [173]. The set of global network properties Y consists of measures that contain information about the topological characteristics of a network. One would expect that matching these global properties will allow the model to synthesize networks that are topologically representative of the structure of the target network.

While the formulation in Equation 4.1 can be useful for training network models, it suffers from a huge flaw because synthesizing isomorphic graphs is the optimal solution for this problem. Synthesizing isomorphic graphs is an undesirable solution because the goal here is not to exactly reproduce the target network but rather learn a model that can synthesize networks statistically similar to one another and the target network i.e. variation is expected/desired (this aspect is explored further in Chapter 7). One way to address this is to define a threshold γ for $\mathbb{E}\left[Q(G|G^*, Y, \boldsymbol{\theta})\right]$ such that when the networks G and G^* are *sufficiently similar* (i.e. $\mathbb{E}\left[Q(G|G^*, Y, \boldsymbol{\theta})\right] \leq \gamma$), the Q value should default to zero. Setting an acceptable threshold γ on Q can help us find *good* generators for a target network. Also, because network generators are stochastic algorithms, it is unlikely that the networks synthesized for a fixed parameters setting will be isomorphic to each other or the target network.

4.2 Action-based model: Implementation

Recent publications [101,207] have shed some light on the emergence of power law degree distributions in networks, showing that the interplay between popularity and

similarity plays a key role in the organization and evolution of scale-free networks. Building on these results, the action-based model defines actions corresponding to different mechanisms of evaluating popularity and (dis)similarity. Actions are used to define local pairwise interaction between nodes to make local topological changes in a network by repeatedly and probabilistically choosing from a pre-defined set of actions, thereby creating a global network structure. A synthesis algorithm $F(\mathbf{M}, \cdot)$ can then be used to synthesize networks containing n nodes using the learned action matrix $\mathbf{M}$, leading to action-based network generators (ABNG). For a given target network, $\mathbf{M}$ is determined by solving an inverse problem. A pictorial representation of this procedure can be seen in Figure 4.1.

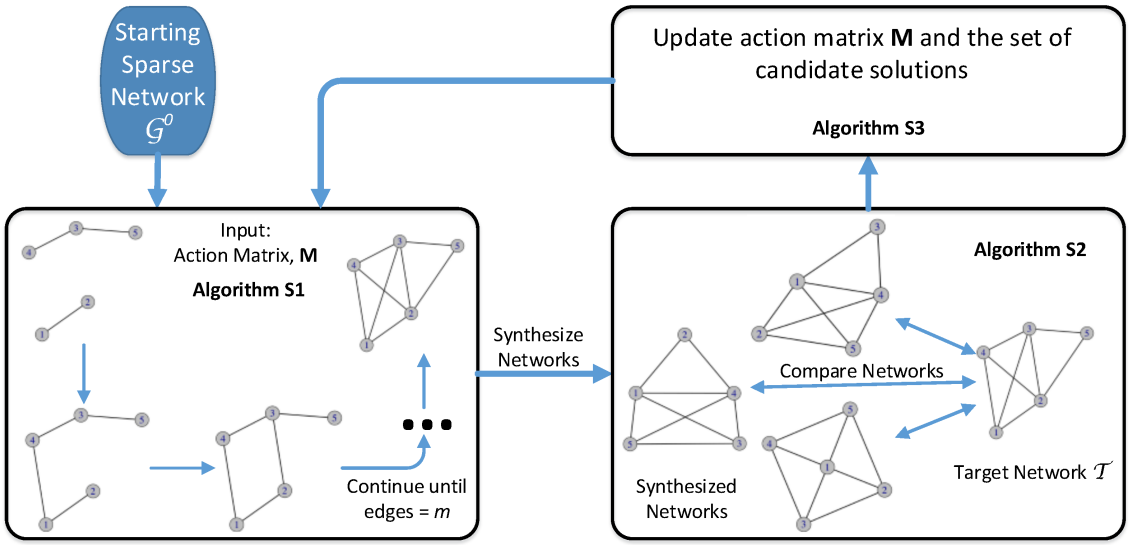


Figure 4.1. A procedure for determining action matrix $\mathbf{M}$: Algorithm S1 probabilistically adds the required number of edges using ABNG-PA(1). Algorithm S2 compares a set of synthesized networks to the target using the user-defined structural characteristics in order to determine the representativeness of action matrix $\mathbf{M}$. Algorithm S3 perturbs $\mathbf{M}$ and retains a set of best-fit solutions. The process repeats until a termination criteria (e.g., number of iterations) is satisfied.

More specifically, consider a target network $G^* = (V, E)$ with $n_t = |V|$ nodes and $m_t = |E|$ edges. Let the action set $A = \{a_1, \ldots, a_k\}$ contain k node actions. Each action provides a well-defined strategy for selecting the other end v_j of edge (v_i, v_j), for instance using preferential attachment, node similarity, node dissimilarity, etc. Let P_i be a probability distribution over the k actions that can be made by node $v_i \in V$. If nodes $v_{i'}, v_i \in V$ choose actions using the same distribution, there will exist $q \leq n$ distinct probability distributions over actions, and each $P_1, \ldots, P_q$ will correspond to a unique *node-type*. For $P^*_{q\times k} = [P_i]$, we define the Action Matrix $\mathbf{M}_{q\times(k+1)} = [P^*|\bar{P}]$, as a condensed representation containing all distinct P_i, and probability vector $\bar{P}_{q\times 1}$ containing probability of choosing actions according to P_i. For a finite set Y of user-chosen network characteristics, the problem of determining $\mathbf{M}$ can be formulated as:

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}\left[Q(G|G^*, Y, \mathbf{M})\right] \\ \text{subject to} \quad & \textstyle\sum_{j=1}^{k} M_{ij} = 1 \qquad\qquad \forall i = 1, \cdots, q \\ & \textstyle\sum_{i=1}^{q} M_{ij} = 1 \qquad\qquad j = k+1 \\ & M_{ij} \geq 0 \qquad\qquad \forall i = 1, \ldots, q \;\text{ and }\; \forall j = 1, \ldots, k+1 \end{aligned} \tag{4.2}$$

where $Q(G|G^*, Y, \mathbf{M})$ is a measure to quantify the dissimilarity between a synthesized network $G = F(\mathbf{M}, \cdot)$ and target G^* based on network characteristics Y.

4.2.1 Synthesizing networks

As highlighted in Figure 4.1, an algorithm that takes action matrix $\mathbf{M}$ as input is required to synthesize networks using the action-based approach. The principle follows from observations by [205] who note that there must exist an assembling algorithm to combine various local mechanisms that lead to the emergence of different complex network structures. A synthesis algorithm $F(\mathbf{M}, \cdot)$ uses an action matrix $\mathbf{M}$ and the local interaction mechanism of actions to synthesize networks. The action-based framework permits the use of different synthesis algorithms with

the options of adding, deleting or rewiring edges (or a combination thereof). Table 4.1 briefly describes four possible algorithms and their respective complexities for synthesizing a network. The complexity of each algorithm is given in terms of action calls, which gives the expected number of times an action will be evaluated during network synthesis. We assume that the synthesis algorithm $F(\mathbf{M}, \cdot)$ terminates when the synthesized network has the same number of edges as the target network. For the algorithms listed below, we also assume that the node set is created in the beginning (i.e., no nodes are added during the synthesis process) and only edges are altered using various actions. When unspecified, it is assumed that at each time step in the synthesis algorithm, all nodes simultaneously update their state by making changes to their edge set. Detailed description of each algorithm is given below:

Table 4.1.
Synthesis algorithms for ABNG: ρ is the probability an edge will be added by a given action matrix, and m_s is the number of edges in the starting network.

Synthesis Algorithm	Description	Action calls
ABNG-PA($\cdot$)	Addition of edges to a starting network with $m_s \ll m_t$	$\frac{1}{\rho}(m_t - m_s)$
ABNG-D($\cdot$)	Deletion of edges from a starting network with $m_t \ll m_s$	$\frac{1}{\rho}(m_s - m_t)$
ABNG-R	Rewiring of all edges in the target network	$2m_t$
ABNG-C	Degree sequence preserving addition of edges to an empty network	$\frac{1}{\rho}m_t$

- **ABNG-PA($\cdot$):** This synthesis algorithm corresponds to addition of edges to a starting network. It follows from preferential attachment algorithms, where new edges are added in each time step. The parameter in ABNG-PA($\cdot$) denotes the number of action-based queries to each node for addition of an edge. The current implementation uses ABNG-PA(1) as the synthesis algorithm i.e. each node is queried for addition of a single edge in a time step. Clearly, the computational complexity of this algorithm is proportional to the number of edges that need to be added to the starting network.

- **ABNG-D(·):** This synthesis algorithm modifies ABNG-PA(·) by deleting edges from a starting network having more edges than the target. Again, the input parameter denotes the number of edge deletion queries in each time step during network synthesis. The computational complexity of this algorithm is proportional to the number of edges that need to be deleted from the starting network.

- **ABNG-R:** Another local operation that actions can perform is rewiring i.e. changing one end of an already existing edge. The rewiring technique is used in *dk*-random graphs [110] to sample from the entire ensemble of networks. In context of ABNG, ABNG-R starts with the original target network and rewires its edges using actions. Every edge of each node is queried for an action-based rewiring resulting in a total of $2m_t$ action calls.

- **ABNG-C:** ABNG-C corresponds to a degree sequence preserving extension of ABNG-PA. It can be seen as a constrained version ABNG-PA where the degree sequence of the target network is preserved. Each node i can add edge (v_i, v_j) under the condition that the degree sequence is not violated. Joint degree distribution preserving synthesis algorithms is also a possibility. It can be seen that starting from a network with no edges, this synthesis algorithm needs to add m_t edges leading to computational complexity directly proportional to m_t.

It should be noted that the computational complexity of a synthesis algorithm can be upper-bounded by considering the complexity of the most expensive action in the action set. Also, as highlighted in Table 4.1, the computational complexity of a synthesis algorithm depends on the action matrix being used as input or ρ, which is the probability an edge will be added by a given action matrix.

4.2.2 Evaluating generator suitability

A research question directly related to network modeling is that of accessing the goodness-of-fit or evaluating the suitability of a model. As seen in Equation 4.2, a

measure Q is required to compare a network G synthesized using the generator and the target network G^*. Recent observations have highlighted the need to consider multiple global characteristics when comparing networks [61–63,110,113,151,153–156]. Ideally, a subset of network properties would be sufficient to capture this dissimilarity, but unfortunately, this set of network properties is unknown (and may not exist). To tackle this problem, ABNG optimizes the action matrix to minimize Q for a flexible set of user-defined properties Y. Thus, the problem of learning an action-based model is formulated as a multi-objective simulation optimization problem (see [208] for an introduction to the multi-objective simulation optimization).

4.3 Methods

As shown in Figure 4.1, the action-based approach is composed of three algorithms, with the tasks of (1) network synthesis, (2) comparison to target network and (3) optimization of the action matrix. The algorithms used in the current implementation are briefly described here. ABNG-PA(1) is used for network synthesis and it is outlined in detail using Algorithm 1. In this implementation, a starting sparse network is required to synthesize networks using ABNG-PA(1) because some actions can potentially become undefined due of lack of any network characteristics. We create this by randomly sampling $0.7 \times n_t$ edges from G^* (see Section 4.8.5 for experiments on varying the starting network). Alternative approaches to synthesizing networks from $\mathbf{M}$ may further improve observed results by allowing deletion and rewiring of edges and therefore capturing different types of local interactions between nodes or allowing construction from an empty starting network.

A multi-objective approach was utilized to evaluate the ability of ABNG to accurately model the desired characteristics of the target network. Consider a set $Y = \{Y_1, Y_2, \cdots, Y_N\}$ of scale-independent global network properties of interest. For each synthesized network, a 2-sample Kolmogorov-Smirnov statistic is used for each Y_i to quantify difference in distribution from the target, although alternative approaches

Algorithm 1 ABNG-PA(1)

Input: Action matrix $\mathbf{M}$, starting network $G = (V', E')$, target network $G^* = (V, E)$ and a set A of k actions
$P_i = P^*[z, :] \quad wp \quad \bar{P}_z, \quad z \leq q$ {assign a node-type using $\bar{P}$}
while $|E'| < |E|$, for every node v_i in G **do**
  $\bar{E} = \emptyset$ {initiate an empty set of edges}
  for $v_i \in V'$ **do**
    choose a_l for $v_i \quad wp \quad P^*_{il} \quad l = 1, \cdots, k$ {choose an action}
    $j = a_l(V|i)$ {choose the other end of the edge}
    $\bar{E} = \bar{E} \cup (i, j)$
  end for
  $E' = E' \cup \bar{E}$ {add edges to the network}
end while
return G

such as KL-divergence or entropy-based measures are possible. More specifically, the d statistic for each pair of synthesized and target networks can be straightforwardly calculated and the mean used as an approximation for the objective function Y_i. Further details are outlined in Algorithm 2.

Algorithm 2 Network Comparison

Input: A set of synthesized networks $\mathcal{G} = \{G_1, \cdots, G_n\}$, target network G^*, and set Y
for all $G_j \in \mathcal{G}$ **do**
  for all $Y_i(G_j) \in Y$ **do**
    $d_i = \sup_{x \in \mathbb{R}} |F_{i,G^*}(x) - F_{i,G_j}(x)|$ {$F_{i,G}$ is cumulative distribution}
  end for
end for
return $(\mathbb{E}(d_1), \cdots, \mathbb{E}(d_N))$ {$\mathbb{E}(d)$ is an estimation of the expected value.}

To quantify network structural dissimilarity, the GP system developed in [56] was used in a meta-analysis in [155] to evaluate six network centrality measures. Results indicated that of the examined centrality measures, the degree distribution, betweenness centrality, and PageRank were the most effective for quantifying the (dis)similarity between the target and the synthesized networks. We use these three measures; however, the framework allows for any user-desired measures, and they are added during our experimental evaluations when necessary.

Given a synthesis algorithm $F(\cdot)$ and set Y, the next goal is to estimate an action matrix based on its average case performance as defined by the optimization problem in Equation 4.2. To solve this multi-objective search problem, we implement Pareto Simulated Annealing (PSA) [209], as it is known to be a useful metaheuristic capable of global optimization in a large search space in a fixed amount of time. Additionally, only one evaluation of the objective function is required at each iteration when compared with population-based algorithms, which require an evaluation for each member of the population. In our experiments, we used other multi-objective optimization algorithms such as NSGA-II [210], MOPBnB [206], and multiple-gradient descent [211], but found PSA to be the most effective one.

The implementation of ABNG begins with the assumption that each node has the same probability distribution over actions. In other words, we assume that all nodes are initially homogeneous with respect to their preference over actions. This implies that all rows in P are identical and the action matrix has dimensions $1 \times (k+1)$ ($\bar{P} = 1$ in this case). Additional rows are dynamically added to $\mathbf{M}$ during the optimization as discussed in Section 4.7. PSA explores the solution space by increasing (or decreasing) randomly chosen individual elements of the action matrix, while accepting worse solutions with a probability decreasing exponentially with the number of iterations.

4.4 Results

Six human-designed network generators and 19 real-world networks (see Table B.1) were selected to evaluate the efficacy of the action-based approach. Human-devised network generators were selected for historical significance and the distinct global network properties that they model. Another benefit of using these generators is that they are designed to be simple (i.e., nodes use a single strategy for forming edges) and the simplicity should be reflected in their respective action-based models. In all experiments PageRank, degree distribution and betweenness were utilized as the global network characteristics as suggested in [163], although the approach is indifferent to this choice. Local transitivity was added as an objective for networks having more complicated structures. The 2-sample Kolmogorov-Smirnov statistic is used to quantify difference in distribution of these properties between the synthesized and target networks.

4.4.1 Modeling networks synthesized by human-devised generators

To test the ability of ABNG to replicate distinct global network properties such as scale-free degree distributions, small-world effect etc, the generator was tested using target networks with ≈ 100 nodes and ≈ 500 edges synthesized using Erdős-Rényi [22], power law [46], small world [13], Barabási-Albert [12], Forest Fire [212] and stochastic block models [143]. Having ground-truth network models allows for controlled experimental comparison across network size (number of nodes), as well as direct comparison of the action matrix to the logic of the generator that synthesized the example target network.

Solutions obtained for Erdős-Rényi, power law, small world and Barabási-Albert models all resulted in a $1 \times (k+1)$ action matrix $\mathbf{M}$ ($\bar{P} = 1$ in this case). An action matrix corresponding to the solution closest to the origin (based on 1-norm) was chosen as the model for each network, and is shown in Table 4.2. Only one node-

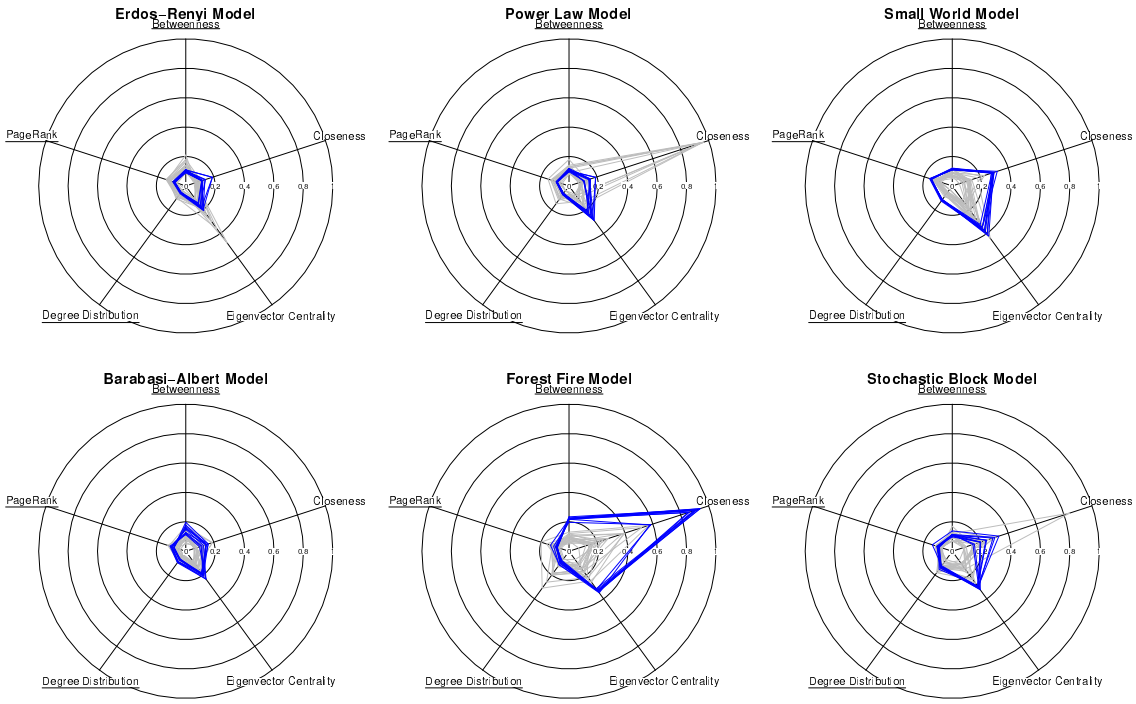


Figure 4.2. Results obtained from optimized models of human-devised generators. Network properties that were used for optimization are underlined. The plots show KS test d-statistic with the outer circle showing value of 1 (maximum possible value). The lower the value, better is the synthesized network. Each gray line corresponds to a network synthesized using the target generator. Each blue line corresponds a network synthesized using an optimized action matrix.

type was discovered, implying a homogeneous strategy when forming edges, which is consistent with these network generator algorithms.

The forest fire and block model networks, shown in Figure 4.2, required two-row action matrices, indicating the existence of two node-types (see Table 4.2). In Figure 4.2, network properties Y considered in the optimization process are underlined, others are provided for context and not expected to be near optimal. Lines closer to the origin imply a lower dissimilarity between target and synthesized networks and are thus more desirable. These results suggest that the synthesis algorithm ABNG-PA(1)

Table 4.2.
The table shows optimized action matrices for networks synthesized by human-devised generators. The following actions were used: Preferential attachment on - average neighbor degree (PAND), degree (PAD), PageRank (PAPR) and betweenness (PAB); Triadic closure (TC); Inverse log-weighted (SLW) and Jaccard similarities (SJ); and No action (NA).

Network↓ |Action→	PAND	PAD	PAPR	PAB	TC	SLW	SJ	NA	$\bar{P}$
Erdős-Rényi	0.669	0	0.149	0	0.006	0.007	0.169	0	1
Power Law	0	0.227	0	0.178	0.023	0.076	0	0.496	1
Small World	0.328	0.011	0.023	0	0	0.020	0.618	0	1
Barabási-Albert	0.132	0	0.038	0.560	0.019	0.225	0.026	0	1
Forest Fire	0	0.008	0	0.016	0	0.549	0.051	0.376	0.767
	0	0	0.041	0.547	0.029	0.019	0.216	0.148	0.233
Stochastic Block	0	0.101	0.322	0	0.207	0	0.035	0.335	0.284
	0.090	0.02	0.063	0.102	0.113	0.079	0.363	0.170	0.716

is capable of modeling the actual network generator using only one target network observation.

Figure 4.3 compares the degree distribution of networks synthesized using ABNG with Erdős-Rényi and Barabási-Albert as target networks. The distributions labeled as ABNG are the averaged statistics of 100 synthesized networks. Standard deviation bars capture the range of 90% of the synthesized networks and show that the average degree in the synthesized networks is representative of the target network.

A particularly interesting outcome was obtained when considering the target network synthesized using the stochastic block model generator. In this instance, the target network contained two communities (30 and 70 nodes, respectively) and the resulting 2-row action matrix had $\bar{P}_1 \approx 0.7$ and $\bar{P}_2 \approx 0.3$ (the corresponding action matrix is shown in Table 4.2). Thus, ABNG was able to accurately infer the existence of two communities and their approximate size in the network. The ability to detect communities in a network is an interesting observation requiring further analysis.

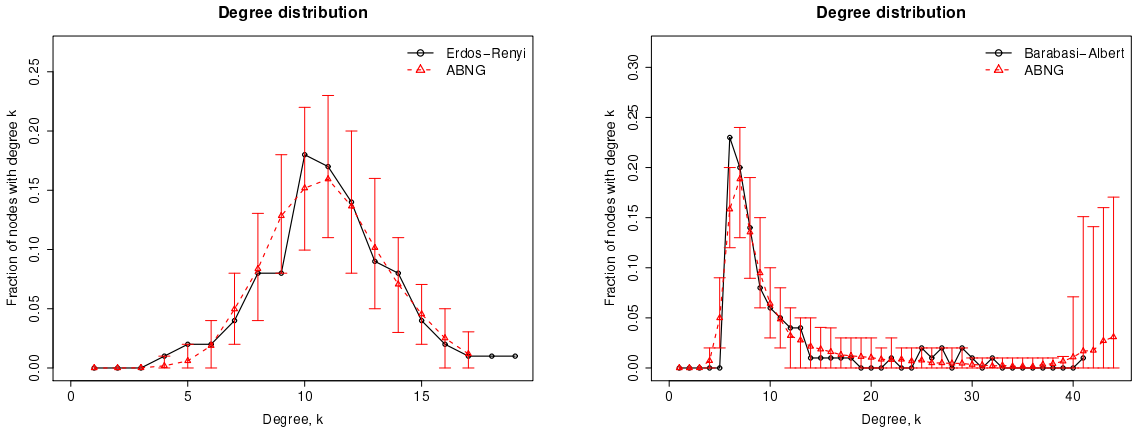


Figure 4.3. Comparing degree distributions of networks synthesized using ABNG with the target network. The deviation bars capture the range of 90% of the synthesized networks, while the line for ABNG is the mean of 100 synthesized networks.

Predicting network growth

In order to determine whether a true network generator can be discovered, an experiment is conducted where a known generator synthesizes a network and multiple network snapshots during its growth are recorded. One of the snapshots is selected as a target network for ABNG, and the goal is to ascertain whether all snapshots can be accurately synthesized using the action-based model for the target. Certain assumptions were made for the target network growth: (i) the networks grow linearly with time such that nodes are added one at a time, and (ii) the network is constructed using a consistent strategy where the action matrix is static. In order to accomplish this an action matrix is used to synthesize a network G_t at time t as learned from the target model, and then the synthesis algorithm iterates to predict the structure of graph G_{t+t_1} and G_{t-t_2} at time $t+t_1$ and $t-t_2$, respectively. If effective, this provides evidence that ABNG-PA(1) can be utilized to predict past or future structures of growing real-world networks only using the action matrix obtained from the present network.

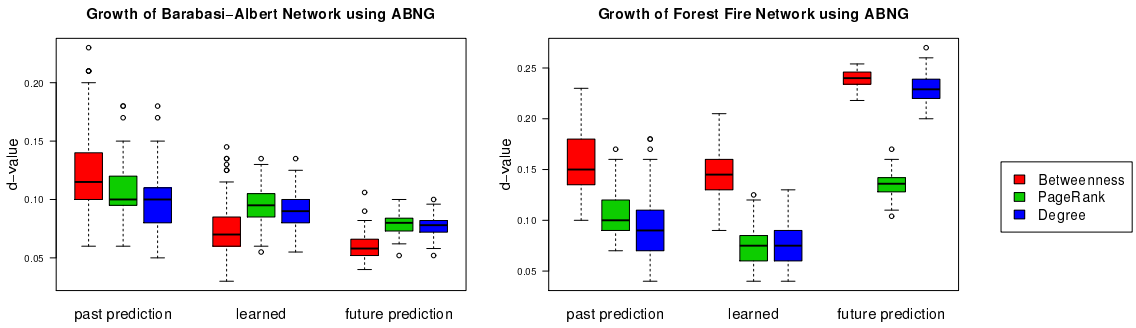


Figure 4.4. Comparison of results between the actual and the inferred networks. KS test d-values used for optimization are shown on the y-axis. Line in the middle of each box is for the median value, points show outliers and whiskers represent minimum and maximum values. Similar median d-values (and associated variation) for various network properties of different networks shows that predicted networks are statistically representative of the target networks.

Target networks were synthesized using the Barabási-Albert and forest fire models because they grow networks by adding nodes in a manner that satisfies the aforementioned assumptions. In both cases, a target network was grown to $t = 500$ (i.e., $n = 500$) where snapshots of the network at $t = 100$ and $t = 200$ were recorded. The target network corresponded to the snapshot at $t = 200$ and the resulting optimized action matrix was used to synthesize networks at $t = 100$ and $t = 500$, respectively. Figure 4.4 provides a comparison between synthesized networks in the three temporal circumstances. The variation and mean of the predicted networks for different network statistics is statistically similar to the learned network. Each plot was generated from 100 synthesized networks.

4.4.2 Modeling real world networks

Complex systems observed in the real-world typically do not have corresponding complex networks that are well modeled by existing standard network generators. Figure 4.5 presents a summary of the results, featuring heat maps for five of these

networks (other results can be found in Section 4.8.2), as well as a visual comparison of the target network with the network synthesized using ABNG. Four popular network models, namely Chung-Lu [47, 48], ERGM [52], synthetic network generator (Synt) [57] and *dk*-random graphs [110] whose parameters were best fit to the respective target networks to ensure fair comparison are also included in the heat maps. The comparison is conducted based on 5 metrics of dissimilarity, namely: 2-sample Kolmogorov-Smirnov distance based on betweenness, PageRank and local transitivity; D-measure introduced in [113]; and the spectral measure of [168] (this measure was rescaled to lie between 0 and 1 as explained in Section 4.8.3). The generators synthesize 100 networks, and the mean values are recorded in the heat maps. Further experimental results are outlined in Section 4.8.2. Note that ERGM is not used in the comparison for the US power grid, protein and social networks because it produced errors while synthesizing networks for these cases. Also, the spectral measure is not used for the US power grid and protein networks because of high computation time.

Heat maps of Figure 4.5 show that *dk*-random graphs are the best generator on all measures for each network except power grid, where ABNG is better on two properties. ABNG provides a competitive alternative to *dk*-random graphs by consistently synthesizing networks having lower or equal dissimilarity to the target, when compared with other generators and some *dk*-random graph variants.

An action matrix corresponding to the solution closest to the origin (based on 1-norm) obtained for each of these five real-world networks is shown in Table 4.3. This can help the user in making some conclusions about the structure of these networks. A common observation is that "no action" tends to have high probability for real-world networks. A possible conclusion here is that only a few nodes add edges in a time step and lead to a power law degree distribution in the network (it can be seen in Table 4.2 that among the human-devised generators, power law network had a high probability for "no action"). For the five networks considered here, we can draw the following conclusions:

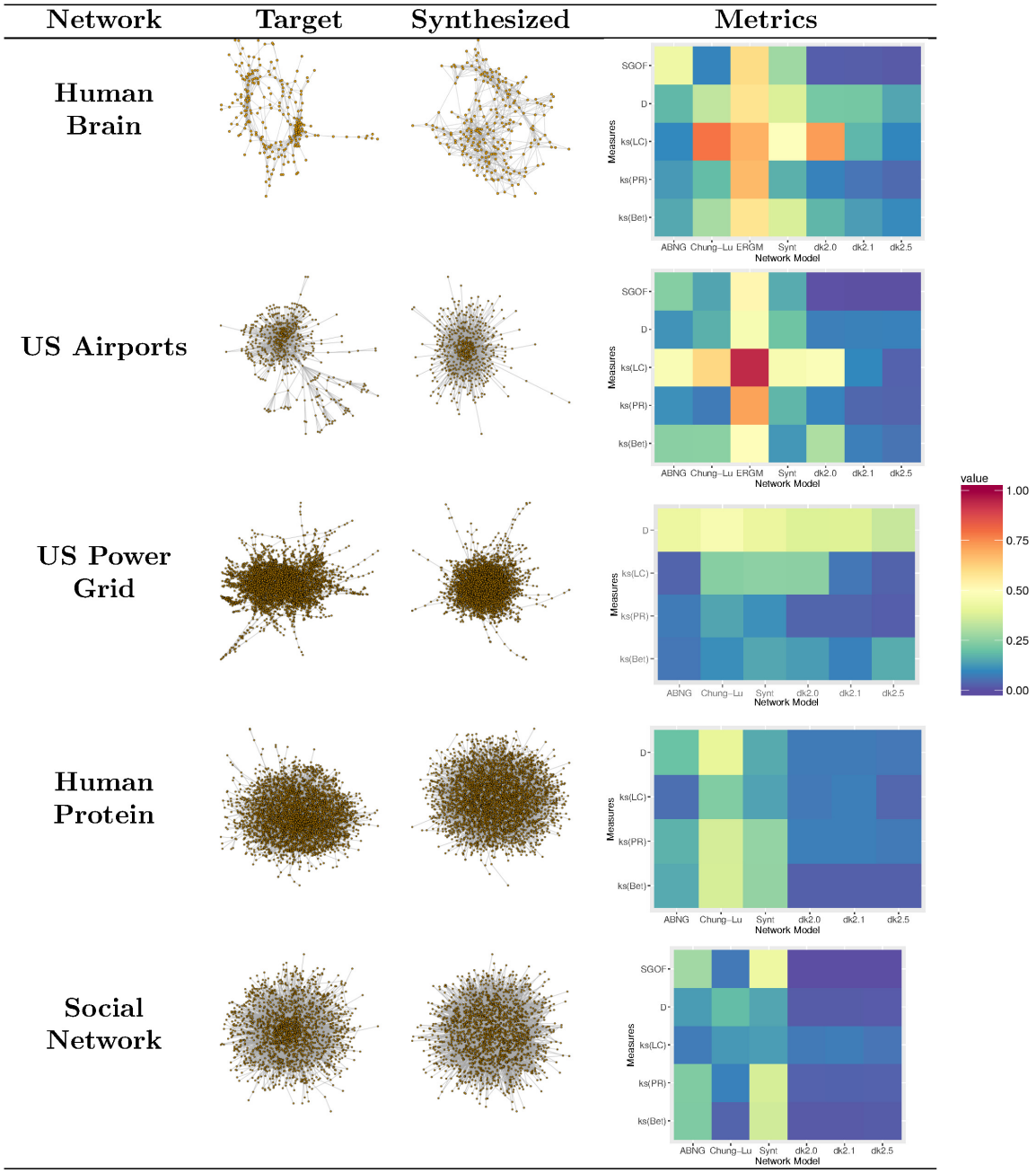


Figure 4.5. Overview of results for five real-world networks: A visualization of the target network together with the network synthesized using ABNG. Each generator synthesizes 100 networks, and the mean dissimilarity values are recorded in the heat maps. The lower the value, better is the synthesized network.

- **Human Brain:** Triadic closure and selective triadic closure are the most dominant actions. This is expected as brain networks tend to have very high clustering coefficients [100]. Although triadic closure is the dominant action, some nodes (corresponding to first row of action matrix shown in Table 4.3) show preferential attachment mechanisms also. This implies that regions of the brain interact with regions having some structural similarity or those that are highly connected.

- **US Airports:** The action-based model obtained has two types of nodes using completely different strategies. The nodes in the first category prefer not to form any connections, while those in the second category connect to nodes based on betweenness i.e. they are more likely to connect to nodes that lie in shortest paths. In the context of airport connectivity, there is a high probability of connecting to "hub" airports.

- **US Power Grid:** The action-based model obtained for the power grid network highlights connection preference to "important" nodes specifically based on PageRank, i.e. with nodes of higher quality and quantity of links.

- **Human Protein:** The action-based model for protein interaction shows existence of two types of nodes, one using "no action" with high probability and the other that use preferential attachment mechanisms based on degree and betweenness. Loosely speaking, proteins either prefer not to interact, or otherwise interact with popular nodes with higher probability.

- **Social Network:** Again, the action-based model shows existence of two types of nodes, one using "no action" with high probability and the other that use preferential attachment mechanism based on betweenness. Interestingly, the two node-types exist in equal proportions in this network. This leads us to the conclusion that people either tend to interact with popular individuals or not interact at all in this social network.

Table 4.3.
The table shows optimized action matrices for five different real-world networks. The following actions were used: Preferential attachment on - average neighbor degree (PAND), degree (PAD), PageRank (PAPR) and betweenness (PAB); Triadic closure (TC); Inverse log-weighted (SLW) and Jaccard similarities (SJ); and No action (NA).

Network↓ |Action→	PAND	PAD	PAPR	PAB	TC	SLW	SJ	NA	$\bar{P}$
Human Brain	0	0.016	0.337	0.043	0.195	0.213	0.127	0.069	0.215
	0	0	0	0	0	0.044	0.378	0.578	0.785
US Airports	0	0	0	0.061	0	0.046	0	0.893	0.804
	0	0.013	0.008	0.956	0.001	0	0	0.022	0.196
US Power Grid	0	0	0.597	0.182	0	0.005	0.146	0.070	1
Human Protein	0	0	0	0.282	0	0.009	0.002	0.707	0.723
	0	0.255	0.004	0.724	0.008	0.009	0	0	0.277
Social Network	0	0	0	0.331	0	0	0	0.669	0.516
	0	0.142	0.036	0.803	0	0.015	0	0.004	0.484

4.5 Discussion

Action-based network generators provide a flexible framework for reproducing complex structure of networks exhibiting different global/structural statistics by formulating network generation as an optimization problem. The approach consists of three distinct parts: (i) network synthesis using algorithm $F(\cdot)$, (ii) computation of dissimilarity using a user-defined set of measures Y, and (iii) an optimization technique to learn parameters $\mathbf{M}$ for a given target network. Experiments have provided evidence that ABNG can capture different network structural properties observed in a wide variety of networks, but further experimentation is required in the context of networks having various types of community structure. Improved approaches at one or more parts of the action-based framework may provide effective solutions to these challenges.

The current synthesis algorithm uses the coarsest network characteristic, the average degree, leading to a search over a large set of networks. This can be modified by adding additional constraints of synthesizing networks having the same degree sequence [107] or joint degree distribution [112]. *dk*-random graphs [110, 113] have shown that adding these local constraints enhances the capability of a generator to capture the target network. This can also be observed from the low dissimilarity values of the Chung-Lu model for the Social Network, where the synthesized networks were able to reproduce different global characteristics by just matching the expected degree. This coincides with observations of [110] that global network properties of the synthesized networks are consequences of copying only a few specific characteristics of the target. Further, it is plausible that many network characteristics are generally correlated and network structure itself may bring out or obfuscate specific correlated characteristics, even if the correlated characteristics aren't specified as part of the original algorithm goal. The ability to capture certain characteristics might also be completely coincidental as a result of the target network structure.

It should be noted that local transitivity corresponds to the 3k-constraint for *dk*-random graphs, and adding it as an objective provides a way to circumvent the issue of non-existence of sampling from 3k-random graphs. Consequently, adding local transitivity to the set of network properties Y leads to discovery of better action-based models for real-world networks. Similarly, adding modularity as an objective could lead to preservation of cluster organization in networks with community structure. [113] have shown the existence of tree-like structures in the Power Grid network and used specialized generators to model this network. Figure 4.5 shows that ABNG and *dk*-random random graphs don't adjust well for such networks (especially on the D-measure), but adding specialized actions can provide a potential solution to this problem, although it comes at the extra cost of optimizing the action matrix. Finally, using more sophisticated optimization or likelihood estimation techniques for discovering the action matrix can improve the ability of the ABNG framework to reproduce networks similar to the target.

A strength of the action-based approach lies in its ability to provide a compact representation for networks having many nodes, and can yield insights into the structure of these networks. Preliminary experiments in Section 4.8.4 discuss the scaling of ABNG-PA(1) synthesis algorithm with the number of nodes. Practically, ABNG is an appealing option when the user desires to perform tasks such as compression and extrapolation. It should be noted that the scalability of the optimization of $\mathbf{M}$ depends on the computational requirements of the optimization algorithm, action set, and objectives. ABNG can be particularly useful if the user has some domain specific knowledge about potential actions or network properties for the target network under consideration.

An important factor influencing the capability of ABNG is the choice of actions. In the implementation discussed in this chapter, actions belonging to four different categories were used (preferential attachment, triadic closure, similarity and no action), but utility of actions based on disassortativity or using node features in annotated networks needs to be explored in future research. In general, any local non-random strategy is a potential candidate for an action. In specific contexts, domain specific information can be used (e.g., to ensure constraints on potential node pairings). Existence of a compact set of actions capturing various mechanisms of local interactions among nodes is central to the action-based approach and can potentially provide crucial answers to the relationship between structure, function and dynamics of real-world networks.

4.6 A worked example

In the synthesis algorithm used in this Chapter, nodes synchronously add edges in discrete time steps. For example, as depicted in Figure 4.6, at $t = 0$ we start with a sparse starting network and actions for each node are evaluated based on the network given at $t = 0$, followed by creation of new networks at $t = 1$, which is them used for evaluation of edges to be added at $t = 2$ and so on. Also, addition of a new edge by a

node is independent of other nodes, and only depends on the outcome of a finite set of actions. To better understand the working of ABNG, let us consider a simple example. Consider a starting network G_0 (shown in Figure 4.6a) at $t = 0$ with adjacency matrix $\mathbf{A}_0$, an action matrix $\mathbf{M} = [0.7 \quad 0.3]$, and ABNG with two hypothetical actions a_1 and a_2. We compute matrices $\mathbf{A}_0^1$ and $\mathbf{A}_0^2$, where the $\mathbf{A}_0^1[i, j]$ corresponds to the probability that node i connects to node j using action a_1. The networks synthesized by ABNG (at $t = 1$) can be sampled using $\mathbf{A}_1 = 0.7\mathbf{A}_0^1 + 0.3\mathbf{A}_0^2 + \mathbf{A}_0$, where each element of $\mathbf{A}_1$ corresponds to the probability of existence of an edge.

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

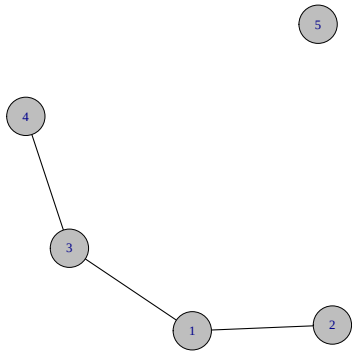
$$\mathbf{A}_0^1 = \begin{bmatrix} 0 & 0.167 & 0.333 & 0.167 & 0 \\ 0.333 & 0 & 0.333 & 0.167 & 0 \\ 0.333 & 0.167 & 0 & 0.167 & 0 \\ 0.333 & 0.167 & 0.333 & 0 & 0 \\ 0.333 & 0.167 & 0.333 & 0.167 & 0 \end{bmatrix} \quad \mathbf{A}_0^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Three networks synthesized using $\mathbf{A}_1$ are shown in Figure 4.6b. The networks synthesized at $t = 1$ can now be used as starting networks for $t = 2$. We gain several key insights from this example:

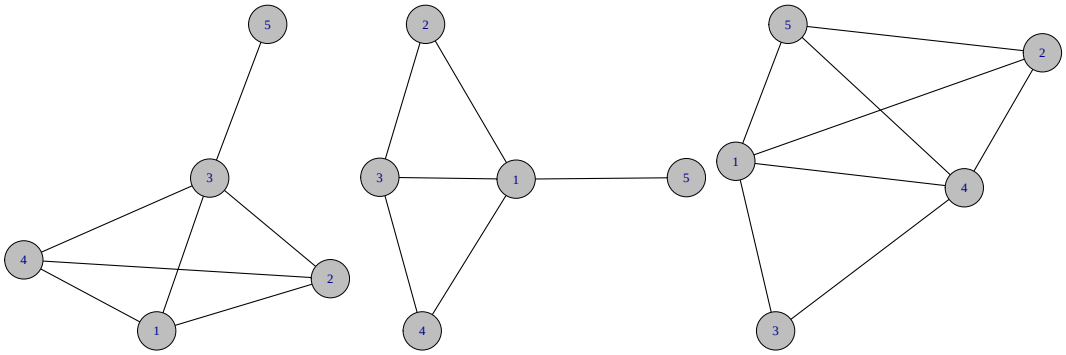
- The actions a_1 and a_2 belong to two different categories of actions, namely probabilistic and deterministic. In a deterministic action, a node v_i selects another node v_j to create an edge, whereas in a probabilistic action there exist probabilities of connecting to different nodes.

- For the rows of $\mathbf{A}_0^1$, it can be seen that the sum is < 1. This means it is feasible that a node might not create an edge even after choosing an action.

- It can be clearly seen that the condition for no multi-edges or self loops is enforced. All corresponding entries have zeros in $\mathbf{A}_0^1$ and $\mathbf{A}_0^2$.
- It should be noted that the matrices must remain symmetric in case of undirected networks.



(a) Starting network G_0.



(b) Three networks synthesized at $t = 1$ as explained in the example above.

Figure 4.6. Network synthesis process of ABNG: It must be noted that the networks shown here have different edges.

4.7 Proposed implementation

4.7.1 Assumptions

Before going into further details of the optimization algorithm, we list the simplifying assumptions in the current implementation of ABNG:

1. *Input network types:* It is assumed that all of the target and synthesized networks are simple graphs, i.e undirected with no self edges. The networks considered for experiments were also unweighted and unlabelled. This does not imply that ABNG is not applicable to such networks.

2. *Network objectives:* No community-specific objectives were considered, although communities are likely in real-world networks. If needed, special objectives can be added to the optimization framework to synthesize networks with more specific community structure.

3. *Starting network:* The current implementation of ABNG needs a starting network with n nodes as input. Furthermore, this starting network cannot be empty because some actions can potentially become undefined due to the lack of any network characteristics. For example, an action based on preferential attachment according to degree of a node would essentially be equivalent to randomly selecting a node in case of an empty network because each node will have a degree of 0. To tackle these issues, we create G^0, a sparse starting network, as input.

4. *Fitness:* As seen in Algorithm 1, even though the network is built over multiple iterations per node, ABNG does not evaluate characteristics for these interim networks, i.e., network comparison is performed only after termination of the synthesis algorithm.

5. *Types of actions:* As discussed in Section 4.3, the synthesis algorithm in the current implementation is restricted to adding edges only. This is a reason-

able assumption because we only consider static target networks, which can be synthesized using this restricted subset of actions.

6. *Stopping criteria:* The network synthesis process of Algorithm 1 is terminated when the number of edges in the network being synthesized is equal to the number of edges in the target network.

7. *Linear network growth:* Due to a single target network being given as input, linear network growth (i.e., one node and/or edge is added to the network at each iteration) is imposed.

8. *Static action probabilities:* We also assume that the generative process is stable, i.e., the probabilities do not change during network synthesis. This is a reasonable assumption as the goal is to model the generative process based on a single snapshot of the input network.

4.7.2 The action set

An action for a node v_i builds edges, all of which have v_i as one endpoint. An action provides a well-defined strategy for selecting the other end v_j of edge (v_i, v_j). This implies that the actions in ABM can only allow local topological changes in the network as a node is assumed to only build edges to other nodes but cannot create edges between other nodes. Every action for node v_i returns another node v_j with probability $\hat{p}_i^j$ to form edge (v_i, v_j):

$$a(V|i) : v_i \rightarrow (v_i, v_j) \quad \text{w.p.} \quad \hat{p}_i^j, \tag{4.3}$$

where an edge (v_i, v_j) is inserted into G by v_i. This approach also enables an action (and hence ABM) to insert edges that can be directed, weighted, self edges, etc. In the current implementation, the actions are limited to adding a single undirected edge.

A question that is central to a successful implementation of ABNG is: *How many actions should be included in the action set?* There is no direct answer for such a question, however, having too many actions will increase the number of parameters (size of the action matrix) and hence make solving the optimization problem more difficult and time-consuming. This might even lead to degenerate actions and consequently complicate the generator. Actions form an integral part of ABNG. They collectively serve as the mechanisms responsible for network synthesis. Hence, choosing a holistic or sufficient set of actions is crucial for implementing ABNG. The current implementation of ABNG uses eight actions belonging to four different categories:

- **Preferential attachment using network centrality measures** - This is analogous to the Barabási-Albert Model [12], which uses node degree as an action to connect to different nodes. Using this action, a node connects to *important* nodes with higher probability, where importance is calculated using network centrality measures. We use degree (PAD), average neighbor degree (PAND), PageRank (PAPR) and betweenness (PAB) as centrality measures for four different actions. An important property of this set of actions is that $\hat{p}$ is the same for all the nodes.

- **Triadic closure** - This action (TC) connects a node to another node that is a neighbor of its neighbor. It captures the phenomenon: *a friend of my friend is also my friend*. Also, triangles form a commonly encountered structure in real-world networks [1]. In case a node has multiple second neighbors, each one has an equal probability of getting selected.

- **Similarity-based actions** - These provide a basis for nodes to connect to similar nodes, another phenomenon observed in most real-world networks [1]. Inverse log-weighted (SLW) and Jaccard similarity (SJ) measures are used in this implementation.

- There is another action (NA) that does not connect the current node to any other node. As we can see from Algorithm 1, ABNG visits every node in the

network to form new edges, and this action exempts a node v_i from making a connection, i.e. $\hat{p}_i = 0$.

For an undirected network $G = (V, E)$, the Big-Oh for the actions used in the current implementation are:

1. Preferential attachment on neighbor degree - $O(|V|(|V| + |E|))$
2. Preferential attachment on node degree - $O(|V| + |E|)$
3. Preferential attachment on PageRank - $O(|V|^2)$ for dense and $O(|E|)$ or $O(k|V|)$ for sparse, where k is the average degree
4. Preferential attachment on node betweenness - $O(|V||E|)$
5. Triadic closure - $O(|V| + |E|)$
6. Inverse-log weighted similarity - $O(|V|^2 k_{max})$, where k_{max} is the maximum degree
7. Jaccard similarity - $O(|V|^2 k_{max})$, where k_{max} is the maximum degree
8. No action - $O(1)$

4.7.3 Optimizing the action matrix

Pareto Simulated Annealing [209] is the multi-objective analog of simulated annealing. It provides a procedure to search for a set of solutions to a multi-objective combinatorial optimization problem. Due to multiple objectives in the problem formulation, more than one efficient solution can exist. Let D be the set of feasible solutions following the constraints defined in the optimization problem in Equation 4.2. For a solution $\mathbf{M}$, $B(\mathbf{M}) \subseteq D$ is the neighborhood of solution $\mathbf{M}$ obtained by changing the probability of only one action at a time, i.e. only one element of the matrix $\mathbf{M}$ is increased (or decreased) in a single iteration of PSA (while keeping the solution

Algorithm 3 Pareto Simulated Annealing

1: **Input:** $\mathbf{M}_0 \in D : q \times (k+1)$ action matrix
2: $S \leftarrow \emptyset$, $z = 1$
3: **while** $z \leq iter$ **do**
4: $\alpha = e^{-\beta \times z}$ {β is some constant}
5: $\mathbf{M} \in B(\mathbf{M}_{z-1})$ {find an action matrix in the neighborhood of the current solution}
6: $G \leftarrow ABNG(\mathbf{M})$ {Algorithm 1}
7: **while** $S \not\prec \mathbf{M}$ **do**
8: $S \subseteq S \cup \mathbf{M}$ {update set of efficient solutions}
9: $\mathbf{M}_{z-1} \leftarrow \mathbf{M}$
10: $\mathbf{M} \in B(\mathbf{M}_{z-1})$ {update the same element of $\mathbf{M}$ as in line 5}
11: $G \leftarrow ABNG(\mathbf{M})$ {Algorithm 1}
12: **end while**
13: $\mathbf{M}_z \leftarrow \mathbf{M} \quad w.p. \quad \alpha$
14: $\mathbf{M}_z \leftarrow \mathbf{M}_{z-1} \quad w.p. \quad 1 - \alpha$
15: $S_z \leftarrow S$
16: **if** $z = iter$ & no change in S for 100 iterations **then**
17: **return** S
18: **else**
19: $iter = iter + 100$
20: **end if**
21: $z = z + 1$
22: **end while**
23: **return** G

feasible as defined in the constraints in Equation 4.2). Our implementation of PSA (see Algorithm 3) starts with an action matrix $\mathbf{M}_0$ generated uniformly at random from D to prevent any bias due to a starting point. A new solution $\mathbf{M}_z \in B(\mathbf{M}_{z-1})$ is

generated using the procedure shown in Algorithm 3, where $B(\mathbf{M}_{z-1}) \subseteq D$ is the set of feasible solutions that can be reached from $\mathbf{M}_{z-1}$ by making a simple move that can only increase (or decrease) the value of one element of $\mathbf{M}_{z-1}$, while maintaining the feasibility of the solution.

Owing to the multi-objective nature of the problem, we need to maintain a set S of potentially efficient solutions. A solution $\mathbf{M} \in D$ is efficient (Pareto-optimal) if there is no $\mathbf{M}' \in D$ such that $\forall j\ Y_j(\mathbf{M}') \leq Y_j(\mathbf{M})$ and $Y_j(\mathbf{M}') < Y_j(\mathbf{M})$ for at least one j. S is updated with $\mathbf{M}_z$ if it is not Pareto dominated[1] by the solutions in S. The algorithm repeatedly increases (or decreases) the value of the same element of $\mathbf{M}_z$ while it continues to be non-dominated by the current set of potentially efficient solutions (Algorithm 3 lines 7-12). This is particularly useful for optimizing the action matrix because it was observed that nodes tend to connect based on simple decisions leading to solutions where a subset of actions have high probability, whereas other actions might have zero or near-zero probability (see Tables 4.2 and 4.3). Finally, if $\mathbf{M}_z$ is not added to the Pareto front, the algorithm moves to the new solution with probability α that decays exponentially with the number of iterations, otherwise it returns to the previous solution $\mathbf{M}_{z-1}$.

The optimization procedure could terminate at a local optima or some other non-optimal stationary point. To help overcome this issue, we use multiple starting points for the optimization process, i.e. the procedure starts with more than one $\mathbf{M}_0$.

In the current implementation, we attempt to find the most simple generator (in terms of how nodes make decisions for connecting to other nodes) for the target network. In terms of the action matrix, the simplest generator can be obtained when all nodes have a common mechanism (same probability distribution over actions) for forming edges, and hence $\mathbf{M}$ is assumed to have dimensions $1 \times (k+1)$. This is followed by dynamically adding more node-types (or rows in the action matrix) to the current solutions. In the algorithm implementation, once a Pareto front (or a set

[1]Pareto dominance is defined in the same way as Pareto optimality, the only difference being that $\mathbf{M}' \in S$, i.e. the new solution, is compared to the current set of potentially efficient solutions.

of potentially efficient solutions) S is found for $\mathbf{M} : 1 \times (k+1)$, a solution is picked at random from S and a new row (generated at random from D) is added to $\mathbf{M}$, making it a $2 \times (k+1)$ action matrix and so on. For initialization, $\bar{P}$ containing the probability of choosing rows in $\mathbf{M}$ is generated uniformly at random. For a $q \times (k+1), q > 1$ action matrix, we assume that only the newly added q^{th} row and $\bar{P}$ need to be optimized because the remaining rows have been optimized for the target network in previous steps. In other words, this procedure of adding new rows to $\mathbf{M}$ implies that the previous rows contain information about how nodes selected actions when their choices were restricted and adding new rows will provide them with more choices. $\bar{P}$ allows nodes to choose among various rows, which may change when new rows are added and hence needs to be considered in the optimization framework.

Though this procedure might restrict the search space for the action matrix, it suffices to provide some insights about the applicability and ability of the action-based approach to synthesize networks. As described in Section 4.3, each row of the action matrix reflects the mixed strategies used by nodes for making connections, and the aim is to find the minimum number of such rows that corresponds to the smallest parameter space capable of synthesizing the target network using ABNG. More rows are added to the action matrix until at least one of the following criteria are met:

- The newly added q^{th} row has probability close to zero in the converged solutions, i.e. $\bar{P}_q \approx 0$. This implies that the network structure can be explained equally well using a smaller action matrix. A threshold value of 0.05 is used in the present implementation.

- The newly added q^{th} row is similar to one of the previous rows i.e., $\mathbf{M}(q,:) \sim \mathbf{M}(b,:), 1 \leq b \leq q-1$. Similarity between two rows can be defined using any vector similarity measure. This implies that the two strategies are practically equivalent. In case such an event occurs during optimization, the $\bar{P}$ values for both the rows are added (and assigned to the initial row) and the copy row is deleted i.e. $\mathbf{M}(q,:)$ is deleted and $\bar{P}_b = \bar{P}_b + \bar{P}_q$.

- Solutions from a smaller action matrix strictly dominate the new solutions, i.e. $S(\mathbf{M}_{b\times(k+1)}) \succ S(\mathbf{M}_{q\times(k+1)})$, $1 \leq b \leq q-1$. This means adding new rows does not improve the quality of the synthesized networks for the given set of actions.

The modified version of Pareto Simulated Annealing used here also incorporates an adaptable number of maximum iterations (Algorithm 3 lines 16-20). After every 100 iterations, the previous 100 solutions are checked for improvement in any of the objectives. If $S_z \neq S_{z-100}$, 100 more iterations are allowed, otherwise the process is terminated at a maximum of 1000 iterations. The strategy resulted from the peculiar behavior of the optimization process that can be observed in Figure 4.13, where convergence of solutions is observed before reaching the maximum number of iterations. This adaptable approach aids the algorithm in identifying potential local optima and consequently stopping the optimization process.

4.8 Additional results

4.8.1 Comparing action matrices with multiple rows

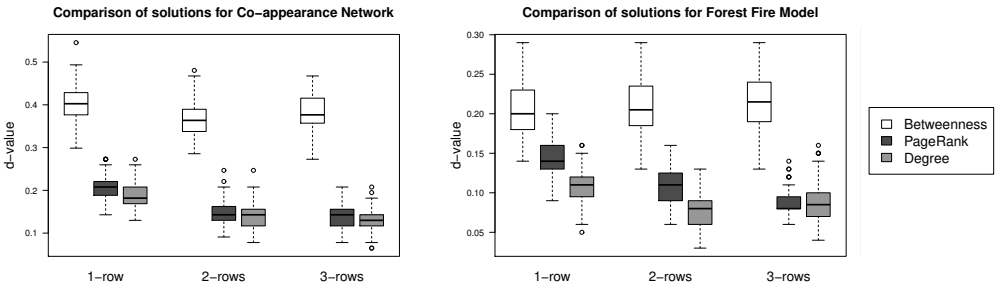


Figure 4.7. Improvement using 2-rows: This box plot highlights the effect of adding additional rows to the action matrix. For the networks shown here, adding a second row to the action matrix produced solutions that Pareto dominated the solutions from a 1-row action matrix. Each box plot shows statistics from 100 networks generated using ABNG.

Figure 4.7 compares results obtained from optimized 1-row, 2-row and 3-row action matrices for the forest fire model and the real-world network of co-appearances of characters in Victor Hugo's novel "Les Miserables". Adding a second row to the action matrix improved the solution quality, but adding a third row did not improve the quality of the synthesized networks with respect to the objectives taken into account here. In other words, the 2-row solutions Pareto dominated the 1-row solutions, while the 3-row solutions lead to a quality-of-fit that was equivalent to the 2-row solutions and hence showed no improvement. Also, the 3-row solutions had $\bar{P}_2 \approx 0$, which implies that there were only two distinct nodes-types.

4.8.2 ABNG for real networks

19 real-world networks were also considered for empirical evaluation of ABNG and details are shown in the table B.1. Radar plots for the networks synthesized using action matrices obtained as solutions for these real-world networks can be seen in Figure 4.8. Real-world networks will likely not be simply described like the human-devised models in the previous experiments, i.e., the action matrix will have more rows. The estimated action matrices can potentially provide insights about the structure of these networks like, how many types of nodes exist in the network, how they weigh actions to form edges etc. Description of the action-based model for five of these networks can be seen in Section 4.4. Figure 4.8 also compares the result of ABNG with some other network generators, namely Chung-Lu [47, 48] and ERGM [52], which were fit to the target network. From the comparison it can be concluded that:

- Networks synthesized using ABNG show the most resemblance to the target networks when evaluated based on the network properties considered here.
- Unlike other models, the output parameters obtained using ABNG (action matrices) can provide a compact representation of structure of the target networks.

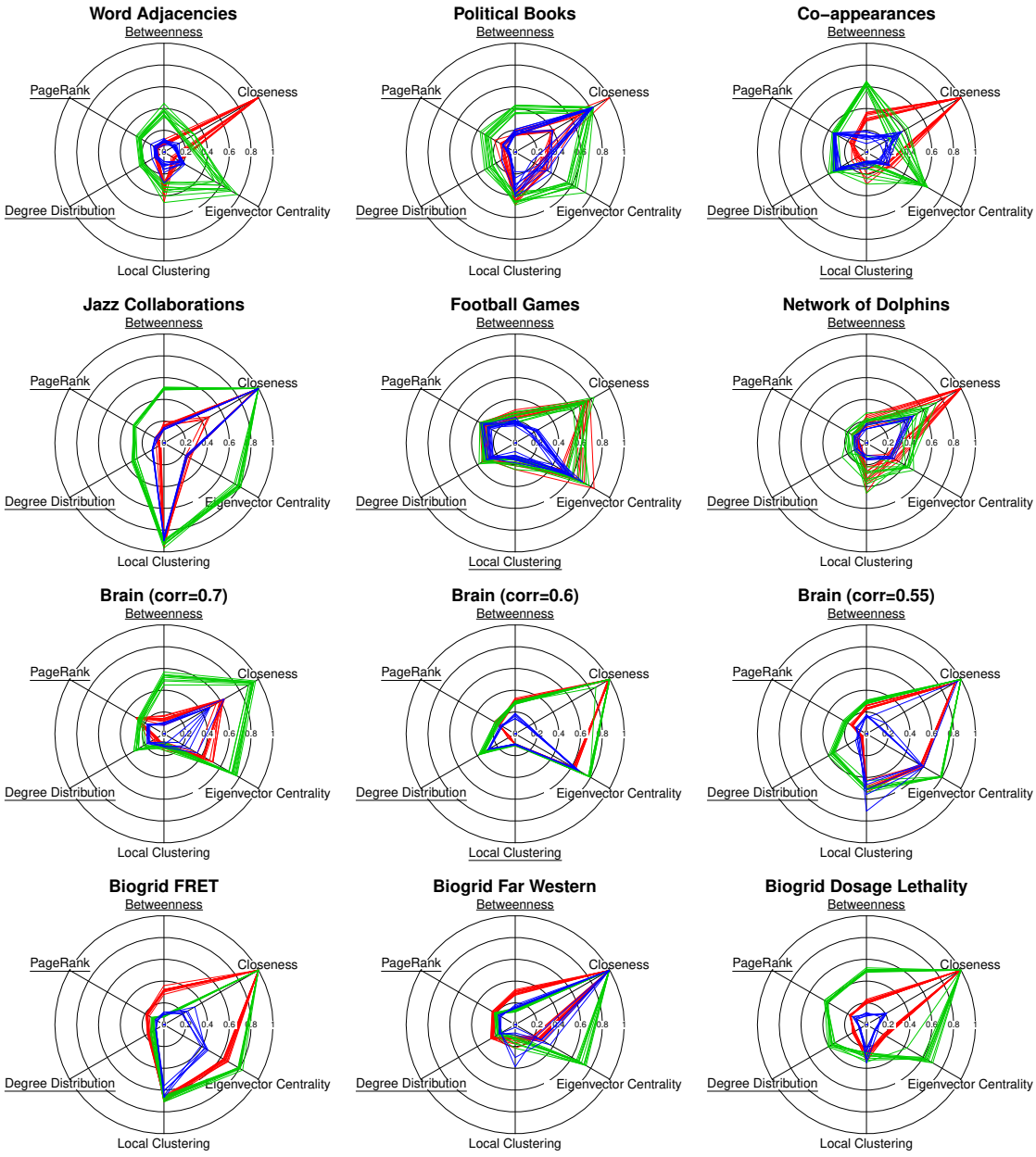


Figure 4.8. Results obtained from the optimized ABNG models for the real-world networks considered here. Network properties that were used for optimization are underlined in the radar plot. The plots show KS test *d-statistic* with the outer circle showing value of 1 (maximum possible value). The lower the value, better is the synthesized network. Figure continues on next page.

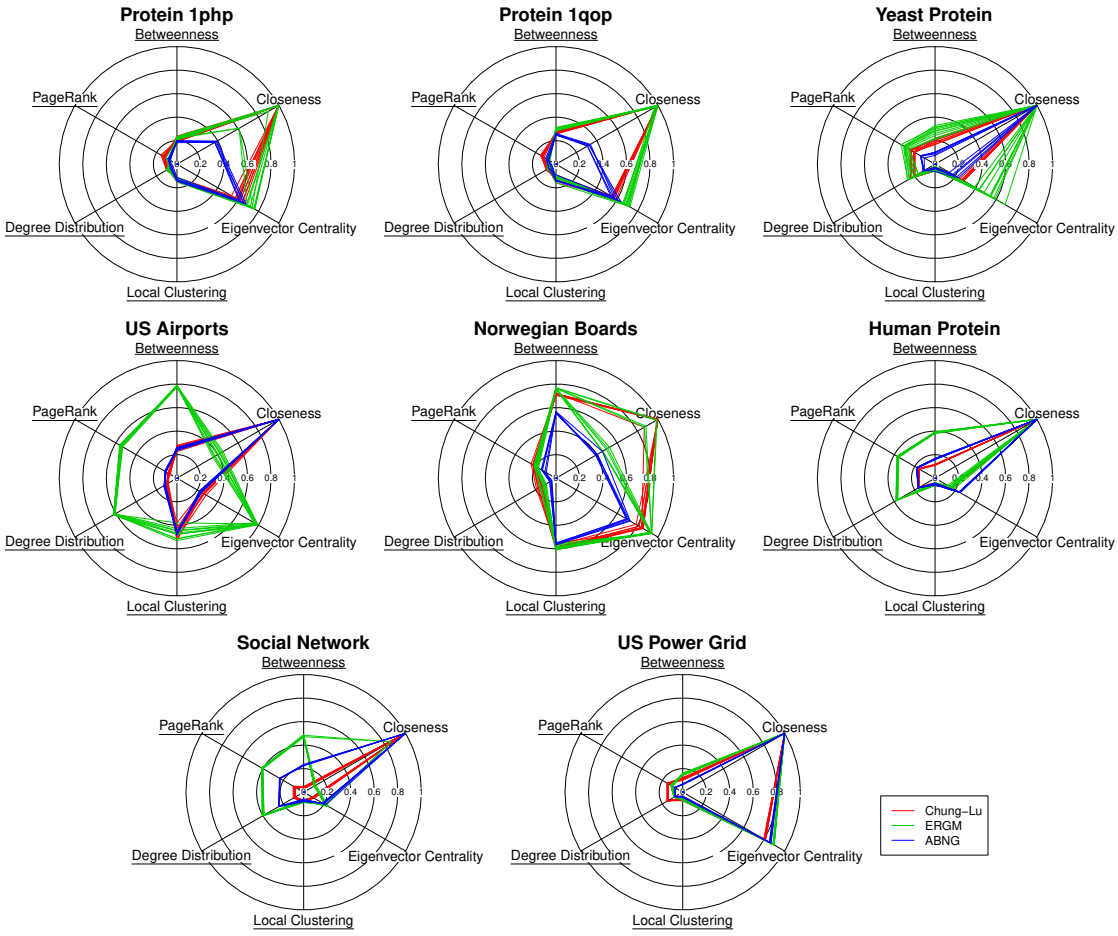


Figure 4.8. Results obtained from the optimized ABNG models for the real-world networks considered here. Network properties that were used for optimization are underlined in the radar plot. The plots show KS test *d-statistic* with the outer circle showing value of 1 (maximum possible value). The lower the value, better is the synthesized network.

4.8.3 Spectral goodness of fit

A new statistic to evaluate how well a network generator explains the structure of the pattern of ties in the target network was proposed in [168]. The current version of the Spectral Goodness of Fit (SGOF) statistic is limited to only unlabeled and undirected networks, which matches with the type of networks synthesized using

ABNG. Because of its simplicity, we use SGOF as a proxy measure for goodness-of-fit for the synthesized networks and do not consider it in the optimization process of ABNG.

The approach calculates the Euclidean Spectral Distance ($E\bar{S}D_{G^*,G} = ||\hat{\lambda}^{G^*} - \hat{\lambda}^{G}||$), where $\hat{\lambda}^{G^*}$ and $\hat{\lambda}^{G}$ are the normalized spectra (of the Laplacian) of networks G^* and G. The spectral goodness of fit (SGOF) can be then obtained by:

$$SGOF = 1 - \frac{E\bar{S}D_{G^*,G}}{E\bar{S}D_{G^*,\mathcal{N}}} \tag{4.4}$$

where $\mathcal{N}$ is the null model. For SGOF calculations, the Erdős-Rényi model is used as the null model. The SGOF measures the amount of observed structure (G^*) explained by a fitted model ($\{G_1, G_2, \cdots\}$), expressed as a percent improvement over a null model, where structure means deviation from randomness [168].

SGOF is bounded above by one, which means that the network generator (or fitted model) exactly describes the target network. Similarly, an SGOF of zero means that synthesized networks are only as good as the random networks, whereas a negative value signifies that the null model is a better approximation of the target network as compared to networks synthesized using the network generator. Figures 4.9 and 4.10 show the SGOF values obtained from 100 networks synthesized using ABNG for both human-devised and real-world networks. The networks for ABNG are synthesized using the action matrix corresponding to the point closest (based on 1-norm distance) to the origin in the Pareto front. Note that SGOF values close to zero are observed for the Erdős-Rényi network because it is itself used as the null model.

To maintain consistency of using a dissimilarity metric ranging between 0 and 1, we transform the obtained SGOF value by using the function $y = 1 - 2^{x-1}$, where y gives us the transformed dissimilarity and x is the SGOF value obtained using Equation 4.4. This transformation is used in the heat maps of Table 4.5.

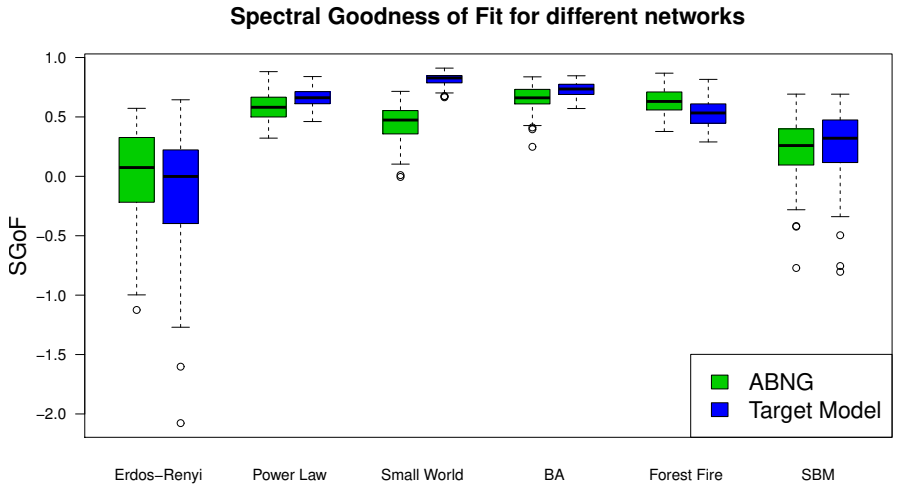


Figure 4.9. Spectral Goodness of Fit for human-devised generators: The plot compares SGOF values for networks synthesized using ABNG and the target model. Each box plot corresponds to SGOF values obtained from comparing 100 synthesized networks with the target network. It can be seen that for most cases ABNG performs as well as the target model.

4.8.4 Scaling with network size

An experiment was also performed to get empirical insights about scalability of the synthesis algorithm described in Algorithm 1. As described in Section 4.2.1, the complexity of any given synthesis algorithm depends on the input action matrix together with the size of the network. To understand the relation between network synthesis time and size of the network (number of nodes), an experiment was performed where each trial used a different action matrix and the size of the network was increased. Also, the mean degree of the networks was kept constant ($\bar{d} = 6$) as the number of nodes in the network was varied. Mean degree of 6 was chosen based on observations of mean degree of real-world networks shown in Table B.1. Results shown in Figure 4.11 provide preliminary insights that network synthesis time scales quadratically with number of nodes in the network. The fitted quadratic model predicts that a

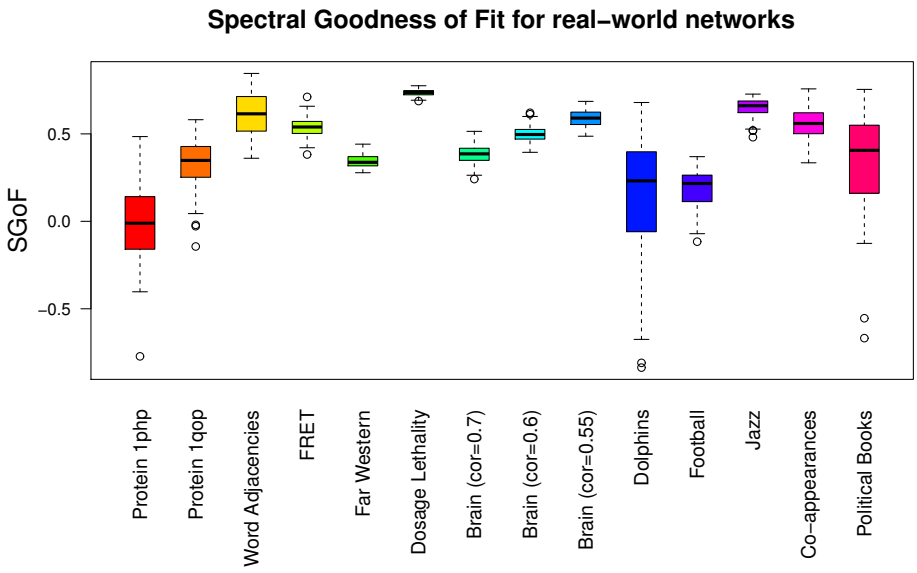


Figure 4.10. Spectral Goodness of Fit for real-world networks: SGOF values obtained for different real-world networks synthesized using ABNG. Each box plot corresponds to SGOF values obtained from comparing 100 synthesized networks with the target network.

network with 100,000 nodes and $\bar{d} = 6$ will require around 1 hour for synthesis. For network of each size, 20 networks were synthesized parallely using ABNG-PA(1) and the total CPU time was recorded. The plot shows mean times for each network size and action matrix. The system used consisted of 10-core Intel Xeon-E5 CPUs with a frequency of 2.60GHz.

4.8.5 Starting network variations

In the experiments conducted so far, it is assumed that the starting network is obtained by sampling $0.7 \times n$ edges from the target network. In this section, different proportion of links are sampled from the target network to see how ABNG performs when provided with different starting networks. For this, we consider four different target networks and five different fraction of links, as shown in Figure 4.12. The synthesized networks are evaluated based on five network measures and corresponding

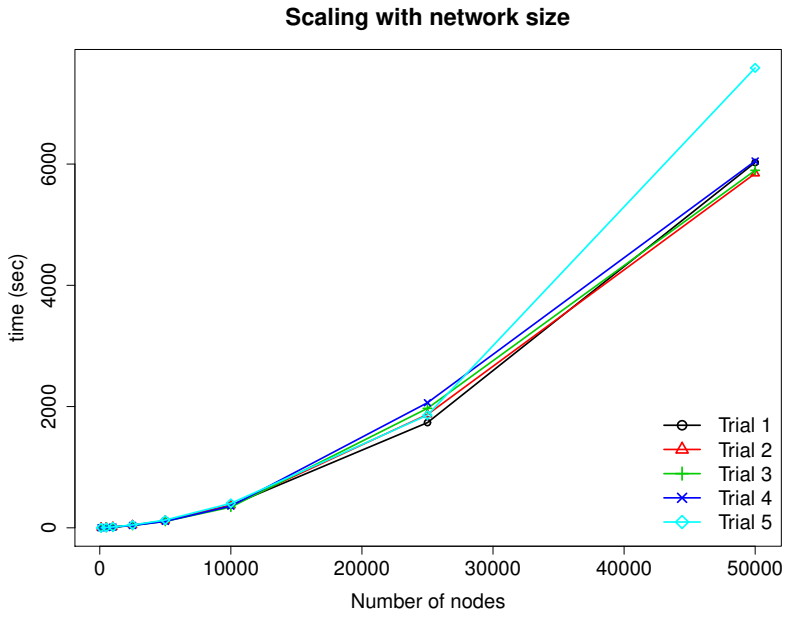


Figure 4.11. Scaling of ABNG-PA(1) with network size: Five different action matrices were used as input for networks containing different number of nodes. All networks have a mean degree of 6 and networks with 100, 500, 1000, 2500, 5000, 10000, 25000 and 50000 nodes were used.

heat maps are shown in Figure 4.12. For each target network, the solution closest to the origin was chosen as the action-based model to synthesize 20 networks and average values for each measure are recorded in the heat maps. The heat maps show that the quality of the synthesized networks does not depend much on the fraction of links in the starting network. Only when the fraction of links in the starting network is $0.25 * n$, a consistent drop in quality is observed for each target network.

In earlier experiments, a fixed starting network was used for synthesis throughout the optimization process. While varying the fraction of links in the starting network, we also relaxed the assumption of using a fixed starting network and sampled a different starting network each time the algorithm was used to synthesize a network. Results indicate that ABNG can synthesize networks having similar quality-of-fit

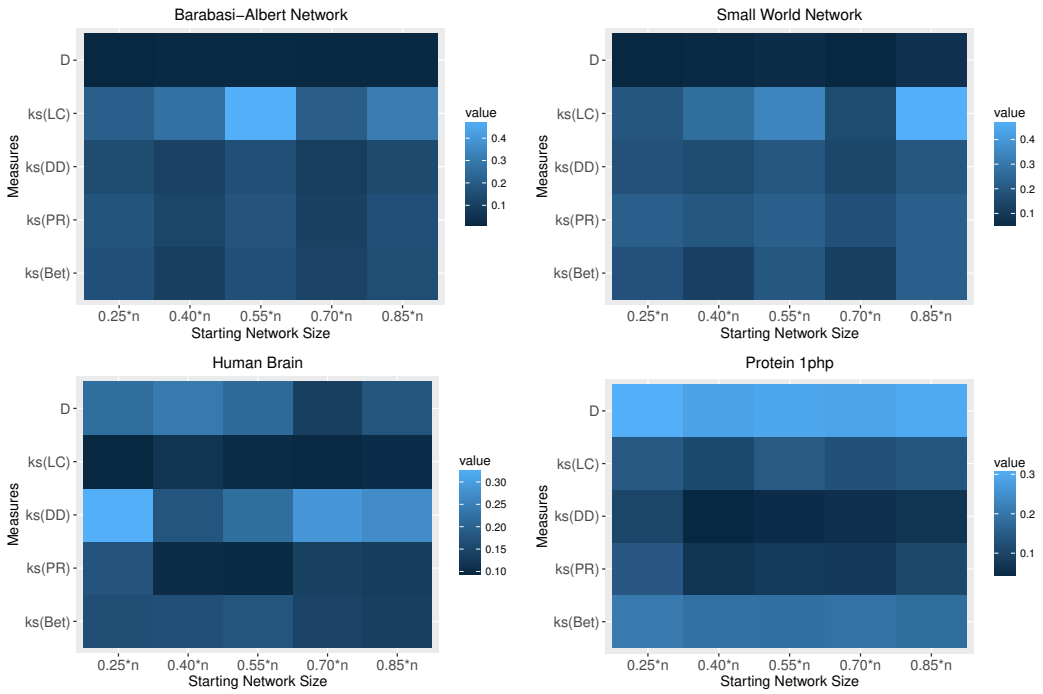


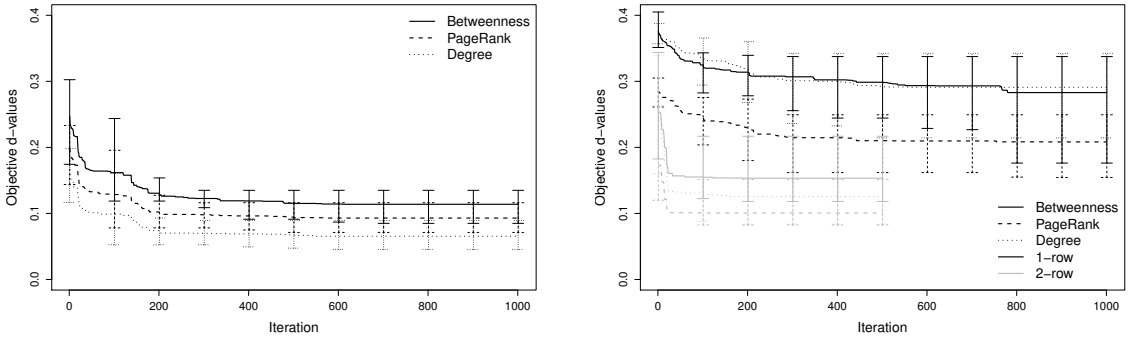
Figure 4.12. Starting network variation: Each generator synthesizes 20 networks, and the mean dissimilarity values for different measures are recorded in the heat maps. The lower the value, better is the synthesized network.

when the assumption of a fixed starting network is relaxed, hence providing evidence about the robustness of the action-based approach.

4.8.6 Analyzing the action matrix

Here, the evolution of the action matrix and the objectives during the PSA iterations are examined to get a better understanding of the process of learning an action matrix in ABNG. Figure 4.13 illustrates two representative examples of evolution of solutions for 1-row and 2-row action matrices when using PSA for optimization in ABNG. The optimization process shows typical characteristics observed in evolutionary multi-objective optimization algorithms, where a lot of improvement is seen in the objective space in the first few iterations and the solutions seem to converge in the

later iterations. Also, it can be seen that the best solutions are found before reaching the maximum number of iterations and can be observed when the curves become flat in Figure 4.13.



(a) Iteration plot for a 1-row action matrix.

(b) Iteration plot for both a 1-row limit and 2-row limit of the action matrix for a network synthesized using the forest fire model.

Figure 4.13. Iteration plots for ABNG action matrix optimization using PSA: The plots show evolution of specific objectives versus the number of PSA iterations. Each line depicts averaged results over 5 restarts for PSA. Error bars show the minimum and maximum values obtained in different iterations of PSA.

Examining different solutions obtained for the same target network leads to an interesting observation. Table 4.4 shows the cosine similarity of 5 different Pareto optimal action matrices ($\mathbf{M}_1 - \mathbf{M}_5$) for the Barabási-Albert model. As is evident, the solutions are very similar because of the high cosine similarity values (the difference is likely to be because of the granularity of the optimization approach), and hence ABNG identifies similar underlying mechanism of network formation each time it optimizes the action matrix for a target network. It shows that ABNG consistently finds the same solutions for a target network. Figures 4.14 and 4.15 show different examples for the evolution of a 1×8 action matrix. Figure 4.15a shows a plot when the optimization process starts with a randomly generated starting point, while

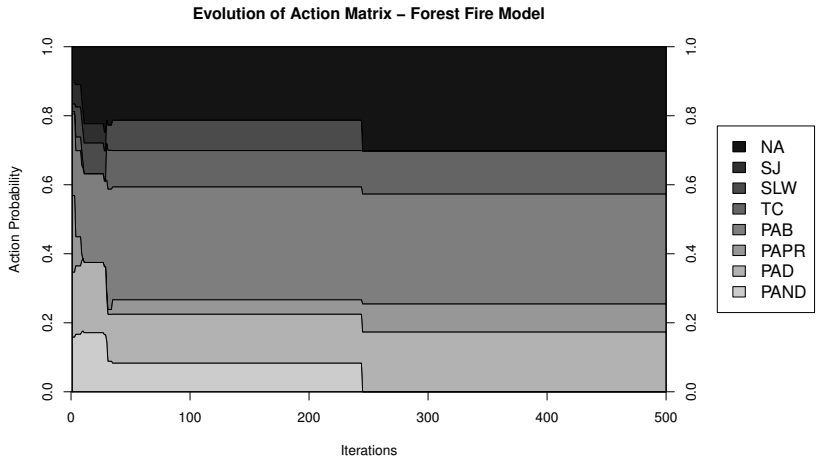
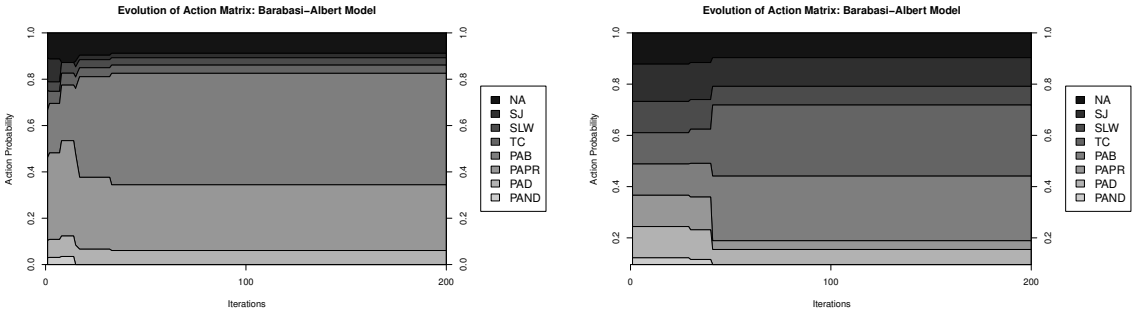


Figure 4.14. Example action matrix evolution: This shows the evolution of an action matrix versus number of iterations of PSA for a network synthesized using the forest fire model.



(a) Starting with a random action matrix.

(b) Starting with uniform probability distribution over actions.

Figure 4.15. Action Matrix evolution: This shows the evolution of an action matrix versus number of iterations of PSA for the Barabási-Albert model.

Figure 4.15b shows the evolution of action matrix when the starting solution had equal probability for each action. Similar to Figure 4.13, the solutions stabilize after certain number of iterations. Also, some actions have very high probability in the final solutions while others might have zero probability. Though the results shown

in Figures 4.13-4.15 are for a few particular networks, they are representative of the other networks that have been considered in this Chapter and hence capture the behavior of ABNG.

Action Matrix	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_3$	$\mathbf{M}_4$	$\mathbf{M}_5$
$\mathbf{M}_1$	1.000	0.996	0.976	0.982	0.985
$\mathbf{M}_2$	0.996	1.000	0.988	0.992	0.996
$\mathbf{M}_3$	0.976	0.988	1.000	0.999	0.993
$\mathbf{M}_4$	0.982	0.992	0.999	1.000	0.997
$\mathbf{M}_5$	0.985	0.996	0.993	0.997	1.000

Table 4.4.
The table shows cosine similarity of five different optimal solutions for the Barabási-Albert model.

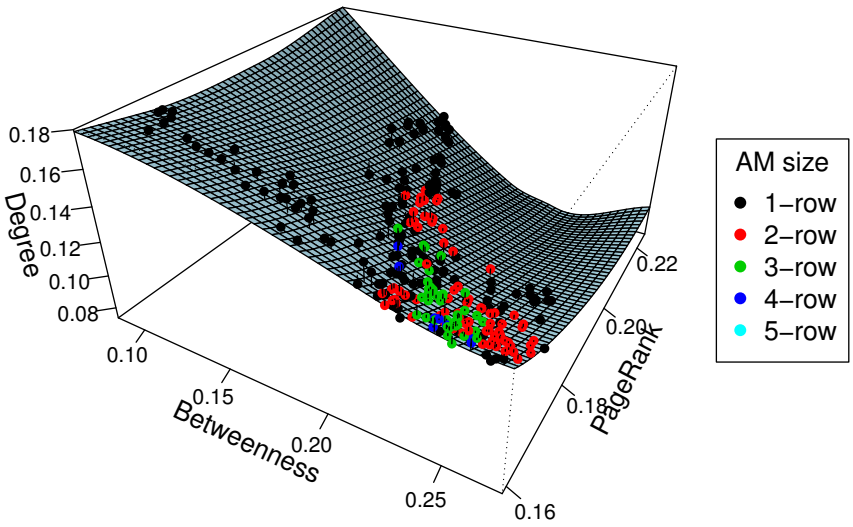


Figure 4.16. A 3D plot is presented to characterize the Pareto front obtained by ABNG optimization process. The plot shows how different solutions (in terms of action matrix size) are spread out in the objective space.

To visualize the diversity of the solutions based on the size of the action matrix, a 3D plot (Figure 4.16) is shown for the three objectives used in the optimization process for the brain network with correlation threshold of 0.7. The example is representative of the distribution of Pareto optimal points obtained when ABNG is optimized for a target network. It is observed that the solutions are more spread out (or scattered) for smaller action matrices (1-row and 2-row) and the solutions seem to be more concentrated at a particular region when considering larger action matrices.

4.8.7 Sensitivity analysis of the action matrix

Another set of experiments involved performing a sensitivity analysis of the action matrices obtained from the optimization process. This helps us understand how the uncertainty in the output of ABNG can be apportioned to different sources of uncertainty in the action matrix. Two types of analysis were done:

1. Change one-variable-at-a-time (OAT), i.e. independently changing the probability of each action by 10%. This captures change in the output due to a change in probability of using a single action and provides evidence for the stability of the outcome. Radar plots for the four different target networks with an optimized 1×8 action matrix are shown in Figure 4.17. It is clear that there is very little variation in the synthesized network properties especially for the properties considered in the optimization framework (underlined in the plot). This is evident from the radar plots as the lines corresponding to change in probability of different actions overlap each other.

   Sensitivity of the action matrix was also tested for a 2-row action matrix obtained for the forest fire model network. In this scenario, the same approach was used to separately perturb the first row, second row and $\bar{P}$ associated with the action matrix. Again, the radar plots of Figure 4.18 provide evidence for the stability of the outcome even in the case of a bigger action matrix.

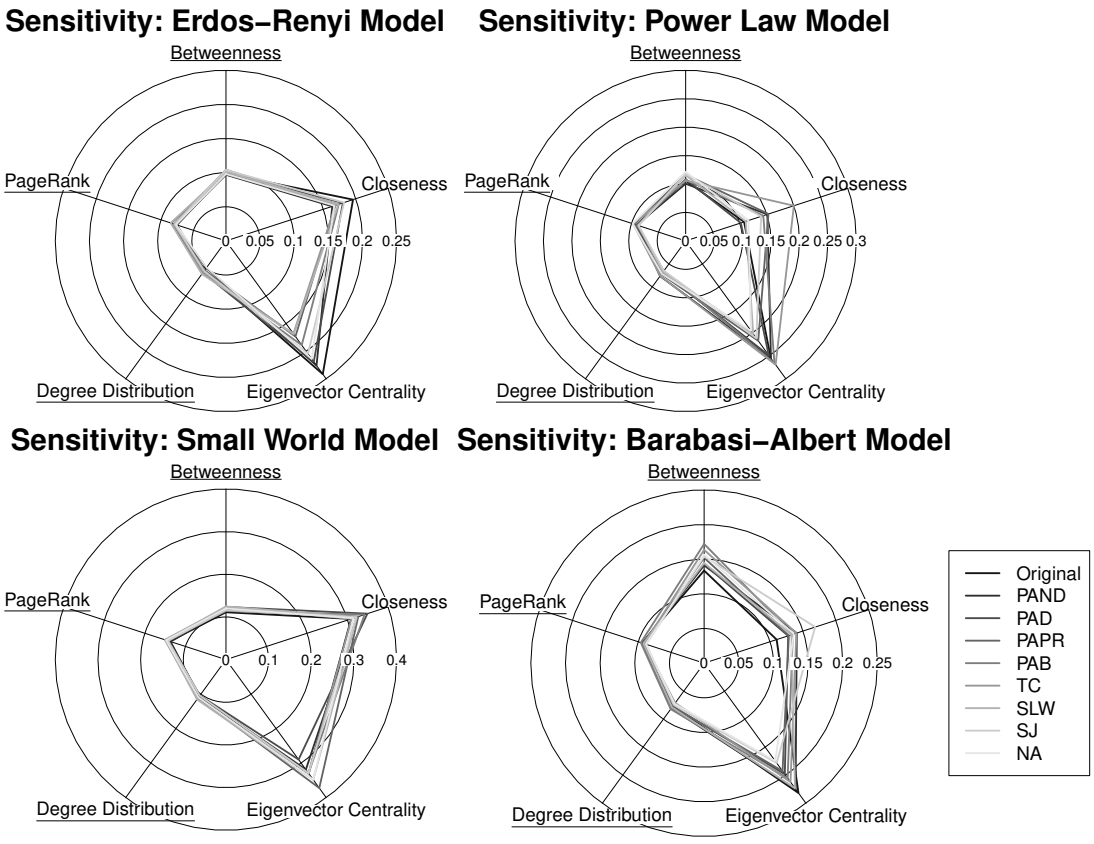
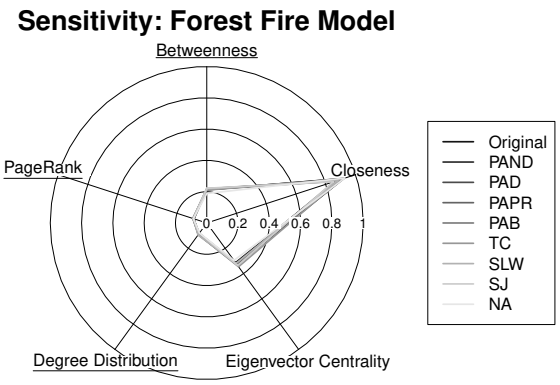
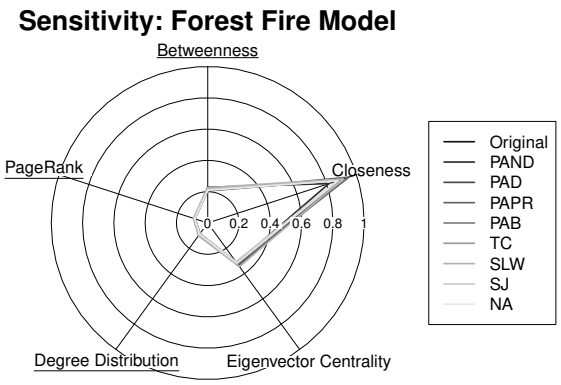


Figure 4.17. Sensitivity Analysis: The plots show KS test *d-statistic* for different network properties. Network properties that were used as objectives for optimization are underlined. Each line corresponds to average of 20 networks synthesized using a 1-row action matrix perturbed using OAT approach.

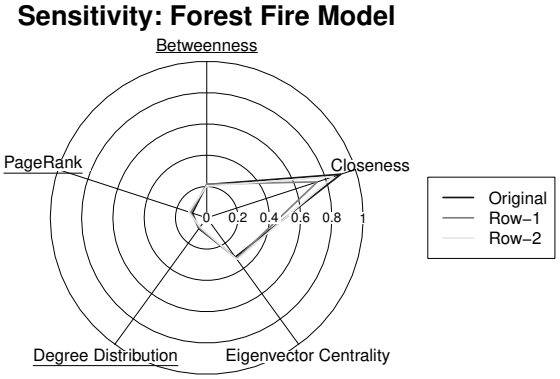
2. The one-at-a-time (OAT) approach does not fully explore the input space since it does not take into account the simultaneous variation of input variables. This means that the OAT approach cannot detect the presence of interactions between input variables. Instead, the second test varies the probabilities of all the actions simultaneously. Radar plots for this version of the sensitivity analysis can be seen in Figure 4.19 and 4.20 for the 1-row and 2-rows cases respectively. For each network, the test is performed five times with distinct variations in the action matrix. Clearly, even using this method there is a lot



(a) Sensitivity Analysis for the first row.



(b) Sensitivity Analysis for the second row.



(c) Sensitivity Analysis for $\bar{P}$.

Figure 4.18. Sensitivity Analysis: The plots show KS test *d-statistic* for different network properties. Network properties that were used as objectives for optimization are underlined. Each line corresponds to average of 20 networks synthesized using a OAT perturbed action matrix.

of overlap in the synthesized network properties for the different variations of the action matrix, especially for the properties considered as objectives for the optimization. The same is true for the case of a 2-row action matrix considered in Figure 4.20.

Results for both the cases reflected the robustness of the solutions obtained by showing that making small perturbations to the action matrix had little effect on the output of the synthesized networks. This provides preliminary evidence for the

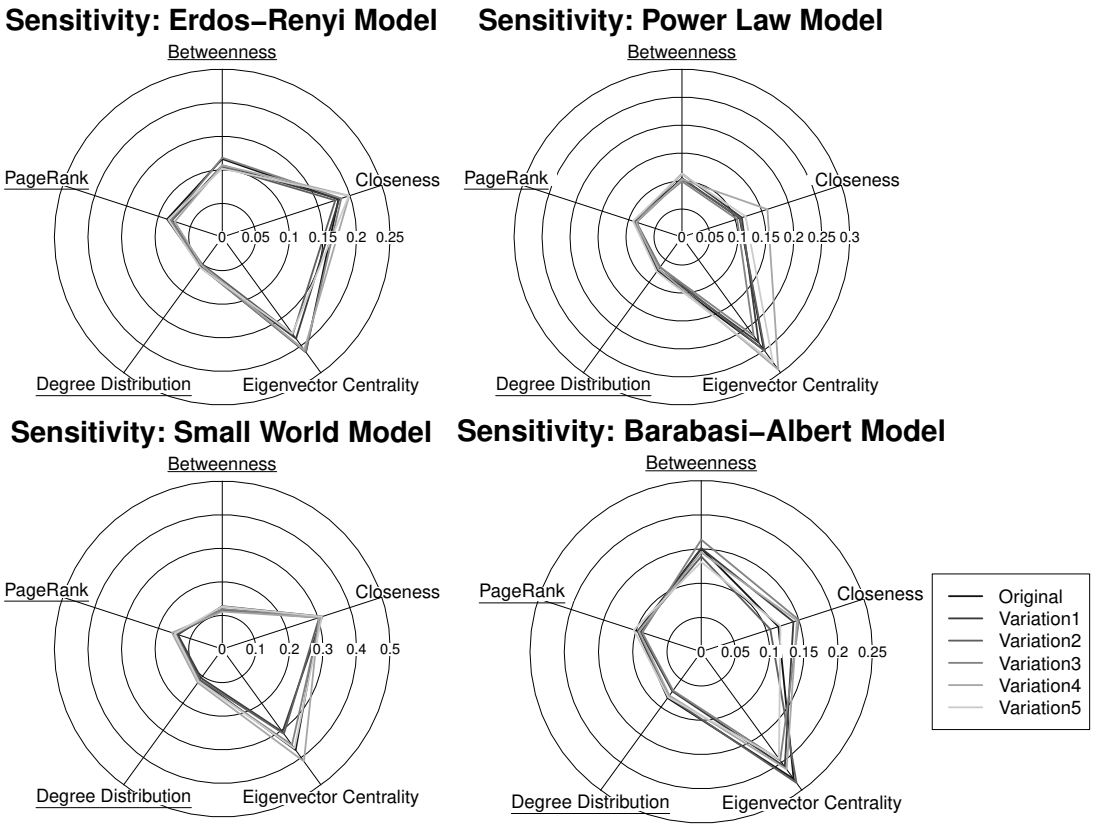
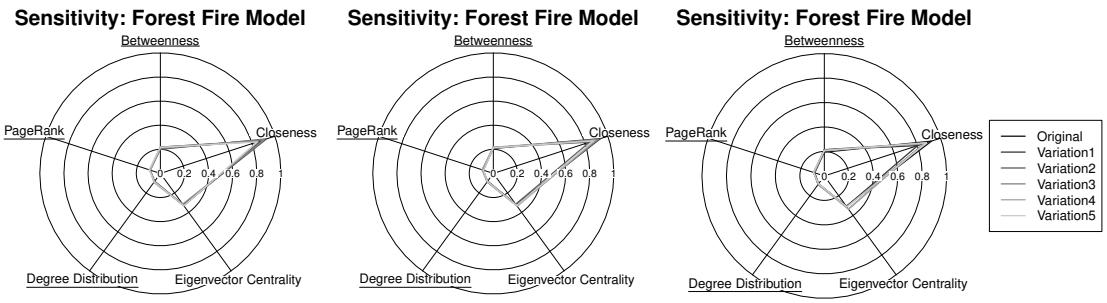


Figure 4.19. Sensitivity Analysis: The plots show KS test *d-statistic* for different network properties. Network properties that were used as objectives for optimization are underlined. Each line corresponds to average of 20 networks synthesized using a simultaneously perturbed action matrix.

continuity in mapping the action matrix to the objective space and that the quality of the synthesized target networks is robust to small changes in parameters.



(a) Sensitivity Analysis for the first row.

(b) Sensitivity Analysis for the second row.

(c) Sensitivity Analysis for $\bar{P}$.

Figure 4.20. Sensitivity Analysis: The plots show KS test *d-statistic* for different network properties. Network properties that were used as objectives for optimization are underlined. Each line corresponds to average of 20 networks synthesized using a simultaneously perturbed action matrix where.

5. MODELING TOPOLOGICALLY RESILIENT SUPPLY CHAIN NETWORKS

The ubiquity of supply chains along with their increasingly interconnected structure has ignited interest in studying supply chain networks through the lens of complex adaptive systems. A particularly important characteristic of supply chains is the desirable goal of sustaining their operation when exposed to unexpected perturbations. Applied network science methods can be used to analyze topological properties of supply chains and propose models for their growth. Network models focusing on the critical aspect of supply chain resilience may provide insights into the design of supply networks that may quickly recover from disruptions. This is vital for understanding both static and dynamic structures of complex supply networks, and enabling management to make informed decisions and prioritizing particular operations.

In this chapter, we propose an action-based perspective for creating a compact probabilistic model for a given real-world supply chain network. The mechanistic nature of our model makes it easy to incorporate domain knowledge. Since the modeler is in control of the mechanisms to include, one is able to encode relevant domain knowledge of known or hypothesized interactions between actors in the system as mechanistic rules. The action-based model for supply chains consists of a set of rules (actions) that a firm may use to connect with other firms, such that the synthesized networks are topologically resilient. The ability to adapt and recover from adverse circumstances is another important feature of complex systems. Consequently, we test the ability of the action-based approach to synthesize resilient supply chain networks, while specifically focusing on the aspect of topological resilience and capturing the heterogeneous roles of different firms in a supply chain by incorporating domain specific constraints [213, 214]. Results analyzing the resilience of networks subjected

to node disruptions show that networks synthesized using the proposed model can generally outperform its real-world counterpart.

5.1 Introduction

Present day supply chain networks (SCNs) are profoundly interconnected structures that emerge from a largely downstream exchange of goods between firms (manufacturers, distributors, retailers, etc.) that are involved in creating a set of final products. Connections are formed or removed as firms use information from a local neighborhood to increase the value they derive from the supply chain without any knowledge of the interconnection structure of the whole supply network. That is, the network itself emerges through the local decisions of firms [215]. Despite this realization, most industrial operations are still built upon overly simplified (often highly linear) models [216]. Other gaps were identified in [217] that suggest a focus on supply chain structure, dynamics and design strategy. Subsequently, there might be a tremendous unlocked potential in supply network efficiency that can be achieved through a complex systems/networks perspective [218,219]. Additionally, three other major challenges have been identified as critical to the study of supply chains through the lens of complex networks [220]: (i) researchers ability to comprehend the complex interactive nature of supply chain formation is limited, especially as the network size increases, (ii) effective metrics for the dynamic nature of supply chain evolution are lacking, and (iii) developing theories to support supply chain design principles in the presence of network adaptation is nontrivial.

Utilizing the knowledge of network science to study supply chain networks was first suggested in [218], where various recommendations for future research directions on bridging the two research areas were laid out. Subsequent examinations of supply chain networks through the lens of network science have primarily focused on analyzing topological characteristics of supply chains and providing summary statistics for describing particular features. This is particularly useful since analysis of topologi-

cal characteristics of the interconnection structure of firms in a supply chain enables managers to reflect on various aspects of the supply chain. For instance, [221, 222] investigated automobile manufacturer supply networks with the aim of understanding the implications of using well known social network measures in the context of supply chains. Similarly, [223] examined the network of automotive firms in southern Italy and discovered high local clustering, while [224] had very similar observations for the Guangzhou automotive network. Though an assortment of comparative investigations have been performed in other industry settings revealing fascinating properties of the networks themselves, consensus on a standard approach for designing supply networks remains generally elusive [217].

While these surveys of real-world supply networks and their reported summary statistics provide insights into predominant characteristics of supply chains, they provide limited insight into the mechanisms by which these networks grow and evolve. A major reason for this stagnation has been a lack of availability of real-world supply network data to study, leading to a significant need for generators (algorithms for creating networks with specific topological properties) capable of synthesizing realistic supply networks that can be utilized to derive deeper insight into their best design principles. The ability of a network generator to synthesize networks with similar underlying summary statistics can help us understand the result of natural and deliberate perturbations on the overall functionality of the supply chain.

We reviewed the literature on network models in Chapter 2, and found that most of the models have limited utility for synthesizing supply chain networks due to the absence of mechanisms to incorporate real-world supply chain constraints. These methods have been shown to be highly unlikely to synthesize networks that share a strong structural resemblance to actual supply networks and are woefully insufficient to study the intricate nature of supply chains, therefore being unsuitable for discovering new design principles [217, 221]. Incorporating constraints on nodes (i.e., firms) is generally outside the capabilities of most existing network generators, which is why

most existing studies have concentrated on general analyses with limited insight into new design principles [217].

A particularly important characteristic of dynamic supply chains is the desirable goal of sustaining their operation when exposed to unexpected disruptions. The goal of supply chain robustness is to sustain operation during such disruptions, whereas the goal of supply chain resilience focuses on designing systems that quickly recover from these disruptions. It is suggested that the definition of resilience and robustness should be established in parallel with the definition of disruption, and [225] shows how some of the important research in supply chains have accomplished this. Robustness and resilience are often used interchangeably in the literature, but in either case the impact fo designing SCNs while accounting for disruptions in the network may be significant. For instance, according to a 2017 report by the Business Continuity Institute [226] 75% of businesses experience at least one supply chain disruption every 12 months (although they suggest the value is likely higher due to underreporting), 22% report cumulative losses of at least $1 million over this time, with 34% reporting at least $270,000. Additionally, 55% reported a loss in productivity, 34% reported that their service was impaired, and 32% reported a loss in revenue. These trends have led to a shift in focus of research from supply chain efficiency to supply chain resilience [225].

Recent work [227] has suggested that topological resilience should be assessed when designing supply chains to ensure sustainable value creation. Robustness and resilience have thus become important areas of study (for simplicity we refer to both as resilience). While [228] was the first to use topology of SCNs for studying resilience, subsequent papers like [229, 230] provide supply chain design insights by examining resilience against both random and targeted attacks. Numerous specialized measures of resilience have also been proposed for supply chains [231], but most analyses concentrate on empirical studies from a centralized context. Outside of the supply chain network community, resilience has also attracted significant attention (see for example [232]). A resilient supply chain should rapidly and effectively respond to

perturbations such as supply or demand fluctuations, or to complete or partial failure of a subset of firms. However, being a complex system, adaptation to changes in the supply chain cannot be dictated by those overseeing or relying upon it. Instead, structural resilience should exist as an outcome of the local linking decisions of various firms within a supply chain without explicit awareness of the overall structure.

A significant amount of research examining complex network models for supply chain networks has focused on using straightforward and conventional strategies as models for SCN evolution. To understand these network models, we first need to introduce two concepts that are essential ingredients of any such approach: (i) each firm belongs to a unique *tier*, which corresponds to its distance (number of hops in the networks) from the consumer in the final supply chain with a restriction that supply-demand relations occur only between firms in adjacent tiers, and (ii) every firm has a fixed *role* in the network, i.e. it is a supplier, manufacturer, consumer etc. Most supply chain models categorize nodes based on their tiers and roles, and then use these attributes to define attachment rules, for example [228, 230] use a heterogeneous preferential linking mechanism that varies based on the role of the firm, [233] proposes an agent-based model that uses firm role and demands to form links, [234,235] use a restricted preferential attachment mechanism based on firm tiers. Others have used more complicated linking mechanisms such as, local selection and preferential attachment [229, 236], random, preferential attachment and similarity [237], and fitness based attachment [238]. Though these models incorporate basic features for modeling SCNs, their simple attachment mechanisms cannot replicate the topology of real-world SCNs, as their capabilities are restricted to synthesizing networks that reproduce a few characteristics like power law degree distributions. Further, only a few of these models focus on creating resilient SCN topologies, without providing any insights into supply chain design principles. Thus, there is still a significant gap in developing a generalizable network growth model that can generate topologies mimicking real-world SCNs [239, 240].

The underlying goal of this research is to define an adequately robust procedure that can synthesize networks exhibiting structural properties observed in real-world SCNs. A network generator is considered to be synthesizing realistic networks if a topological comparison between the synthesized and real networks is, with high likelihood, statistically similar across a subset of user-desired topological characteristics. In this way, the objective isn't to exactly replicate the topology of a given real-world network because there is no utility in synthesizing isomorphic networks as it provides no additional insights. Further, given the input is a single SCN observation, strong assumptions about the dynamical growth of the network need to be made, and the network generator needs to be robust to any such assumptions. Finally, the parameters of the optimized network generator should ideally provide additional insights into the local decisions of the firms that might have lead to creation of the observed network topology.

5.1.1 Main contributions

We focus on modeling of supply chain networks by utilizing the action-based framework [114] for learning a compact probabilistic model for a given material flow SCN. The proposed framework can learn a compact model using a single observation of a real-world supply network and the obtained parameters can be used to synthesize, with high probability, statistically similar networks to a given supply network. We utilize tier information to impose linking constraints among firms, while preserving the tiered structure of the target SCN. The modified network generator that captures critical real-world constraints concerning rules by which firms exchange goods is described in Section 5.2. The novel framework is then used for modeling and synthesizing 10 realistic SCNs in Section 5.3. The applicability of the framework for modeling real-world supply networks is tested, and the resilience of the synthesized networks is analyzed by subjecting them to random and targeted node disruptions. The probabilistic model can also be used to infer growth mechanisms of real-world

SCNs by examining the optimized parameters. Finally, Section 5.4 concludes the chapter with some conclusions and directions for future research.

5.2 Action-based model for SCNs

The problem of discovering a network generator can be posed as a non-linear inverse problem having the form $Y(G^*) \vdash F(\mathbf{M}, \xi)$ (i.e. $F(\mathbf{M}, \xi)$ can be inferred from $Y(G^*)$). The target network G^* and set of p user-desired network structural properties $Y(G^*) = \{Y(G^*), \ldots, Y_p(G^*)\}$ of interest are given as input to the system. Therefore, the goal is to infer $\mathbf{M}$ under the assumption that network formation is performed by the forward operator F. Here, F is an algorithm capable of synthesizing networks based on a random process $\xi \in \Xi$ that can be used to obtain a finite set of networks by repeated simulation of $F(\mathbf{M}, \xi)$.

[241] considered network generation as an optimization-based reverse-engineering problem and concluded that a "good" forward model should consider both the structure and function of the network (although a procedure to accomplish the task was not given). The forward operator of the action-based model assumes that networks emerge through local interaction among nodes that make linking decisions while completely ignoring the global network topology. This assumption is particularly appropriate for modeling supply chain networks because its overall structure can be understood as a self-organizing system that consists of various entities engaging in localized decision making [240]. We thus propose to incorporate domain specific rules and constraints in our action-based approach so that it can be used as a centralized approach for designing robust and resilient SCNs. In the next few sections, we provide details regarding changes that need to be made in the model previously described in Chapters 3 and 4 in order to deal with the intricacies of modeling realistic supply chain networks.

5.2.1 Action set for SCNs

While empirical studies have highlighted that it is highly unlikely that a real-world SCN might have evolved through a single linking mechanism, it is possible to conceive growth and design principles from the global properties of existing SCNs [240]. The action-based framework provides a platform for probabilistically aggregating various local linking mechanisms using a generative algorithm. Each action in ABNG serves as a single linking mechanism, which when combined with an appropriate synthesis algorithm $F(\cdot)$ can synthesize networks exhibiting varying topological characteristics. In the context of SCNs, an action is a decision process that a firm uses to select firms that it should supply its materials to. The supply chain literature provides a rich source for potential decision processes [217, 221, 230, 237, 242, 243], while providing insights regarding how to choose a set of actions that may lead to construction of topologically resilient SCNs. The idea is to carefully choose actions for network synthesis at the micro level such that the resilience of the whole supply network gets mirrored at the macro level. The reason behind this choice is that creation of resilient structures is an expected outcome of the local linking between nodes rather than a goal of the participating firms (a firm is more likely to focus on its operational efficiency). The ability to adapt and recover their previous functionality from adverse circumstances is an important property of complex adaptive systems, and the setup considered in this Chapter will allow us to evaluate the action-based model's ability to design such systems.

Recent research [243, 244] has suggested that existence of power law degree distributions in supply chain networks has a positive affect on its resilience. Preferential attachment mechanisms have been shown to synthesize networks that perform well under random failures and are among the most prominent rules for making linking decisions, hence making them a perfect candidate for actions for SCN modeling [228, 229, 235, 237]. Preferential attachment also leads to creation of networks exhibiting power law degree distributions. A variety of preferential attachment mech-

anisms based on network centrality metrics can potentially lead to creation of a few different hubs, hence dispersing the influential nodes across the overall supply chain. Further, networks with power law degree distributions that are formed by fractal mechanisms show greater resilience against cascading failures as compared to those obtained from the simple preferential attachment mechanisms.

Consequently, the action set A will include *preferentially selecting a node based on its out-degree, in-degree, vertex betweenness and closeness.* These centrality metrics can induce the creation of a diverse range of hub nodes leading to creation of an overall resilient network structure. The use of network properties like betweenness and closeness for preferential linking can be seen as a proxy for more practical information such as price, performance, and quality that are more relevant in the context of supply chains [236, 245]. A fractal mechanism based on *difference in total degree* (resulting in repulsion between hub nodes) has also been shown to produce resilient structures [102, 243], and is included as an action. It is possible that a firm does not prefer one particular firm over another based on the actions described above, leading to an action corresponding to *random selection among the firms satisfying the tier constraints.* An action is also based on *connecting with closer nodes*[1] *with higher probability* [229]. Finally a firm might *choose to not add an edge*, which is the final action in A. It should be noted that in the presence of no edges in the network all actions become equivalent to a random action, i.e. randomly selecting a firm satisfying the tier constraints. This is further explained along with the synthesis algorithm in Section 5.2.2.

5.2.2 Network synthesis

The synthesis algorithm of ABNG (see Section 4.2.1) allows a network modeler to easily integrate domain specific rules or constraints by implementing a problem specific set of node actions (e.g., ways firms could interact with each other). More-

[1]This action is currently based on shortest distances in the network. If available, node information about location of firms can also be used.

over, the modeler may wish to ensure a specific network backbone, which can be easily accommodated by defining the initial topology before executing the Monte Carlo simulation. Termination conditions for the synthesis algorithm are user defined, e.g., certain number of edges created or topological characteristics have satisfactorily emerged.

Incorporating tier constraints

Supply chain networks are formed from heterogeneous types of nodes, where each node has a specialized task. Hence, extra care must be taken to appropriately capture critical real-world constraints concerning the rules firms use to exchange goods. Some are trivial, while others are context-dependent. Failure to reasonably accommodate these constraints in the generative process will severely limit its utility to providing only very general insights into SCN design principles. Previous research has suggested that SCNs should be modeled as tiered networks, where each tier contains nodes performing different functional tasks and the constraints of edge formation apply to the entire set of nodes in a particular tier [220, 230, 242, 243, 246]. As described in Section 5.1, each firm in a supply chain belongs to a unique *tier*, which corresponds to the number of hops from the consumer in the final supply chain. For a network $G = (V, E)$, the set of nodes can be partitioned into l tiers $V = T_0 \cup \cdots \cup T_{l-1}$, such that for $\alpha \neq \beta$, $T_\alpha \cap T_\beta = \emptyset, \forall \alpha, \beta \in \{0, \ldots, l-1\}$. Similarly, tiers also introduce constraints on the possible set of edges, such that a node $v_i \in T_\alpha$ ($\alpha \geq 1$) can only supply materials to a node $v_j \in T_{\alpha-1}$, i.e. supply-demand relations exist only between firms in adjacent tiers. Algorithm 4 shows how ABNG is used to synthesize tiered SCNs for a given action matrix $\mathbf{M}$ by restricting the actions to select nodes that satisfy tier constraints.

Algorithm 4 Synthesis algorithm $F(\mathbf{M}, V, m, \xi)$ for target network $G^* = (V, E)$

1: Create a network $G = (V', E')$ with $V' = V$ and $E' = \emptyset$
2: Using $\mathbb{P}(T = t)$, probabilistically assign a node-type t to each $v_i \in V'$
3: **while** $|E'| < |E|$ **do**
4: Select a tier $\alpha \in \{0, \ldots, l-1\}$, followed by a node $v_i \in T_\alpha$
5: Probabilistically select an action a_l for v_i using $\mathbb{P}(A = a_l|t)$
6: Add edge (v_i, v_j) to G as determined by a_l and satisfying $v_j \in T_{\alpha-1}$
7: **end while**
8: **return** G

5.2.3 Optimization and determining generator suitability

As seen in Equation (4.2), the problem of finding an action matrix $\mathbf{M}$ is framed as a multiobjective problem. The decision to frame this within a multiobjective context is based on numerous observations in network science literature arguing that it is a robust approach to determining generator suitability [61, 63, 153, 163]. To solve this multi-objective search problem, we implement Pareto Simulated Annealing (PSA) [209], as it is known to be a useful metaheuristic capable of global optimization in a large search space in a fixed amount of time.

Choice of objectives

An objective of the current research is to learn the action-based model for a given SCN, while preserving its resilience properties. Network resilience has emerged as a critical topic in supply chain research, and a summary of several metrics that may help understand supply network resilience can be seen in [244]. Recent research [228, 229] has uncovered the importance of network topology in determining resilience of SCNs under random and targeted disruptions, hence highlighting the importance of considering it as an essential component of SCN modeling. To preserve the structural and resilience properties of the real-world SCN, the action-based framework uses the

2-sample Kolmogorov-Smirnov statistic to quantify difference in distribution of node level properties between the synthesized and target networks. In the current experiments, node betweenness, in-degree and out-degree are utilized as the global network characteristics $Y(G^*)$, but the approach is indifferent to the choice of objectives, and other properties with alternative approaches like KL-divergence or entropy-related measures can be used to quantify the difference in distributions of network properties. The choice of objectives is based on the efficacy of these measures to capture the essential structural features of a network, especially in the the context of our action-based approach [114]. Further, [244] uses these network properties to understand supply network resilience in different network structures.

5.3 Results

The approach described in Section 5.2 is used to infer action-based models for each of the real-world SCNs listed in Table B.2. Additionally, the action-based framework was also used to synthesize supply networks that reflect the observations of empirical studies, such as power law degree distributions and disassortative mixing. The inferred probabilistic models are also used to draw conclusions about the individual local mechanisms that are primarily responsible for link formation in SCNs. The networks synthesized using the learned models were then subjected to random and targeted disruptions of nodes, and evaluated against two resilience metrics. This provides an indirect yet effective way of analyzing the ability of a network to remain functional under adverse circumstances.

5.3.1 Modeling SCNs

To test the ability of ABNG to replicate distinct global network properties observed in real-world SCNs, the generator was tested using 10 target networks from the dataset described in Section B.2. The list of SCNs that were considered for modeling is provided in Table B.2 along with eight relevant network level metrics of the

target networks and the corresponding networks synthesized using ABNG. Figure 5.1 presents a summary of the results, featuring heat maps for the 10 target networks. The solution closest to the origin (based on 1-norm) was chosen as the action-based model and was used to synthesize 20 networks each. The mean dissimilarity values are recorded in the heat maps by comparing the 2-sample Kolmogorov-Smirnov statistic value for betweenness, in-degree and out-degree between the target and synthesized networks (these were used as objectives during the optimization). Additionally, we also provide mean values of absolute deviation of the average path length, network centralization and network heterogeneity between the target and synthesized networks (see [240, 244] for details and definitions of these properties and their relevance in the context of supply chains).

As seen in Figure B.1, the target networks might impose strict constraints on how nodes in two tiers are connected, for example the green and blue nodes in the supply chain of Computer Peripheral Equipment have a one-to-one mapping. The constraints imposed by such specialized sub-structures might lead to synthesis of disconnected networks using ABNG. Synthesis of SCNs that are not fully connected is not a desirable outcome. To deal with this issue, a clean-up phase was devised to ensure every node participating in the supply chain is connected to the synthesized SCN. If the algorithmic procedure described in Algorithm 4 synthesizes a disconnected network, the clean-up phase is initiated to create a connected supply chain by randomly connecting a disconnected node to a node that is already a part of the overall supply chain, while adhering to tier constraints described in Section 5.2.2. The networks obtained after the clean-up phase are then subjected to further analysis.

An action matrix corresponding to the solution closest to the origin (based on 1-norm) obtained for three of the real-world SCNs is shown in Table 5.1. This can help the user in making some conclusions about the structure of these networks and the propensity with which each action is used to form links. ABNG was also used to synthesize an artificial SCN where the objective was to match the statistical properties observed in most empirical studies, i.e. in and out degree distributions with power

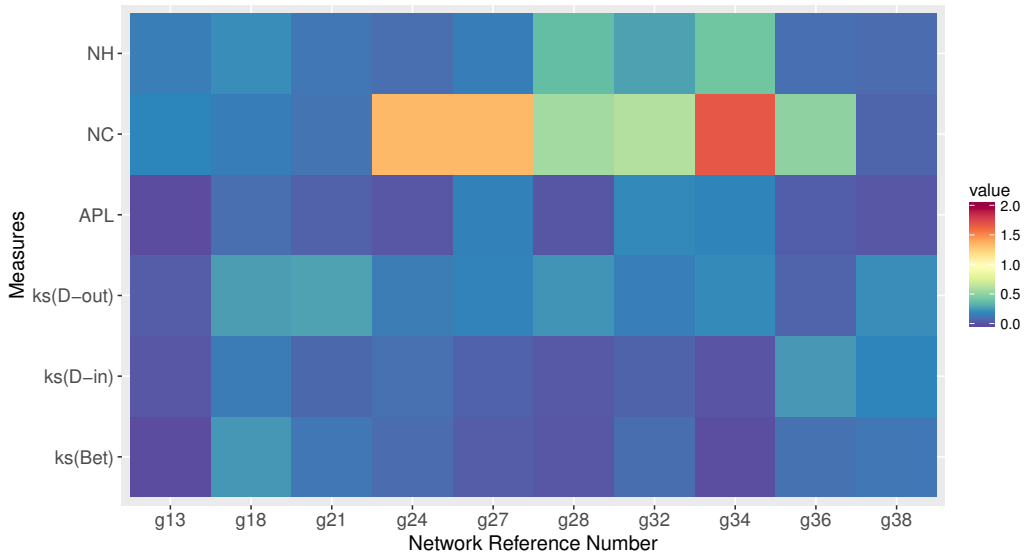


Figure 5.1. Results of measures for the 10 SCNs modeled using ABNG. The solution closest to the origin (based on 1-norm) was chosen as the action-based model.

law coefficients $\alpha = 2$ and disassortative mixing among nodes (see Figure 5.2 for a visual of the synthesized network). The action matrices obtained for each of these networks have some similarities, but with subtle differences. It can be seen that most of the SCNs have a corresponding action-based model consisting of nodes belonging to only two or three different node-types, i.e. most of the firms use similar local mechanisms to form links. A common observation is that "no action" tends to have high probability. A possible conclusion here is that only a few nodes add edges in a time step, leading to a power law degree distribution in the network. Further, most networks use preferential attachment on in-degree, degree difference and betweenness as dominant mechanisms for forming links. This provides evidence that firms that get more supplies tend to attract more connections, firms tend to link disassortatively and try to connect with nodes in shortest paths. The SCN on computer storage devices is an exception where nodes tend to link based on closeness and out-degree with higher

Table 5.1.
The table shows optimized action matrix for a few SCNs. The following actions were used: Preferential attachment on: out-degree (PAOD), in-degree (PAID), degree difference (PADD), betweenness (PAB), closeness (PAC); Random selection (Rand); Inverse shortest distance (InvSD); and No action (NA).

Network ↓ | Action→	PAOD	PAID	PADD	PAB	PAC	Rand	InvSD	NA	$\mathbb{P}(T = t)$
Perfumes, Cosmetics, and	0.000	0.174	0.015	0.108	0.037	0.076	0.158	0.432	0.188
Other Toilet Preparations	0.007	0.249	0.179	0.074	0.000	0.008	0.000	0.483	0.812
	0.077	0.032	0.009	0.373	0.026	0.000	0.000	0.483	0.508
Power-Driven Handtools	0.053	0.193	0.236	0.223	0.096	0.021	0.178	0.000	0.091
	0.018	0.000	0.000	0.094	0.023	0.010	0.000	0.855	0.401
	0.266	0.000	0.388	0.024	0.082	0.089	0.000	0.151	0.203
Computer Storage Devices	0.226	0.017	0.143	0.116	0.167	0.017	0.314	0.000	0.186
	0.205	0.000	0.208	0.169	0.229	0.123	0.066	0.000	0.611
Artificial SCN	0.020	0.001	0.000	0.183	0.000	0.000	0.000	0.796	0.851
	0.132	0.156	0.251	0.091	0.132	0.071	0.108	0.059	0.149

probability. It should be noted that the action based on distances does not get a high probability because the current version uses distance as number of hops in the network. If geographical location is available in the dataset, this action is likely to have much higher probability.

5.3.2 Resilience analysis

The supply chain literature emphasizes that specific measures are required for evaluating topological resilience of SCNs by incorporating the role of various nodes in the network. Analytical measures of resilience commonly used in the network science literature [247, 248] are unable to account for node heterogeneity, which is a critical aspect in SCN modeling. For example, [229, 249] point out that in the context of SCNs, the distance between two supply nodes or two demand nodes are not as important as that between a supply and a demand node. To tackle this issue, researchers rely on simulation based metrics to analyze topological resilience through

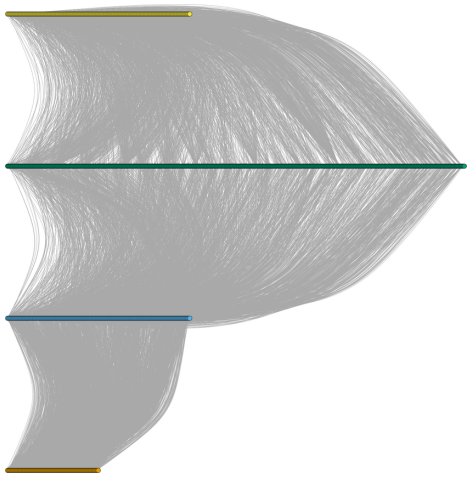


Figure 5.2. A visual representation of the tiered structure of artificial SCN synthesized using ABNG replicating basic SCN properties of power law degree distributions and disassortative mixing. The network consists of 1000 nodes (100, 200, 500, 200 nodes in tiers 0, 1 ,2, 3 respectively) and 7000 edges

customized metrics. The usual approach consists of simulating random or targeted disruptions by removing nodes from the network. [240] provides an outline of the methodological framework that is typically used for analysis of topological resilience of SCNs. This procedure consists of sequentially repeating the following steps:(i) simulate node removal, and (ii) measure the relevant resilience metrics. This can provide general insights into the topological aspects of SCN resilience.

To incorporate the heterogeneous roles of firms for resilience analysis in SCN, the set of nodes V can be divided into supply (V_S) and demand (V_D) nodes. We assume that the final consumers are the demand nodes ($V_D = T_0$), and every other node is a supply node ($V_S = V \backslash V_D$). For a network to be resilient, the most important requirement is to ensure that the demand nodes have access to at least one supply node. *Supply availability rate* measures the percentage of demand nodes that have access to supply nodes, hence providing an estimate of whether the demand nodes have access to supplies for maintaining normal operations. As shown in Equation 5.1,

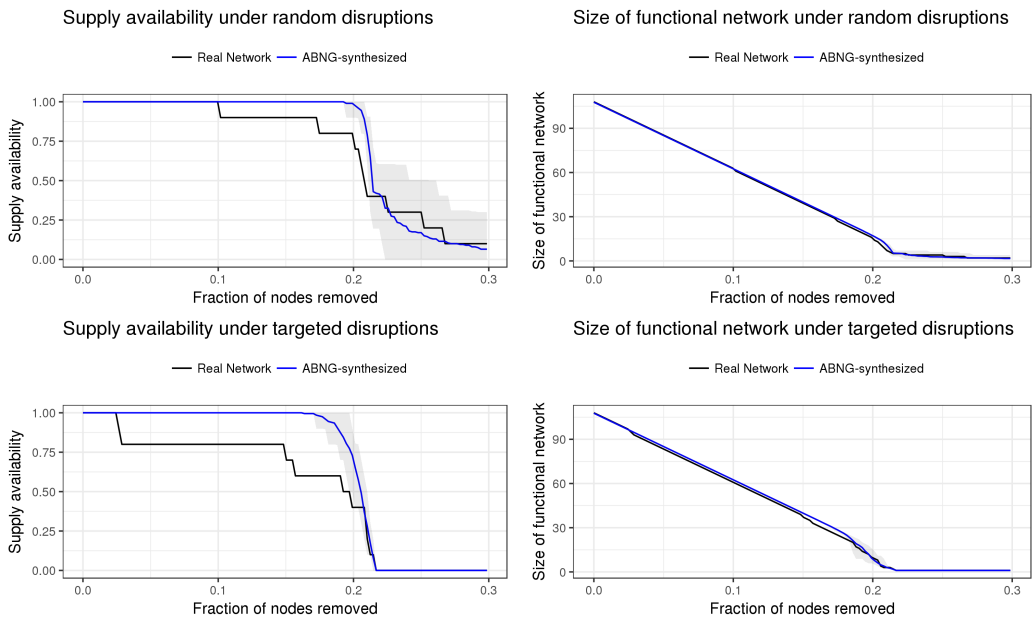


Figure 5.3. Resilience analysis: SCN of semiconductors and related devices

the supply availability rate S_A can be calculated as the ratio of number of consumers that still have access to a supplier to the total number of consumers.

$$S_A = \frac{|V'_D|}{|V_D|}, \text{ where } V'_D = \{v_i \in V_D | \exists v_j \in V_S : \exists \text{ path between } v_i \text{ and } v_j\} \tag{5.1}$$

The second measure, *size of the functional network*, corresponds to the number of nodes in the largest connected component that has at least one supply node, thus serving as a measure of supply network connectivity. For calculating the size of the largest functional network, we first need to find the largest connected component that satisfies the required conditions. Let V_{sub} be the set of nodes in the remaining functional network, then a node in V_{sub} should satisfy the following requirements:

$$(i)\ \forall v_i, v_j \in V_{sub} : \exists \text{ path between } v_i \text{ and } v_j, \text{ and } (ii)\ \exists v_k \in V_{sub} : v_k \in V_S \tag{5.2}$$

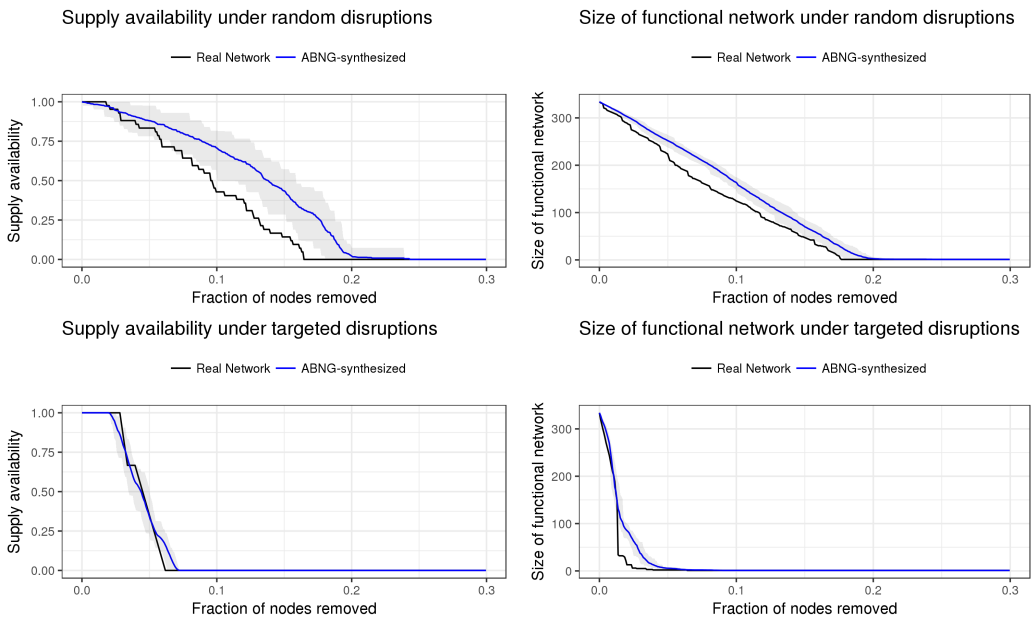


Figure 5.4. Resilience analysis: SCN of power-driven handtools

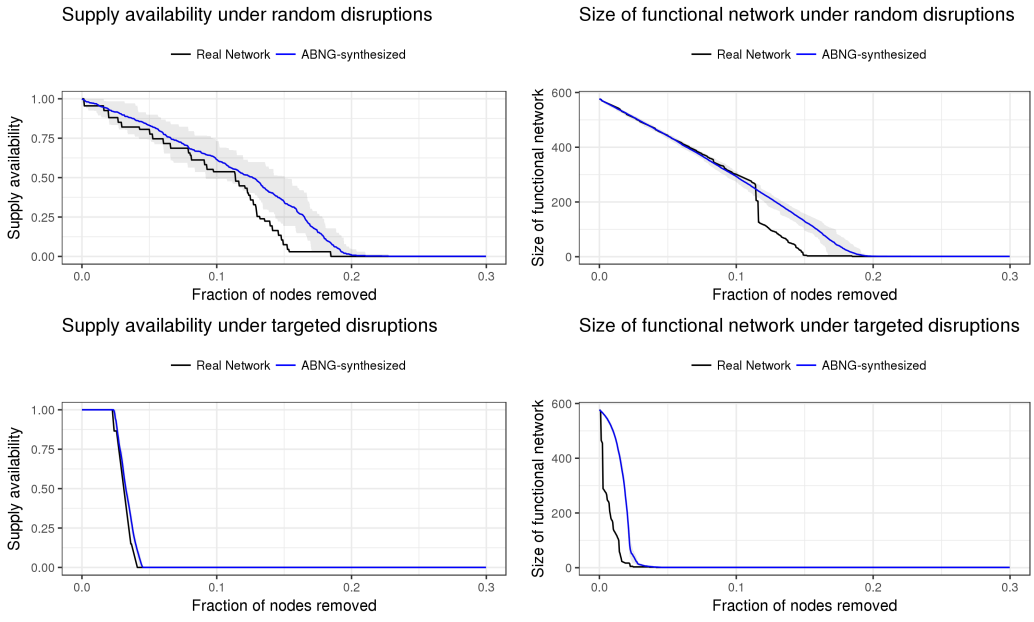


Figure 5.5. Resilience analysis: SCN of computer storage devices

Once a set of measures for evaluating topological resilience of a network have been decided, the sequential procedure described earlier can be used to perform analysis on a network. In this research, we simulate random disruptions by randomly removing a supply node from the network, and targeted disruptions remove a supply node with the highest total degree (sum of in and out degree). Though we perform degree-based targeted disruptions, variations that use different centrality measures can also be used.

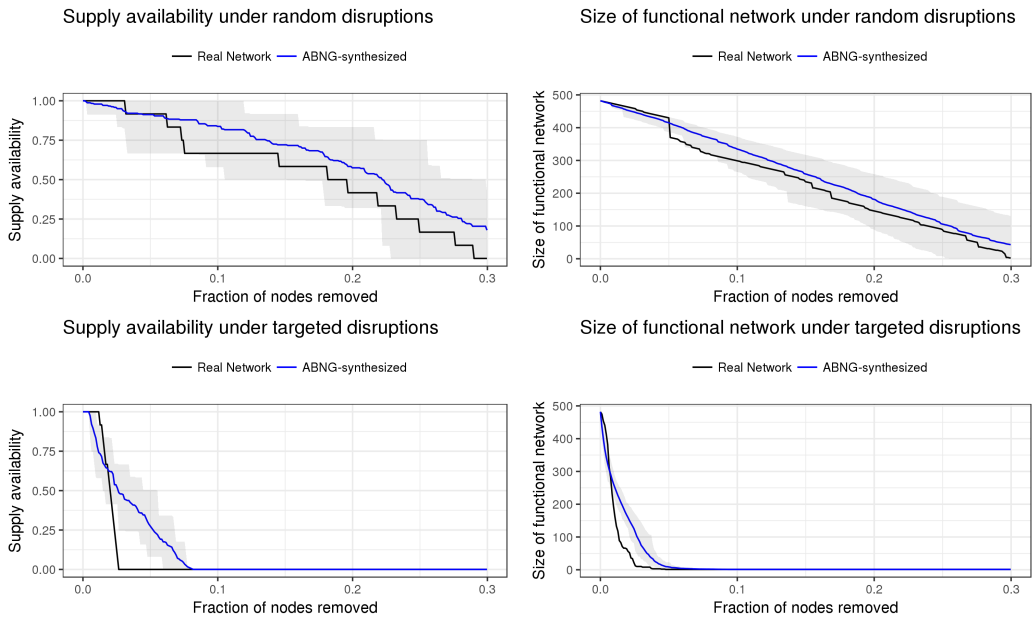


Figure 5.6. Resilience analysis: SCN of electromedical and electrotherapeutic apparatus

Figures 5.3-5.7 show the effect of random and targeted removal of nodes from the synthesized and real-world networks on supply availability and size of the functional network under disruptions. The ABNG-synthesized network corresponds to those created by the modeling procedure described in this chapter (followed by the clean-up phase, if required). Each blue line corresponds to the average resilience values of 20 synthesized networks, with the highlighted region capturing the networks lying between the 5th and 95th quantile. It is interesting to note that average resilience of

the networks synthesized using our approach are generally comparable to or better than the considered real-world networks under both disruption scenarios. [239] analyzed the SCNs that we use as target networks and concluded that the networks show high structural resilience. This implies that the actions used in this research can be used to make informed decisions leading to design of resilient supply networks as the synthesized networks have resilience comparable to the target networks.

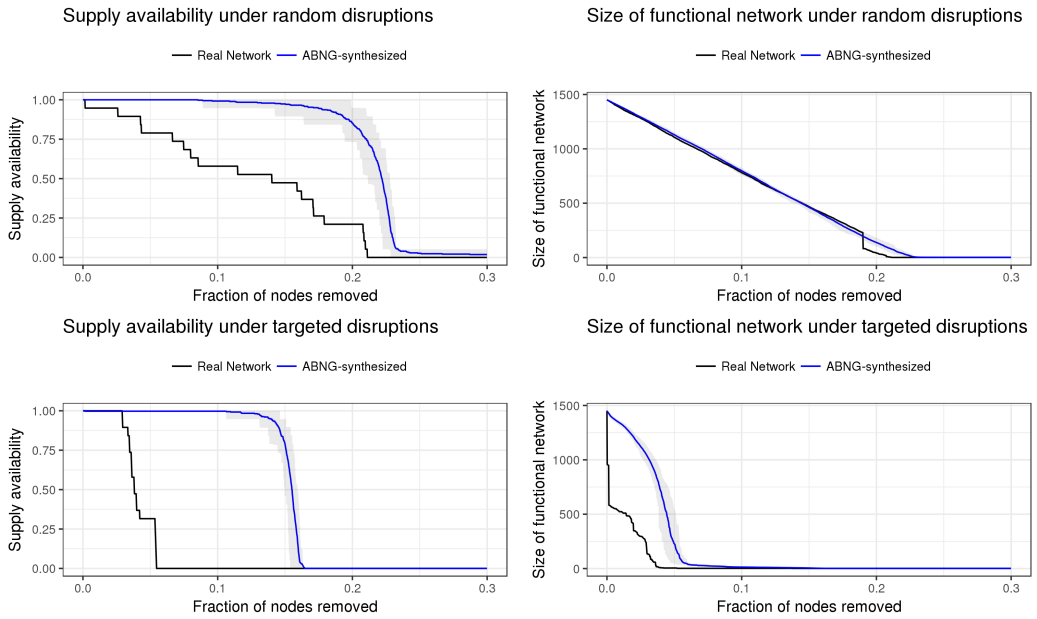


Figure 5.7. Resilience analysis: SCN of farm machinery and equipment

The artificial SCN shown in Figure 5.2 was also subjected to resilience analysis, and the results can be seen in Figure 5.8. It should be noted that because there were no constraints from a target network, the synthesized networks are fully connected. Another important observation is that there is very little variation (small highlighted portion in Figure 5.8) because of the absence of real-world constraints on the network structure.

Overall, the proposed model is capable of synthesizing networks that are structurally similar to real-world SCNs only by utilizing a few global network properties in form of objective functions $Y(G^*)$, and also incorporating context-dependent con-

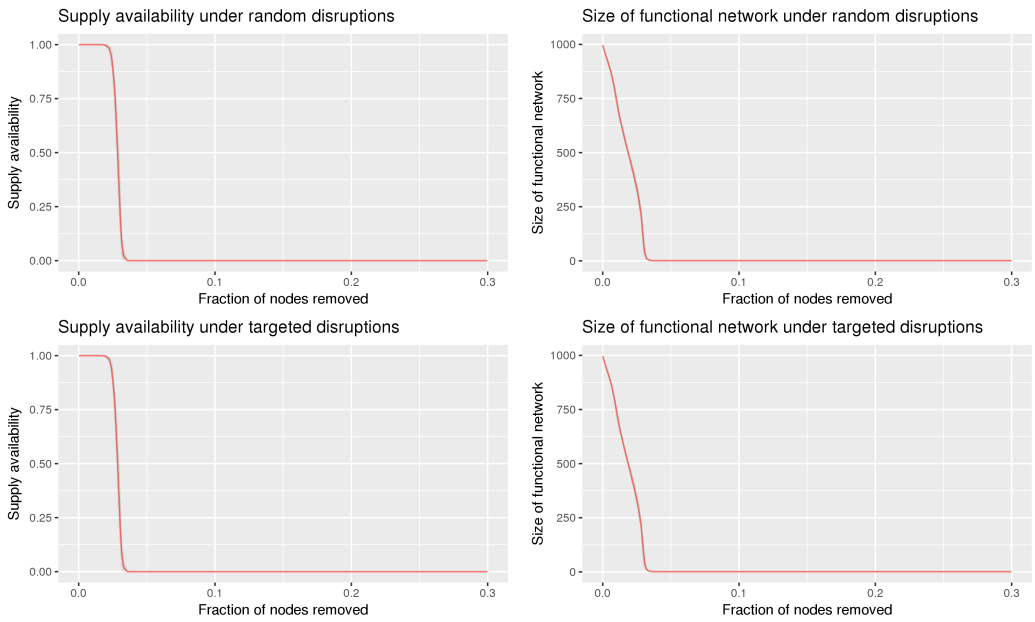


Figure 5.8. Artificial supply network

straints on link formation between firms in different tiers. The ability of the synthesized networks to retain functionality under disruption demonstrates that the micro level linking decisions (actions) of individual firms can be used in a probabilistic manner to synthesize topologically resilient network structures. Using the information from node level actions and corresponding probabilities can provide network designers with a better understanding of supply chain dynamics, and hence make informed decisions regarding designing systems that can retain functionality under disruptions.

5.4 Conclusions and future work

The objective of this research was to investigate the possibility of using a network model to synthesize resilient supply networks capable of structurally replicating a given real-world supply chain. The framework incorporates essential features of SCNs like node heterogeneity by using tier information and allowing different mech-

anisms to connect with firms. The results indicate that decisions by firms at the micro level can lead to creation of networks that exhibit topological resilience, hence providing insights into network design principles. The framework can be extended to capture dynamics of such networks by adding features such as arrival of new nodes and rewiring of existing edges. Information regarding node demands and incorporating tighter constraints on demand fulfillment can make the model more representative of real-world SCNs, but the unavailability of material flow data might hinder the progress in this direction. Availability of demand data will also lead to synthesis of connected networks by ensuring that individual demands are satisfied. Firm fitness (generally evaluated using profit or loss) is seen as an important driving mechanism for supply chain evolution [250, 251] and can be included as an additional action in future research, and also used as a metric for addition and removal of firms/nodes in the network. This can effectively capture the dynamics of supply chains and how the evolution of the network effects its resilience.

6. ACTION-BASED MODELS FOR STRUCTURAL BRAIN NETWORKS

Recent developments in network neuroscience have highlighted the importance of developing techniques for analysis and modeling of brain networks. A particularly powerful approach for studying complex neural systems is to formulate generative models that use wiring rules to synthesize networks resembling the topology of a given connectome. Successful models can highlight the principles by which a network is organized (identify structural features that arise from wiring rules versus those that emerge) and potentially uncover the mechanisms by which it grows and develops. Previous research [252,253] has shown that such models can validate the effectiveness of spatial embedding and other (non-spatial) wiring rules in shaping the network topology of the human connectome.

In this Chapter, we propose variants of the action-based model [114] that combine a variety of generative factors capable of explaining the topology of the human connectome. We cross-validate our models by evaluating their ability to explain between-subject variability. Our analysis provides evidence that geometric constraints are vital for connectivity between brain regions, and an action-based model relying on both topological and geometric properties can account for between-subject variability in structural network properties. Further, we test correlations between parameters of subject-optimized models and various measures of cognitive ability and find that higher cognitive ability is associated with an individual's tendency to form long-range or non-local connections.

6.1 Introduction

The network of connections between neural elements of the human brain, often referred to as the human connectome [254, 255], creates an intricate and complicated structural network [256, 257]. The human connectome is an anatomical network, where nodes consist of neural elements (neurons or brain regions), and edges correspond to physical connections (synapses or axonal projections) between different neural elements. The network map of the human connectome can be used to describe, explain or predict the behavior of the physical network it represents [258]. In the past decade, network neuroscience has highlighted the importance of developing a wide variety of techniques for analysis and modeling of brain networks [100, 259–263]. Topological analysis based on various network measures has provided evidence for the non-random topology of the connectome, and has aided our understanding of the organization of the human brain [247, 264].

Early application of networks in neuroscience mainly focused on gathering summary quantities trying to find common features describing the organization of most biological neural networks [265], see [247] for a review. These summary quantities have been used to detect functional integration (shorter path lengths and efficiency) and segregation (high transitivity and presence of clusters) in the brain. The importance of individual brain regions and pathways can be computed using centrality metrics, such as betweenness, closeness, etc. Eventually, these measures have been useful for topological analysis and characterizing structural patterns observed in the network representation of the brain.

An alternative approach for studying complex neural systems is to formulate generative models that use wiring rules to synthesize networks resembling the topology of a given connectome [265] (see Figure 6.1 for a pictorial description). Successful models can highlight the principles by which a network is organized (identify structural features that arise from wiring rules versus those that emerge) and potentially uncover the mechanisms by which it grows and develops [258]. For example, the

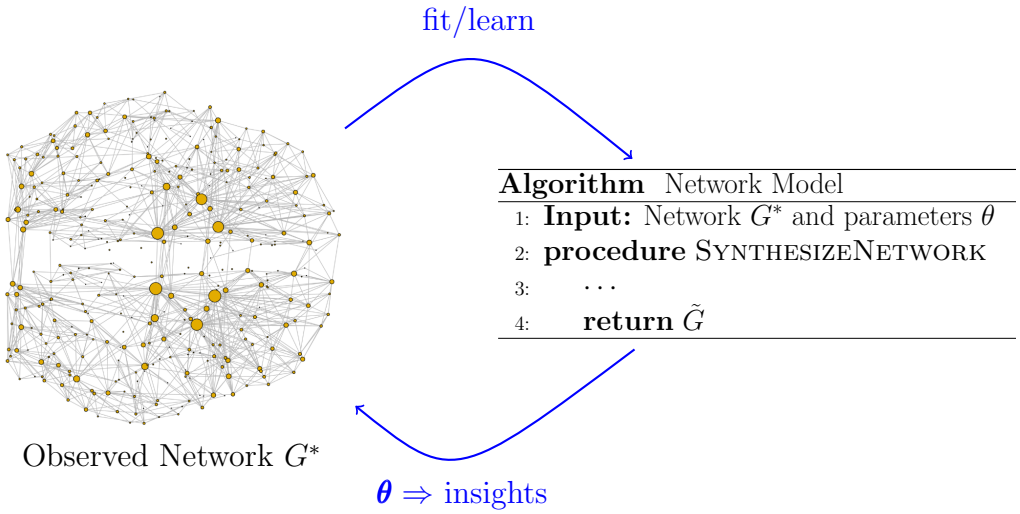


Figure 6.1. A generative network model uses a network representation of the as an input to learn parameters $\boldsymbol{\theta}$, which can then be used for draw insights about the topology of the network G^*.

spatial embedding of the brain, along with the economical wiring constraints that arise from this embedding play a vital role in crucial network characteristics, such as efficient network communication and information processing [264]. Previous research has shown that generative models can validate the effectiveness of spatial embedding and other (non-spatial) wiring rules in shaping the network topology of the human connectome [252, 253]. Such models can also provide insights into potential mechanisms that give rise to functionally important network attributes [160]. In addition to providing explanations for the wiring rules and processes of network formation, generative models for brain networks have the following advantages [265]:

1. They can compress our descriptions of the network representation of the brain and highlight potential regularities in their organization.

2. The ability to provide a compact description of the wiring rules enables these models to make out-of-sample predictions about unobserved network data.

3. Under appropriate assumptions, these generative models can uncover network mechanisms responsible for the structural organization of the brain.

While early work on generative modeling of the human brain (using resting state functional connectivity [252] or the connectome [253]) utilized at most one or two generative factors for predicting the topology, there has been growing interest in developing generative models that incorporate multiple rules for the probability of connections between regions of the brain [266, 267]. This is an assumption that is foundational to the action-based approach, where links are created using a combination of several pre-defined actions/rules whose probabilities are optimized for a given input network.

The choice of input network is particularly critical for generative modeling of the human connectome as the network should typify the complex structure of the entire set of brain networks [266–268]. In this Chapter, we use the structural brain networks of 100 unrelated subjects from the HCP dataset [269] to create a group representative median network G^*. This network is used to train four generative models: (i) null model based only on geometric distances, (ii) action-based model from Chapter 4, (iii) a variant of (ii) with an additional action based on geometric distances, and (iv) action-based model with visibility, where wiring rules use both topological and geometric properties to create edges. Each model is cross-validated by evaluating their ability to explain between-subject variability when trained on a single group representative network G^*. Finally, we also learn action-based models and its geometric counterpart, ABNG (vis), for each subject to study correlations between measures of cognitive ability and model parameters.

6.2 Methods

As in previous Chapters, for each of the generative models (briefly described in Sections 6.2.1-6.2.4), we formulate the problem of determining parameters $\boldsymbol{\theta}$ as a multi-objective optimization problem:

$$\begin{aligned} &\text{minimize} \quad \mathbb{E}\left[Q(G|G^*, Y, \boldsymbol{\theta})\right] \\ &\text{subject to} \quad \boldsymbol{\theta} \in D, \end{aligned} \tag{6.1}$$

where G is a network synthesized by a generative model with parameters $\boldsymbol{\theta}$ in the feasible domain D, and $Q(G|G^*, Y, \boldsymbol{\theta})$ is a measure to quantify the dissimilarity between a synthesized network G and the group representative network G^* based on a user-defined set of network characteristics Y. We minimize the expectation of Q to account for the stochasticity in the networks synthesized by a generative model.

Recent observations have highlighted the need to consider multiple global characteristics when comparing networks [61–63, 110, 114]. For our experiments, we use the first three terms of the *dk*-series [110] (i.e., Y = degrees + correlations + clustering/transitivity) as they have been shown to almost fully define local and global organization of most real-world networks. The 2-sample Kolmogorov-Smirnov D-statistic is used to quantify difference in distribution of these properties between G and G^*. As the resulting problem is multi-objective in nature, we obtain a set of Pareto efficient solutions after solving the optimization problem described in Equation 6.1. For each of the models, the solution closest to the origin, i.e. the one with lowest sum of objectives based on 1-norm, was chosen as the representative parameter setting.

6.2.1 Null model

The most basic model we consider in our experiments assumes that the probability of connection P_{ij} between nodes v_i and v_j is a function of the Euclidean distance d_{ij} between them

$$P_{ij} \propto \exp(-\eta d_{ij}). \tag{6.2}$$

This model assumes that the topology of the connectome can be attributed to minimization of the wiring cost, and the parameter $\eta \geq 0$ can be optimized (as

formulated in Equation 6.1) to determine the degree of cost penalization. NSGA-II [210] was used to solve the optimization problem for the null model, which resulted in $\eta \approx 0.73$ as the most representative solution. As illustrated in Figure 6.5, networks synthesized using the estimated parameter setting for the null model were unable to match the between-subject variability in the topological properties.

6.2.2 Action-based model

The second model we consider for our experiments is the action-based model introduced in Chapter 4. The action set used in our experiments consists of $K = 8$ actions, which are listed in Table 6.1 along with the representative action matrix. It should be noted that the estimated action matrix contains only three actions that have a probability greater than 0.05, implying that multiple mechanisms play a dominant role in the organization of the connectome. Further, the results in Figure 6.5 shows that while ABNG can characterize the between-subject variability in degree and assortativity, it fails to capture the clustering distributions.

6.2.3 Action-based model with distance

The importance of minimizing wiring cost necessitates an action that uses the spatial embedding of the connectome to create links between two nodes. To utilize this additional geometric information, an action is added to the model described in the Chapter 4, where a node v_i probabilistically selects v_j based on Euclidean distance. This new variant, ABNG (dist), uses the optimal cost penalization learnt in the null model ($\eta \approx 0.73$) to create an action based on geometric distance between nodes, as described in Equation 6.2. As in previous Chapters, the action matrix is optimized using PSA and the learnt model is used to synthesize networks. As seen in Figure 6.5, ABNG (dist) outperforms ABNG in capturing the distribution of properties with the help of the additional action that is chosen with probability 0.197 in the representative action matrix (see Table 6.1).

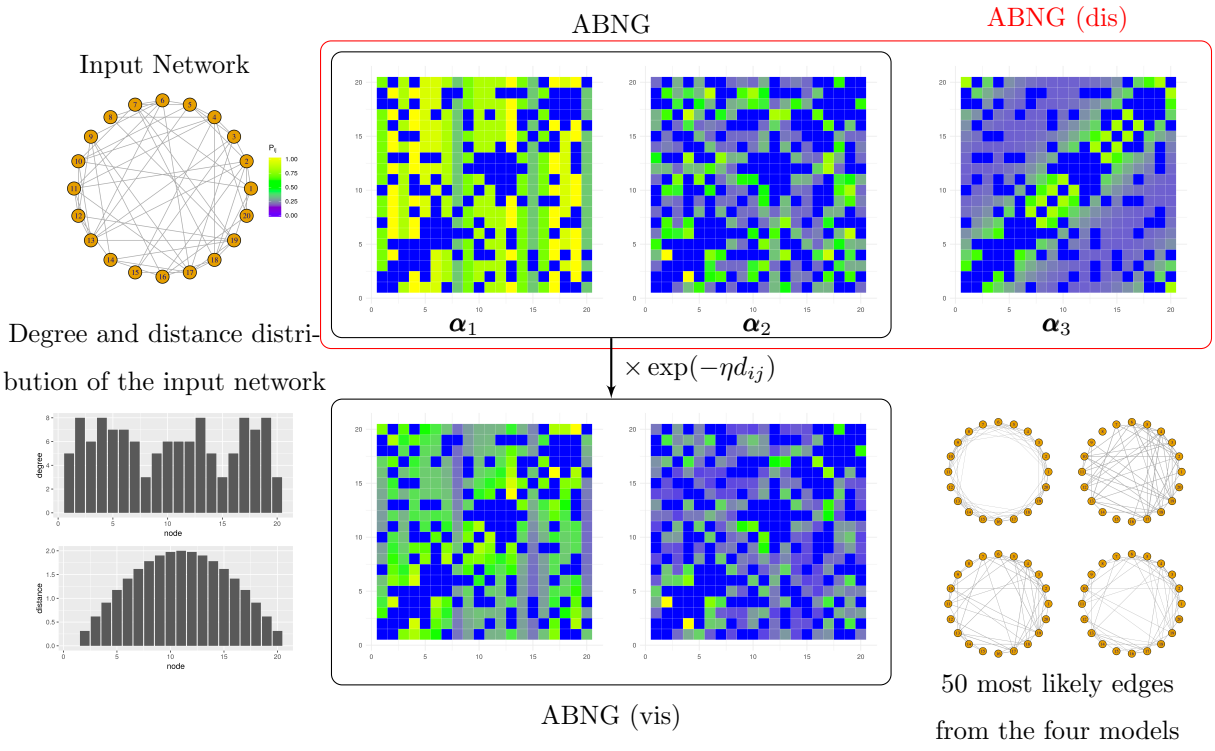


Figure 6.2. Pictorial explanation of how the different generative models work using a toy example. Given the input network, different models use or combine various rules to determine the probability of a new edge. In this toy example, ABNG uses two action: (i) preferential attachment based on degree, and (ii) inverse log-weighted similarity, which leads to output matrices $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ shown in the black rectangle labelled ABNG. ABNG (dist) also uses an action based on geometric distances shown in matrix $\boldsymbol{\alpha}_3$. The null model uses only $\boldsymbol{\alpha}_3$ to determine probability of a new edge, while ABNG (dist) combines the output of all three actions enclosed in the red rectangle. Finally, ABNG (vis) determines the probability of an edge by multiplying the output of actions in ABNG with a distance penalty term $\exp(-\eta d_{ij})$. To highlight the differences between the four models, we show output networks with 50 most likely edges. The following parameters were used: ABNG $\mathbf{M} = [0.4, 0.6]$, ABNG (dis) $\mathbf{M} = [0.3, 0.3, 0.4]$, ABNG (vis) $\eta = 0.5$, null model $\eta = 1$.

6.2.4 Action-based model with visibility

Previous research on generative models for the brain have highlighted the effectiveness of combining a distance based penalty with non-geometric rules to infer the probability of connection between different regions of the brain [252, 253]. This is a phenomena observed in many spatially embedded networks that have evolved to optimize similar functional requirements – high efficiency of information transfer between nodes at low connection cost – or to attain ideal balance between functional segregation and integration [100].

We propose the intuitive concept of *restricted node visibility* (or visibility in general), which assumes that when a node decides to create a link, the probability of creating a link is determined by a combination of actions and external factors intrinsic to the nodes. For example, in networks that exist in the Euclidean space (structural brain networks, transportation networks, etc.), a visibility function can be defined using node locations (distance in terms of node attributes can also be used as an input for defining visibility, but only after careful exploration of their relationship with network structure). In the context of ABNG, visibility can be seen as a way of skewing an action such that a node is more likely to connect with particular sets of nodes, consequently leading to the formation of communities. The idea has some similarity to network models that infer an embedding of nodes based on topology and use it to synthesize networks, see Section 2.2.4 for a review of these models.

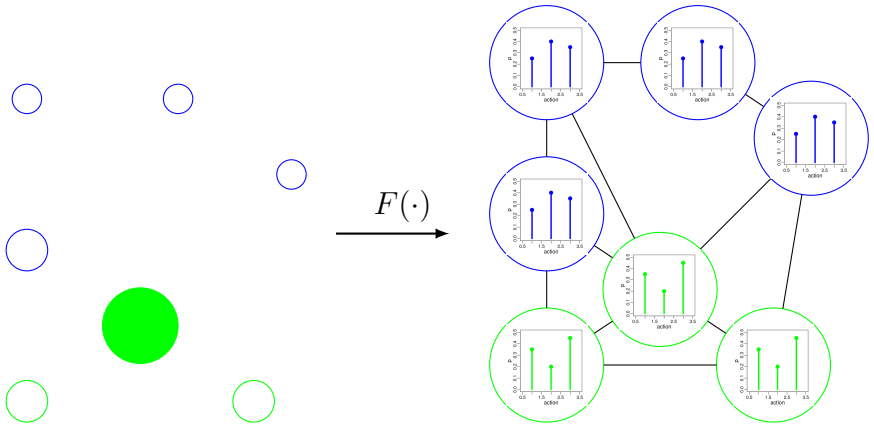
To combine the effect of multiple non-geometric actions with a distance-based penalty, we propose the action-based model with visibility, where the probability of an edge between two nodes is proportional to the output of an action (α_{ij}) scaled by the geometric distance between the nodes:

$$P_{ij} \propto \alpha_{ij} \times \exp(-\eta d_{ij}). \tag{6.3}$$

The overall idea is that the likelihood of an edge between a pair of nodes depends on a combination of their topological properties and geometric distance. We would

a_1	a_2	a_3	$\bar{P}$
0.25	0.4	0.35	0.6
0.35	0.2	0.45	0.4

(a) Example action matrix with two node-types, blue and green.



(b) Pictorial description of how the action-based model with visibility is used to synthesize networks using the action matrix shown above.

Figure 6.3. Pictorial description of the action-based model with visibility: The probability that the green colored node connects to other nodes is based on distance to other nodes and actions. The *visibility* of each node with respect to the green node is depicted by its size.

like to point out that by setting $\eta = 0$ in Equation 6.3 we can recover the original model proposed in Chapter 3, thus making the action-based model with visibility a generalized version of the action-based model. The process of learning such a model consists of two steps: (i) learn an action-based model, and (ii) estimate the visibility parameter η for the model learnt in step (i). In our experiments, we used the action matrix optimized for ABNG followed by optimization of the visibility parameter using NSGA-II [210] to learn ABNG (vis) parameters ($\eta = 0.11$) for the group representative

network G^*. The results in Figure 6.5 shows that ABNG (vis) is the best model among the ones considered here.

6.3 Experiments and results

To access the validity and effectiveness of the generative models proposed in Section 6.2, we designed a few different experiments, as outlined in Figure 6.4. The first step is to create a group representative median network G^* using the measurements of structural organization of the brains of the 100 unrelated subjects in the HCP dataset [269]. This is a crucial step as the models discussed in Section 6.2, similar to most models in the literature, are designed to learn parameters using a single input network. Thus, creating an input network that can capture the structural regularities of a cohort of subjects can facilitate the learning of better models. Previous research [266–268] has highlighted the importance of choosing a representative network for the parameterization of generative models for the brain.

Once the median network is constructed, it can be used as the input to learn parameters for each of the models. The parameterized models are then used to synthesize networks and their ability to capture the between-subject variability can be evaluated. While the group representative network G^* can capture the structural regularities of the cohort of subjects, it is expected that there will be subtle distinct features that are important for interpreting the difference between individuals [270]. The next step is to parameterize the models separately for each subject, and test if the fitted parameters can provide insights that can discern these individual differences. Structural brain networks are quantitative measurements of white matter micro-structure, whose integrity is crucial for healthy cognitive function [271]. Consequently, we decided to use our best model, ABNG (vis), for understanding the relationship between the structural organization of human brains and the cognitive ability of subjects.

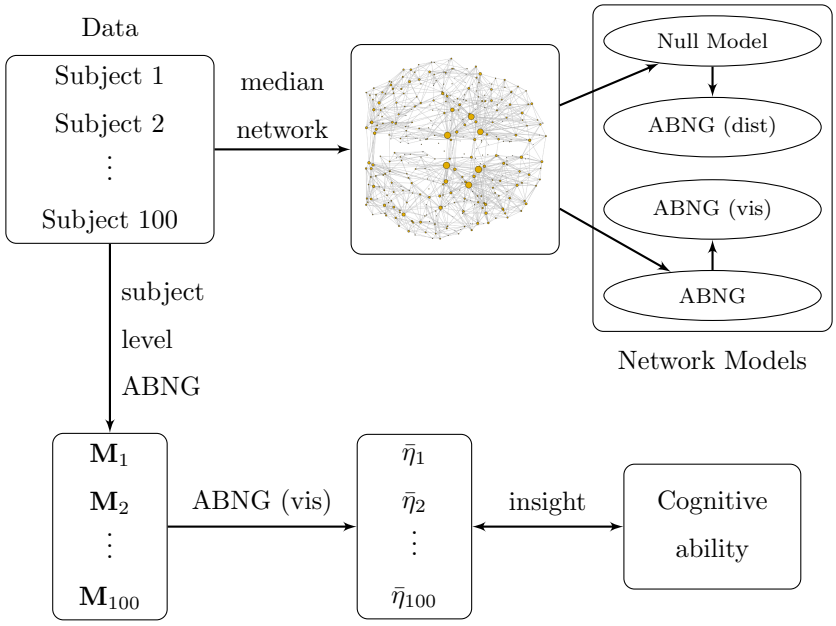


Figure 6.4. The structural brain networks of 100 unrelated subjects from the HCP dataset [269] are used to create a group representative median network G^*. This network is used to train four models described in Section 6.2. Each model is cross-validated by evaluating their ability to explain between-subject variability when trained on G^* (see Figure 6.5). We also learn action-based models $\mathbf{M}$ and its geometric counterpart ABNG (vis) for each subject to study correlations between measures of cognitive ability and mean model parameters $\bar{\eta}$ (see results in Figure 6.8).

6.3.1 Model cross-validation

Models fitted using Equation 6.1 should be able to synthesize networks that replicate some properties of the group representative network G^*. For a model to be useful, it needs to be cross-validated, for example, by using the best-fitting parameters from a model to synthesize networks that provides good estimates for the topological properties of a second network that was not involved in the model-fitting process. Such a procedure can help us ensure that the generative model is identifying wiring rules and not overfitting the observed data [265]. In our evaluation of the network models, we test their ability to reproduce the topological variability across subjects, and the

results are presented in Figure 6.5. The goodness-of-fit of the different variants of the action-based approach can be compared with the null model, thus showing that the proposed models performs better than a simple model based on geometric wiring rules. Figure 6.5 comprises of three different plots:

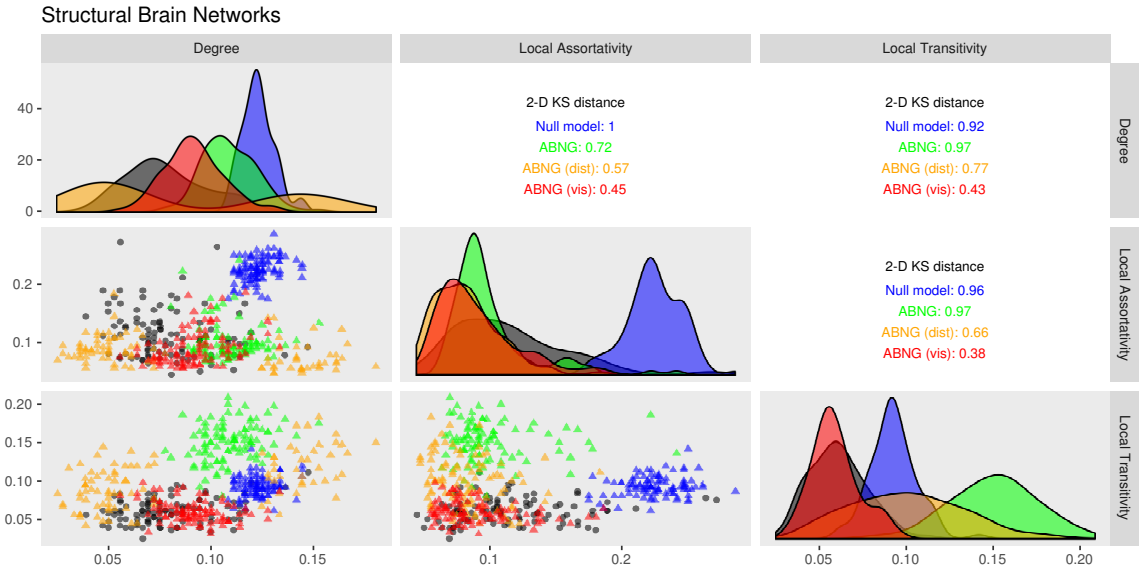


Figure 6.5. Empirical evaluation of the ability of the aforementioned network models to capture the between-subject variability using the group representative network G^* as the input.

1. Scatter plots below the diagonal show each synthesized/real network as a point in a network dissimilarity space, where the coordinates are computed using the Kolmogorov-Smirnov distance of the associated properties when the network is compared to the observed network G^* (the observed network itself is at the $(0, 0)$ position). Network models (colored triangles) showing higher overlap with the real brain networks (black dots) are better.

2. In the blocks above the diagonal, we quantify the extent to which a given generative model is able to reproduce the between-subject variability of topological

network properties of the 100 subjects using the 2-D KS distance [272] (lower the better).

3. Plots along the diagonal show the density distributions of the Kolmogorov-Smirnov distance of the associated properties when the network is compared to the observed network G^*. Similar density distribution to the real brain networks (black curves) implies good match in the properties.

Table 6.1.
The table shows optimized action matrices for structural brain networks *without* sub-cortical regions. The following actions were used: Preferential attachment on - average neighbor degree (PAND), degree (PAD), PageRank (PAPR) and betweenness (PAB); Triadic closure (TC); Inverse log-weighted (SLW) and Jaccard similarities (SJ); No action (NA); and Euclidean distance (ED). $\bar{P}$ corresponds to $\mathbb{P}(T = t)$, while η is the optimal distance penalty parameter for each of the models. The parameters are color coded to match with Figure 6.5: null model is blue, ABNG is green, ABNG (dist) is yellow, and ABNG (vis) is red.

				Triadic closure			No action		Distance Penalty	
PAND	PAD	PAPR	PAB	TC	SLW	SJ	NA	ED	$\bar{P}$	η
0	0.004	0.013	0.091	0	0.492	0.384	0.016	-	1	0.111
0	0	0	0.030	0.731	0	0	0.042	0.197	1	0.731
Preferential attachment					Similarity		Euclidean Distance			

The results in Figure 6.5 clearly show that all variants of the action-based approach outperform the null model. Further, the action-based model with visibility turns out be the best model, which is in agreement with past observations stating that the organization of the human brain arises from a combination of wiring rules based on wiring cost reduction and topological attachment mechanisms [252, 253]. In addition to learning accurate models for data, the parameters of our action-based approach

can be used to draw conclusions about potential mechanisms for network formation. The action matrices shown in Table 6.1 suggest that multiple mechanisms might be at play in the organization of the human brain. The parameters for the action-based model show that homophilic attachment (action based on similarity of neighborhood) mechanisms are the most important, but preferential attachment on betweenness is also crucial. Interestingly, the fitted distance penalty parameter for ABNG (vis) is smaller than the one obtained for the null model leading to a model that can better explain the individual variability.

6.3.2 Cognitive ability from structural connectivity

Our analysis so far has mainly been concerned with the ability of generative models to learn the between-subject variability of various topological properties while using only a single group representative network as the input. The assumption that structural brain networks of different subjects are expected to show similar connectivity patterns is pivotal to such an analysis [273]. But there are subtle differences in the connectivity patterns of different subjects [267, 268], which necessitates the parameterization of these generative models for the brain networks of each individual subject. Analyzing the model parameters for individuals can shed some light on the differences in the structural organization of brains of different subjects by comparing the estimated parameters of the generative models.

For this purpose, we use our best model, ABNG (vis), which involves first obtaining action matrices $\mathbf{M}_1, \ldots, \mathbf{M}_{100}$ for each subject followed by estimation of the respective visibility parameters (see Figure 6.4 for a pictorial representation of our procedure). In the optimization of the action matrix for each subject, we use the action matrix obtained for the group representative network as the starting solution (shown in green in Table 6.1), and perform a local search for each subject. After obtaining action matrices $\mathbf{M}_1, \ldots, \mathbf{M}_{100}$ for each subject, we perform multi-dimensional

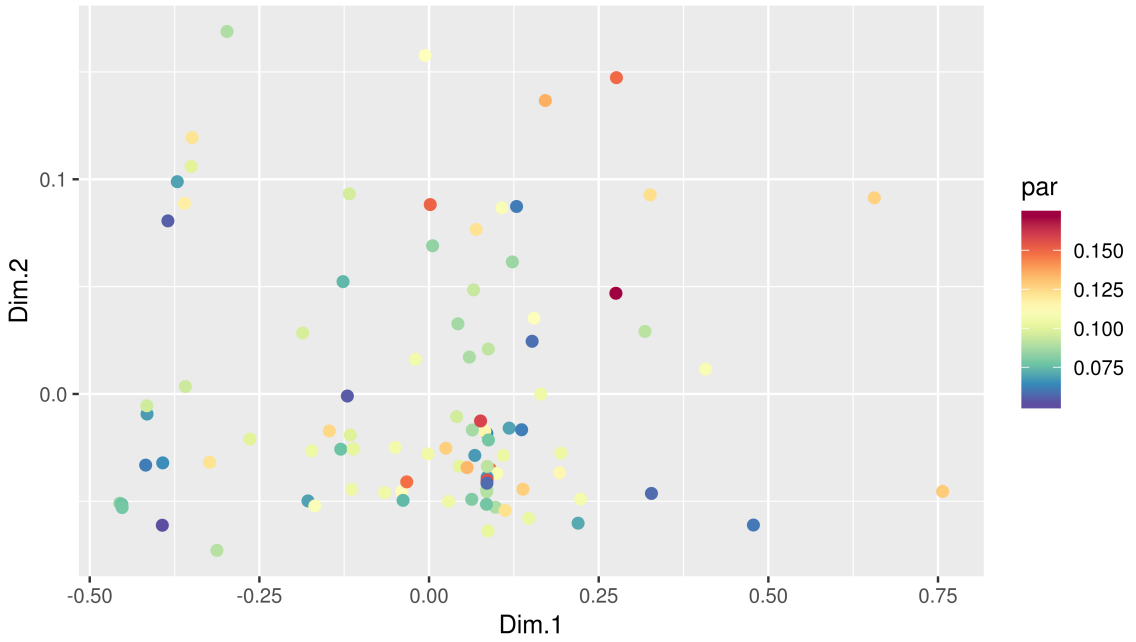


Figure 6.6. Multi-dimensional scaling of a single representative action matrix for 100 subjects, color shows mean visibility parameter $\bar{\eta}$).

scaling on the most representative action matrix for each subject to highlight the variation in the subject-level models, and the results are shown in Figure 6.6.

Although little is known about the organization principles that lead to individual differences between the connectomes, it is widely believed that these differences are associated with cognitive functioning [258, 274, 275]. Consequently, a recent topic of interest in neuroscience has been to uncover how individual differences in the network architecture leads to differences in general intelligence [275]. General intelligence is typically associated with the ability of an individual to perform a wide variety of cognitively challenging tasks well, which can be empirically computed as the first component of the principal component analysis of multiple measures of cognition [276]. For the HCP dataset [269], general intelligence can be computed using the following six measures of cognitive ability: (i) fluid intelligence (PMAT24_A_CR), (ii) episodic memory (PicSeq_Unadj), (iii) cognitive flexibility (CardSort_Unadj), (iv) lan-

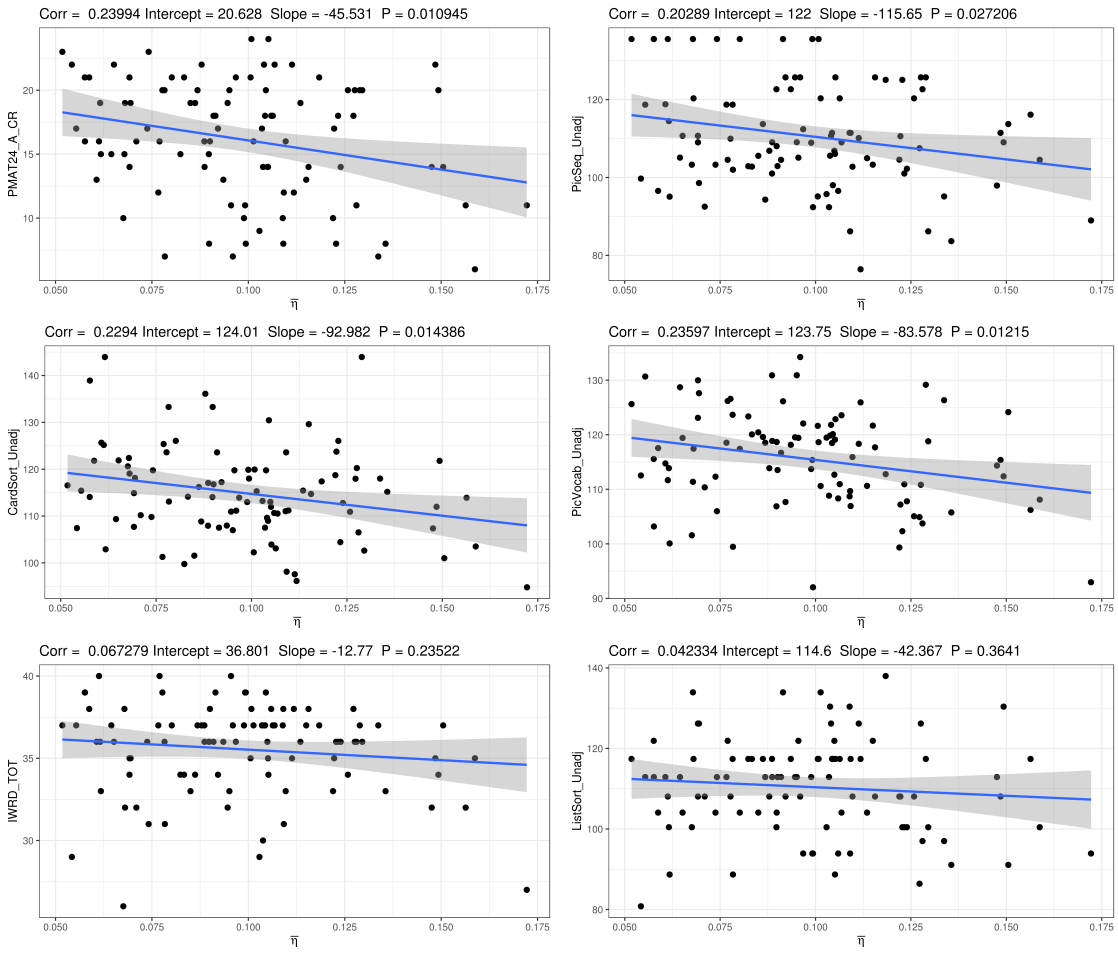


Figure 6.7. Correlation of $\bar{\eta}$ with various measures of cognitive ability

guage and vocabulary comprehension (PicVocab_Unadj), (v) verbal episodic memory (IWRD_TOT), and (vi) working memory (ListSort_Unadj). It seems obvious that the patterns in the structural connectivity are somehow related to an individuals' general intelligence. In fact, there has been research supporting that the efficiency of network topology is positively associated with cognitive ability [277, 278]. The visibility parameter in ABNG (vis) can serve as a proxy measure for the extent of functional integration and segregation in the structure of an individuals' brain, which plays a pivotal role in the ability of an individual to perform a variety of functional tasks [279].

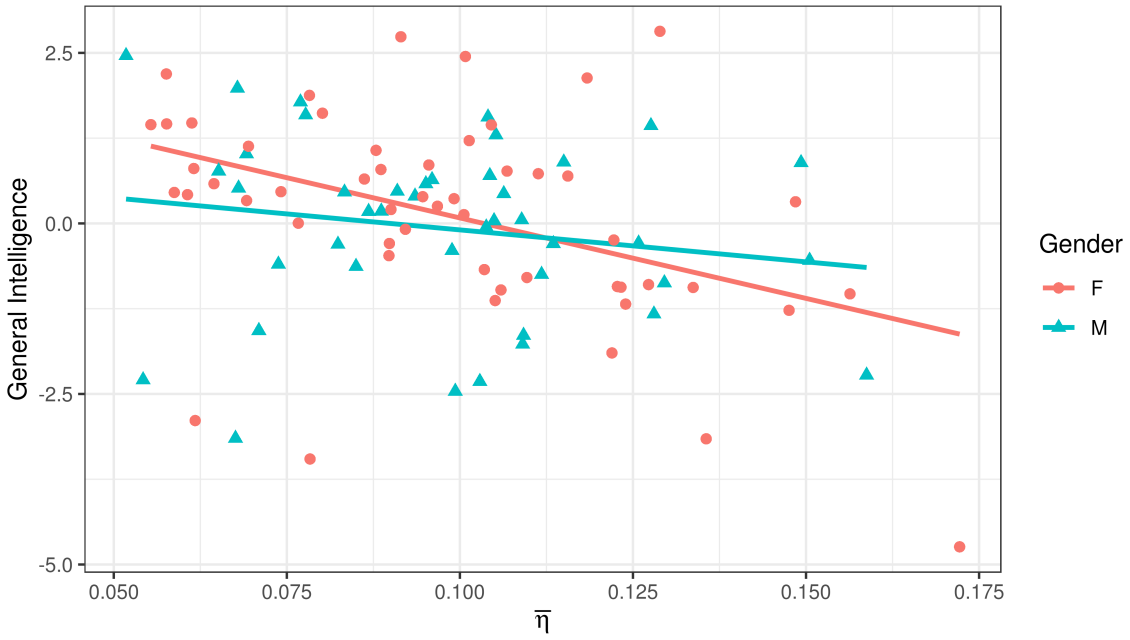


Figure 6.8. The plot shows results for correlation between general intelligence and $\bar{\eta}$ ($r = -0.318$, $r_M = -0.178$, $p_M = 0.259$ and $r_F = -0.429$, $p_F = 0.001$), where F and M are gender of the subjects.

Figures 6.7 and 6.8 plot the mean visibility parameter $\bar{\eta}$ (averaged across the Pareto front) for an individual against measures of cognitive ability for subjects. In Figure 6.7, the correlations and p-values are reported for the six different measures of cognitive ability listed above. Using a significance level of $\alpha = 0.05$, we can conclude that the correlation is not insignificant for four of the cognitive ability measures. Similarly, Figure 6.8 plots the general intelligence as the measure of cognitive ability and also distinguishes subjects based on their gender. Overall, the plots show that individuals with lower value of $\bar{\eta}$, i.e. showing a tendency to form long-range or non-local connections, obtain higher scores in the different tests for evaluating cognitive ability. In Figure 6.8, we also observe differences in the correlation between the two quantities when the gender of individuals is taken into account.

6.4 Conclusions

In this Chapter, we explored the ability of the action-based model (and its variants) to capture the between-subject variability in topological properties of structural brain networks while using a single group representative network as the input. Though the action-based model performed better than other generative models proposed in the literature, it failed to capture the local transitivity of the connectome. To tackle this issue, we proposed to use the spatial embedding of the brain and introduced geometric distances between various nodes as an additional factor responsible for the topology of the connectome. This enabled us to combine multiple topological properties (in the form of different actions) and their interaction with geometric distances to create better models for the human brain, something that prior models were unable to accomplish [252, 253]. Our results show that actions-based models with geometric constraints using wiring rules based on homophilic attachment and preferential attachment on betweenness can synthesize networks resembling human connectomes.

While other models such as exponential random graphs [268,280] and the weighted stochastic block model [281] have also been used for generative modeling of the connectome, they lack interpretability and are incapable of recovering mechanisms and rules that lead to the formation of an observed network. The action-based model with visibility outputs an action-matrix that shows the relative importance of various actions/mechanisms for a particular input network, and the visibility parameter highlights the role wiring cost plays in the organization of the connectome, thus providing a compact representation of the connectome.

The ability of our proposed models to synthesize networks that account for the topological properties and between-subject variability in these properties raises the possibility that the models can provide insights into the factors that have shaped the emergence of specific architectural or performance characteristics [263]. To test this hypothesis, we use our best model, ABNG (vis), to study differences in estimated parameters for different individuals and discover that the value of distance penalization

is significantly correlated with cognitive ability in the form of general intelligence. We find that the differences in structural connectivity have some association with the the cognitive ability, specifically with the extent of functional integration and segregation. An interesting outcome of our experimental analysis is that the correlation is more pronounced in the female subjects of our dataset, which must be subjected to further examination using other neuroscience techniques.

7. QUANTIFYING THE VARIABILITY IN NETWORK POPULATIONS AND ITS ROLE IN GENERATIVE MODELS

In an ideal scenario, a generative model should be able to synthesize networks that are likely to evolve from the 'true' process that created the observation, but most models are not designed to accomplish this task. Due to the scarcity of data in the form of multiple networks that have evolved from the same process, generative models are typically formulated to learn parameters from a single network observation, hence ignoring the natural variability of the 'true' process.

In this chapter, we highlight the importance of variability in evaluating generative models and present a way of quantifying the variability for a finite set of networks. The evaluation scheme compares the statistical properties of networks in a dissimilarity space. Using the dissimilarity space, we evaluate the ability of four generative models to synthesize networks that capture the variability of the 'true' process. Our empirical analysis quantifying the ability of network models to replicate characteristics of a population of networks suggests that models aimed at exploring the generative mechanisms by fitting using a single network fail to capture the variability in the network population. Our work highlights the need for rethinking the way we evaluate the goodness-of-fit of new and existing network models and devising models that are capable of matching the variability of network population when available.

7.1 Introduction

As depicted in Figure 7.1, statistical modeling of networks based on a single observation assumes that G^* is somehow representative of the 'true' process A^*. Learning

a model using a single observation that does not reflect the variability in the process A^* can potentially bias a model to synthesize networks that over-fit G^*. Such a setup could restrict the generalizability of the model by failing to synthesize networks that provide a reliable representation of the 'true' process. To understand this, we need to define the concept of a network population.

Definition 7.1 (Network Population) *Let $G_1(V_1, E_1)$ be an arbitrary network that has non-zero probability of being synthesized using the process A^*. A finite set of k such realizations $\mathcal{G}_{A^*} = \{G_i(V_i, E_i)\}\ \forall i = 1, \ldots, k$ is called a network population, where V_i and E_i are the sets of nodes and edges in G_i.*

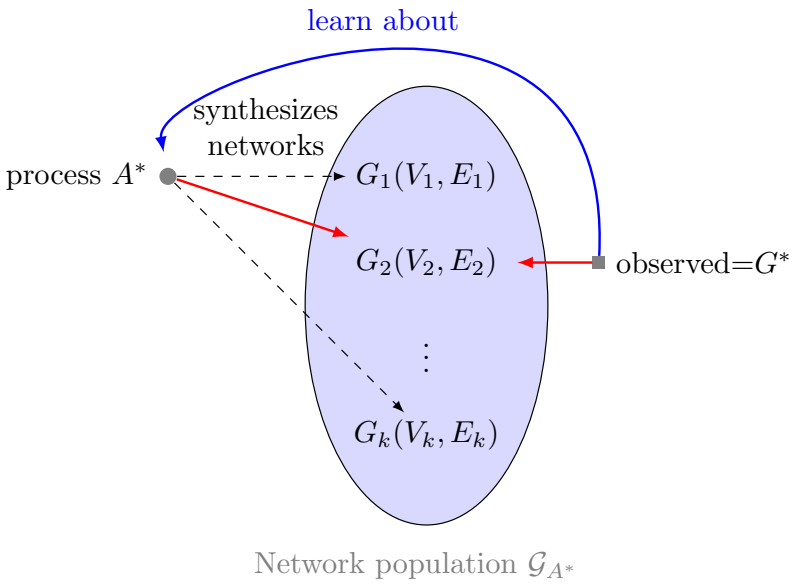


Figure 7.1. Procedure used for evaluating network models: Assuming G^* is not an outlier, how well do existing network models approximate the process A*?

To illustrate the variability in networks synthesized using a single known process A^*, we use the preferential attachment process of Barabási-Albert [12] to synthesize a population of 100 networks (labelled as "data" in Figure 7.2). The stochasticity in the generative process of the BA-model leads to the synthesis of networks that show some

variability in their global network properties (degree assortativity and transitivity are used in Figure 7.2). As depicted in Figure 7.1, we typically observe a single network from the population and ideally would like a generative model to be capable of learning about the original process A^* using the observation G^*. Consequently, one network from the population was selected at random as input for training two other network models, namely *dk*-random graphs as model 1 and action-based networks as model 2 (see Section 2.2 for more details), which were then used to synthesize populations of 100 networks each. Finally, we compare the distribution of degree assortativity and transitivity in the three populations in Figure 7.2. We observe that the networks synthesized by model 1 all have exactly the same global network properties. While there is no variability in these properties, a simple comparison with the observed network G^* might lead to a conclusion that the model aptly describes the underlying process A^*, which can prove to be highly misleading. On the other hand, the population of networks synthesized by model 2 shows more variability, but fails to match the network properties of G^* or the original population.

The inability of network models to synthesize realistic network populations necessitates the evaluation of network models using a well-formulated methodology that treats the observed network as a sample originating from some unknown process A^*. Statistical hypothesis testing for goodness-of-fit typically involves measuring the discrepancy between observed values and the values expected under the model in question. Similarly, in the context of networks, we would like to *evaluate the ability of a candidate model to approximate the network population $\mathcal{G}_{A^*}$ using a single observed network G^** (see Figure 7.1 for a pictorial representation). Although in most cases we do not have a population of independent instances of networks that can be used to draw a set of samples [282], it has been shown that it is possible to establish a baseline test set to evaluate the ability of a network model to capture the distribution of network populations [161]. [283] made a similar observation, where it was shown that evaluating the complexity of a network model using a single network can be bi-

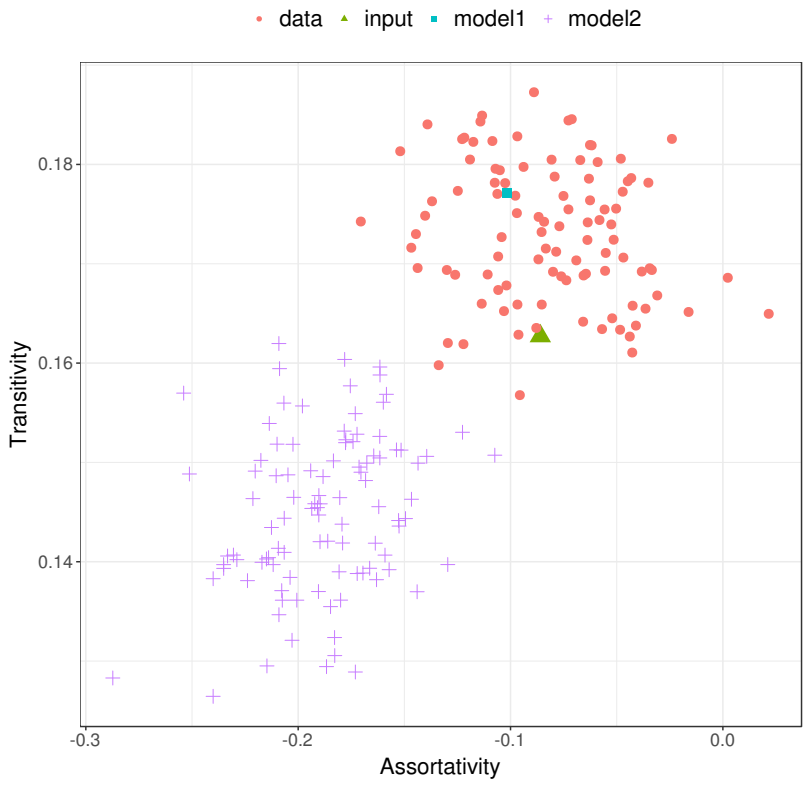


Figure 7.2. Distribution of global network properties assortativity and transitivity in populations of networks synthesized by the Barabási-Albert model and two other models. The inability of network models to replicate the distributional properties of the original process A^* highlights the need for devising better techniques for training and evaluating models using network populations.

ased because of the variability in the samples, and therefore introduced a complexity measure based on network populations.

Even though network science has provided us with numerous methods to compare pairs of networks (see [158,159] for reviews), comparing network populations has been a relatively unexplored area. Given the importance of natural variability in generative processes [161], it is imperative to devise methods that can be used to compare populations and quantify their variability. Such methods can help researchers in the

development of models that augment our current understanding of complex networks and the underlying process they represent.

One way to quantify the information content in a finite population of networks is to use entropy-based measures to compute the amount of randomness in the population. [284] review the diverse contexts and applications that led to the development of various entropy-based measures for graphs/networks. Information-theoretic metrics have also been used for quantifying the difference between pairs of complex networks [166]. Methods for quantifying the Shannon entropy of canonical and microcanonical network ensembles have also been proposed [285, 286], which were subsequently extended to the case of stochastic blockmodels [287]. While closed form expressions can be obtained for simple null models, achieving the same for a population of real-world networks is rather unlikely. Alternatively, data-driven techniques to quantify the information content and variability in network populations can aid the rapid development of more explanatory network models that capture the distributional properties of a population.

While the inability of certain network models to reproduce the naturally occurring variability in networks can be attributed to the fact that they sample each edge independently through Bernoulli distributions [120], there could be other factors contributing to the lack of variability. There are at least two ways of overcoming this obstacle: (i) systematic development of models that can replicate the distributions of populations, or (ii) devising techniques that can allow existing models to utilize additional information from the population.

There has been recent work on mapping networks into natural Euclidean spaces for conventional hypothesis testing [175], which can help us determine if two groups of networks are significantly different in statistics. Latent space models (LSM) [18] were proposed to synthesize networks by mapping the nodes into some low-dimensional Euclidean space while keeping the relationships. [288] adapted LSMs for multiple input networks and proposed a joint LSM considering multiple networks in the posterior. In [289], the context was extended to multiple observations drawn from a com-

mon population, and a non-parametric model was proposed. [290] proposed another Bayesian model where the probability of an observation is based on the Hamming distance to the Fréchet mean of the group. Similarly, [291] proposed a hierarchical modeling framework to learn better models of network populations. Although these approaches incorporate information about multiple networks to learn better representations for the population, they fail to explicitly account for the natural variability in the population.

To recap, an ideal generative model M would exactly correspond to the true process A^* that defines the dynamical processes responsible for the observed data G^*. That is, if A^* defines a probability distribution $\mathbb{P}_G(A^*)$ $\forall G \in \mathcal{G}_{A^*}$, then $\mathbb{P}_G(M)$ and $\mathbb{P}_G(A^*)$ would be identical. As stated above, A^* is usually unknown and the number of observed networks in the data G^* are usually small (sometimes only one).

In this chapter, we expand on previous work [161] about the distributional properties of four competing generative models: Chung-Lu model, *dk*-random graphs, exponential random graphs, and action-based network generators (Section 7.2.2 briefly describes how the model are best-fit to the data). We consider networks drawn from three known processes and six real-world populations. As described in Figure 7.1, model parameters are fit using a representative sample G^* chosen at random. In Section 7.2, we propose the construction of a dissimilarity space that measures the distributional properties of network populations with respect to the observed network G^*. In Section 7.3 the learned models are used to synthesize networks followed by an investigation of their distributional properties. This evaluation is done by comparing the statistical properties of the synthesized networks with the properties of the corresponding population of networks[1].

[1]Code available at `https://github.com/dlguo/network-variability`.

7.2 Experimental setup

7.2.1 Dissimilarity space

Evaluation of the distributional properties of a generative model requires a well-defined methodology that correctly represents the distribution over networks. Although model-based techniques for hypothesis testing of networks has been proposed in the literature [174, 292, 293], they heavily rely on the choice of a baseline model. Alternatively, one could build on the concept of a network morphospace [160], which provides a coarse-grained approach for classifying and mapping network architectures according to a set of network-level structural characteristics. The network morphospace can be transformed to a network dissimilarity space ($\mathfrak{D}_{G^*} \subset \mathbb{R}^d$), where networks are placed based on their dissimilarity to the single observed network $G^* \in \mathcal{G}_{A^*}$ with respect to a variety of dissimilarity measures (see Definition 2.1). The true process and network models also have counterpart distributions $\mathbb{P}_{\mathfrak{D}_{G^*}}(A^*)$ and $\mathbb{P}_{\mathfrak{D}_{G^*}}(M)$ in the network dissimilarity space. In an appropriately defined dissimilarity space, if $\mathbb{P}_{\mathfrak{D}_{G^*}}(M)$ sufficiently approximates $\mathbb{P}_{\mathfrak{D}_{G^*}}(A^*)$, we might be able to conclude that model M can synthesize networks that belong to the same population as the observed network G^*.

The utility of such a network dissimilarity space relies heavily on the choice of dissimilarity measures used for network comparison. Network science provides numerous quantitative tools to measure and classify different patterns of local and global network architectures across disparate types of systems. The development of methods for the pairwise comparison of networks is an active area of research and in recent years many new methods have been introduced (see [157–159] for reviews).

Any dissimilarity measure that defines a real-valued distance akin to the one in Definition 2.1, which goes to zero for a pair of isomorphic networks, can be used in the dissimilarity space. A set of node-level measures that could prove particularly useful for the network dissimilarity space is provided by the *dk*-series [110], which is a systematic series of properties $(Y_0, Y_1, \dots)$ of network structure defined in a way such

that each Y_i provides more detailed information about the network structure and Y_n fully characterizes a network with n nodes. [110] have shown that the first three terms in the dk-series (Y = degrees + correlations + clustering/transitivity) are capable of almost fully defining local and global organization of most real-world networks that do not exhibit community structure.

As discussed previously, most generative models are aimed at inferring the generative process using a single network observation. In our experiments with the dissimilarity space proposed above, we assume a single network randomly drawn from the network population serves as the input network G^* for the generative models. The rest of the networks in the population are treated as unobserved samples, and are used to evaluate the performance of models on matching the variability in the network population. Although Section 7.3 only shows the result for one randomly drawn G^*, the analysis and conclusion are consistent for different G^* samples (See Section 7.4 for further details).

7.2.2 Model fitting

For the Chung-Lu model, we directly used the degree distribution of G^* to compute the probability of a link between two nodes. For ERGMs, the following feature counts $\boldsymbol{\phi}(G^*)$ were used as they are known to be capable of circumventing the degeneracy problem (see [294, 295] for more details): (i) total number of edges, (ii) geometrically weighted degree distribution, (iii) geometrically weighted dyadwise shared partner distribution, and (iv) geometrically weighted edgewise shared partner distribution. We used $dk2.5$ for sampling from the population of dk-random graphs. For ABNG, degree distribution, local assortativity [296, 297], and local transitivity of the observed network were used as the set of network properties in the objective function.

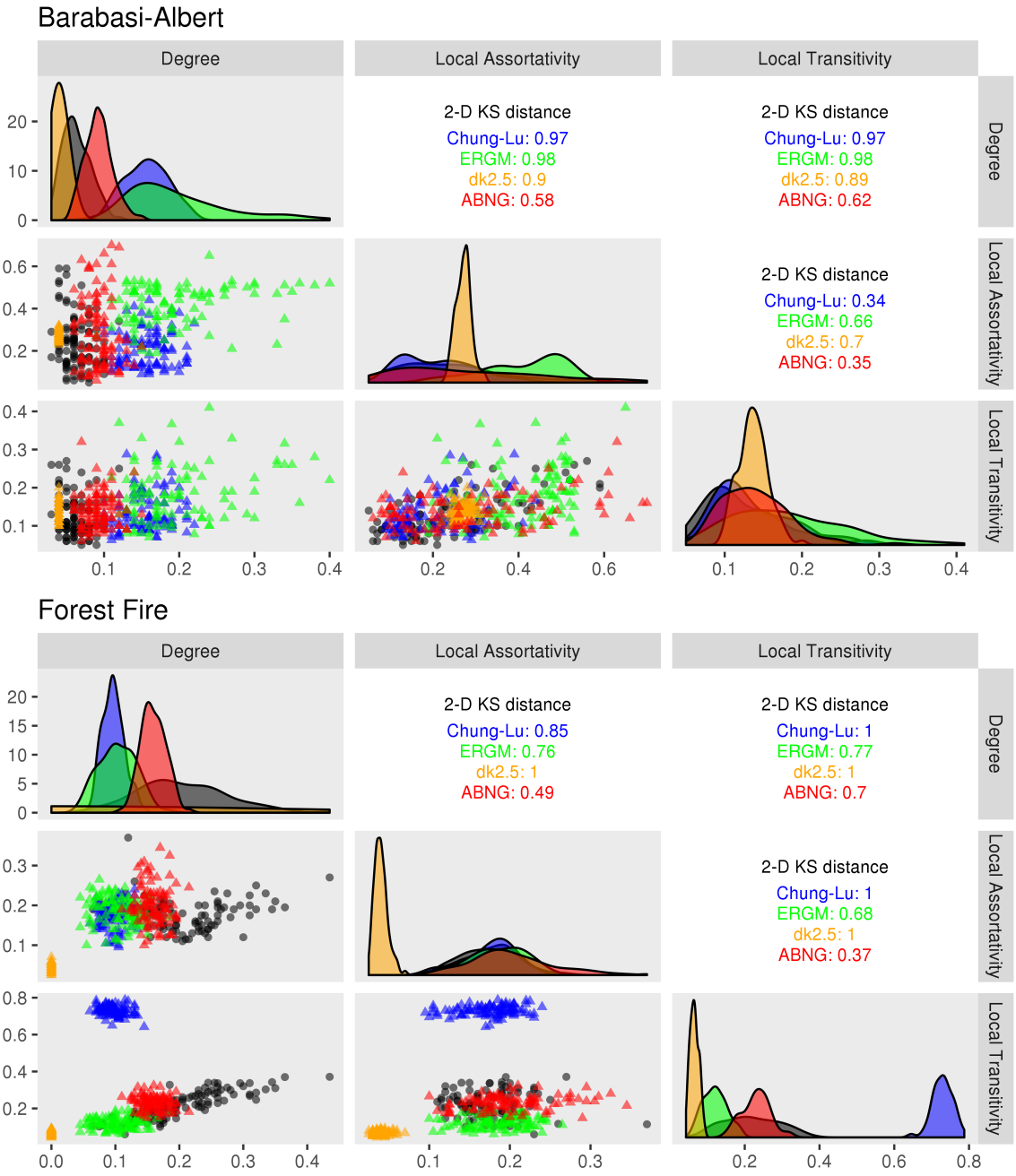


Figure 7.3. Empirical evaluation of the ability of network models to approximate the ground truth system based on observation of a single network. The Barabási-Albert and Forest Fire models are used as the true generators.

7.3 Experimental results

In our experiments to evaluate the distributional properties of generative models, we propose to use the Kolmogorov-Smirnov statistic for evaluating the dissimilarity between networks based on node-level properties of degree, correlations and clustering. To examine the ability of existing generative models to approximate the ground truth process using a single network observation (assuming it is representative of the true process with respect to the measures of interest), we propose two different experiments: (i) a controlled experiment where a known process is used to create a population of networks, and (ii) set of real-world networks that have most likely evolved from a common generative process (for example, social interaction networks of different villages) or generative processes that share the same mechanisms.

7.3.1 Networks without community structure

Figure 7.3 shows the results for the first set of experiments when the Barabási-Albert [12] and Forest Fire models [212] are used as the true processes A^*. For the second experiment, we consider the five real-world network populations described above, with results presented in Figures 7.4, 7.5 and 7.7. Results presented in Figures 7.3-7.7 are composed of three different plots:

1. Scatter plots below the diagonal show each synthesized/real network as a point in the network dissimilarity space, where the coordinates are computed using the Kolmogorov-Smirnov distance of the associated properties when the network is compared to the observed network G^* (the observed network itself is at the (0,0) position). Network models (colored triangles) showing higher overlap with networks originating from the true process (black dots) are better.

2. In the blocks above the diagonal, we evaluate the amount of overlap between $\mathbb{P}_{\mathfrak{D}_{G^*}}(A^*)$ and $\mathbb{P}_{\mathfrak{D}_{G^*}}(M)$ using the 2-D KS distance [272] (lower the better). This

quantifies the extent to which a given generative model is able to reproduce the distributional properties of the population representing the true process.

3. Plots along the diagonal show the density distributions of the Kolmogorov-Smirnov distance of the associated properties when the network is compared to the observed network G^*. Similar density distribution to the ground truth (black curves) implies good match in the properties.

Based on Figures 7.3 and 7.4 we can easily conclude that ABNG consistently outperforms the other models considered here by replicating the natural variability of network populations when computed in the dissimilarity space for both the experimental settings. For the social networks in Indian villages and Travian-Trades networks, ERGM generates dense graphs (causing spikes to the right in degree KS plot) because of model degeneracy. The plots also show that dk-random graphs, which are considered to be the state-of-art, fail to capture the variability of the true generative process and potentially over-fit the observed network. This leads us to question the fundamental idea behind dk-random graphs, i.e. whether exactly preserving the distribution of differently sized subgraphs of a given network leads to a good model for real-world networks. In fact, in most cases we see that the Chung-Lu model, by matching the degree distribution in expectation, outperforms dk-random graphs by synthesizing networks with more variability. These results highlight the need for evaluating the ability of a generative model to capture the distributional properties of a network population as comparing only with the observed network might produce misleading results.

The results in Figures 7.3 and 7.4 suggest that network models, when carefully designed, can potentially capture the structural variability in network populations (when evaluated in the dissimilarity space) by using a single network as input. But when this analysis was extended to more complicated network populations, all network models failed to synthesize populations that resemble the original one, and the results are presented in Figure 7.5.

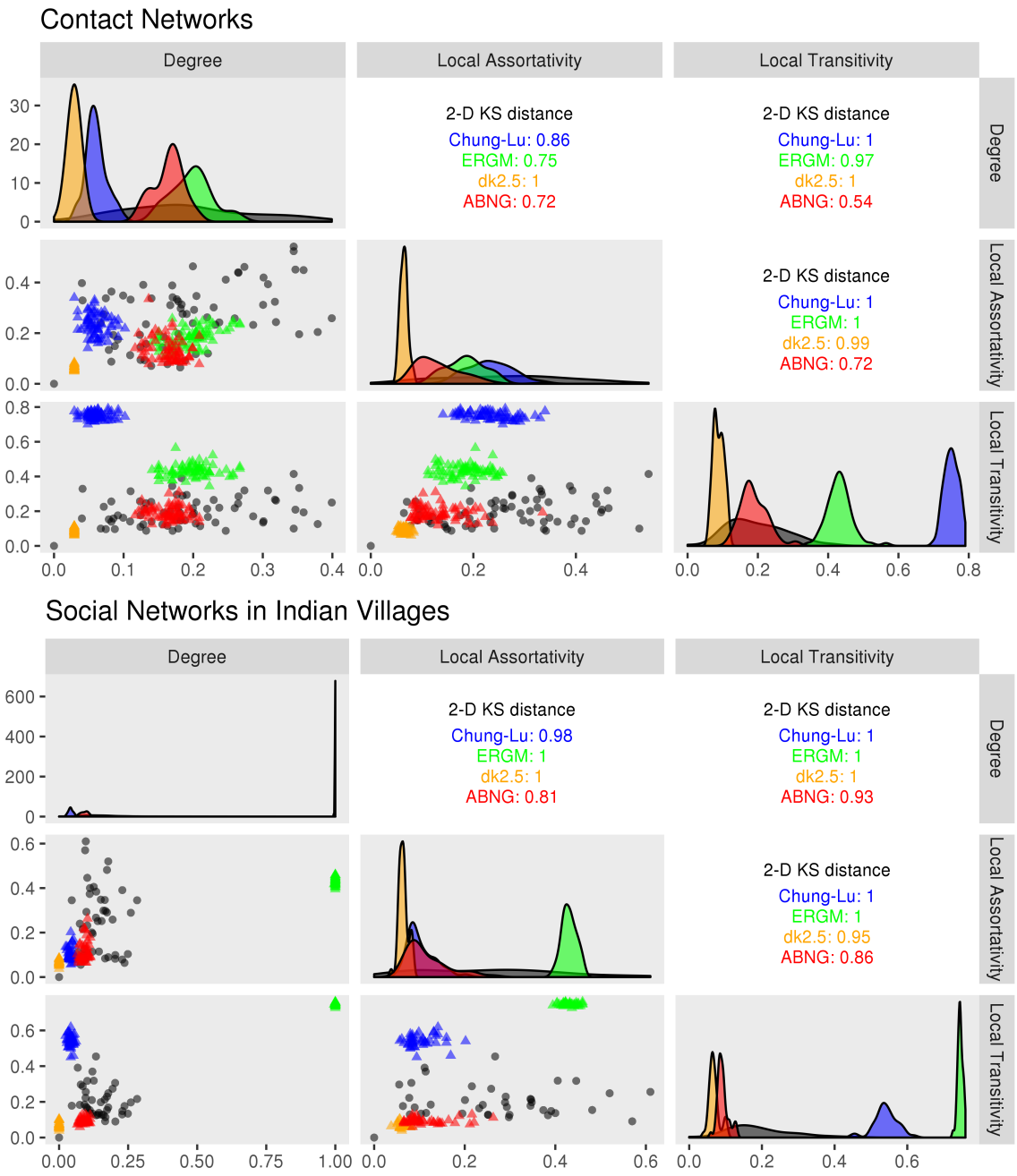


Figure 7.4. Empirical evaluation of the ability of network models to approximate the ground truth system based on observation of a single network. Two real-world datasets were considered: contact networks, and social networks in Indian villages.

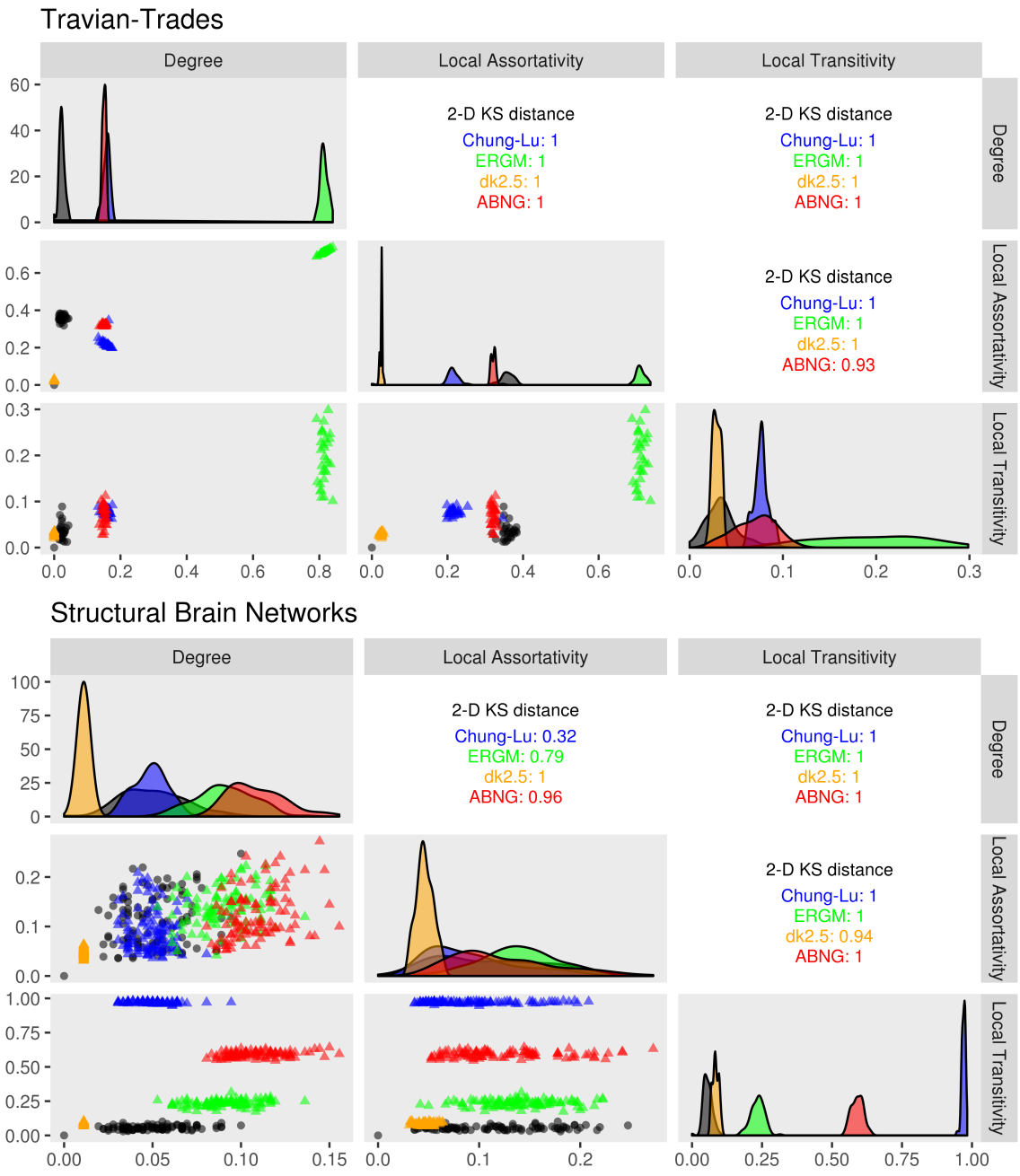


Figure 7.5. Empirical evaluation of the ability of network models to approximate the ground truth system based on observation of a single network. Two real-world datasets were considered: Travian trades and structural brain networks.

7.3.2 Networks with community structure

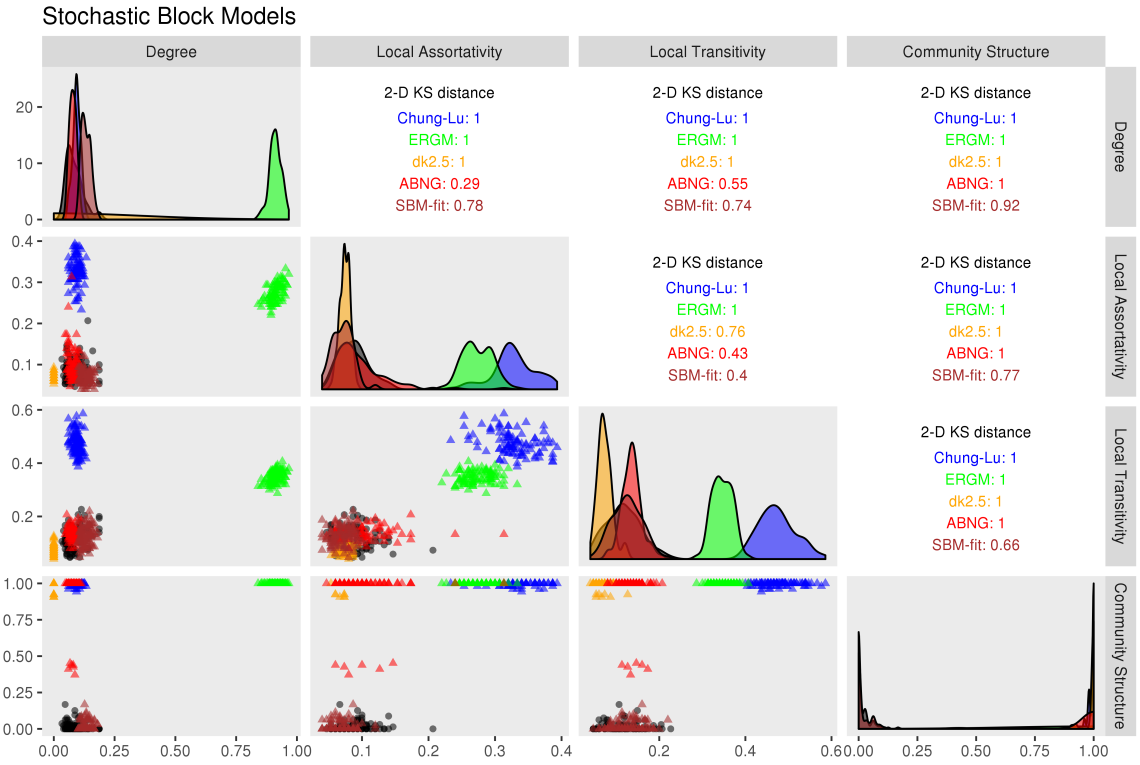


Figure 7.6. Empirical evaluation of the ability of network models to approximate the ground truth system based on observation of a single network. The stochastic block model is used as the true generator, and the ability of different models to replicate the community structure is tested.

While the network dissimilarity space defined in Section 7.2 works well for networks without communities, it will prove ineffective for networks with community structures, which is a property seen in most real-world networks [140]. In this section, we extend the network dissimilarity space by adding a fourth dimension to compare the community structures of two networks. The following procedure was used for comparing community structures in the extended network dissimilarity space:

1. Compute node memberships using a community detection algorithm (Infomap community detection algorithm [298] was used in our experiments).

2. Sort the communities based on sizes, i.e. community 1 is the largest community.

3. Compare the sorted memberships using the normalized mutual information measure [299].

We also add the microcanonical stochastic block model [300] (referred to as SBM-fit in the plots) to our set of generative models and evaluate its ability to replicate the community structure of these networks.

Again, we performed two different experiments to test the validity of our extended network dissimilarity space: (i) a controlled experiment where the true process is known, and (ii) set of real-world networks (with communities) that have most likely evolved from a common generative process. For the first case, we used the standard version of the stochastic block model [142, 143] with 3 communities of different sizes, and the results can be seen in Figure 7.6. As expected, ABNG performs well on the original measures, but fails to reproduce the community structure, while the fitted SBM is the most likely candidate capable of replicating the true process. This is an expected result as the four original models are not designed to create networks with communities. Figure 7.7 shows the results for the networks of Autonomous Systems and Travian messages, where only the fitted SBM was able to capture some of the features of the true process. Results presented in Figure 7.7 show the inability of the microcanonical block model to reproduce the local transitivity of the true generative process, thus creating an exciting direction for future research.

In summary, our empirical analysis in the dissimilarity space has highlighted the discrepancy between observed network population and synthesized network as well as the importance of considering distributional properties of network populations for evaluating generative models of complex networks. This shows that there is an urgent need to rethink the network modeling problem and create new models that can reproduce the variability in the structural properties of network populations.

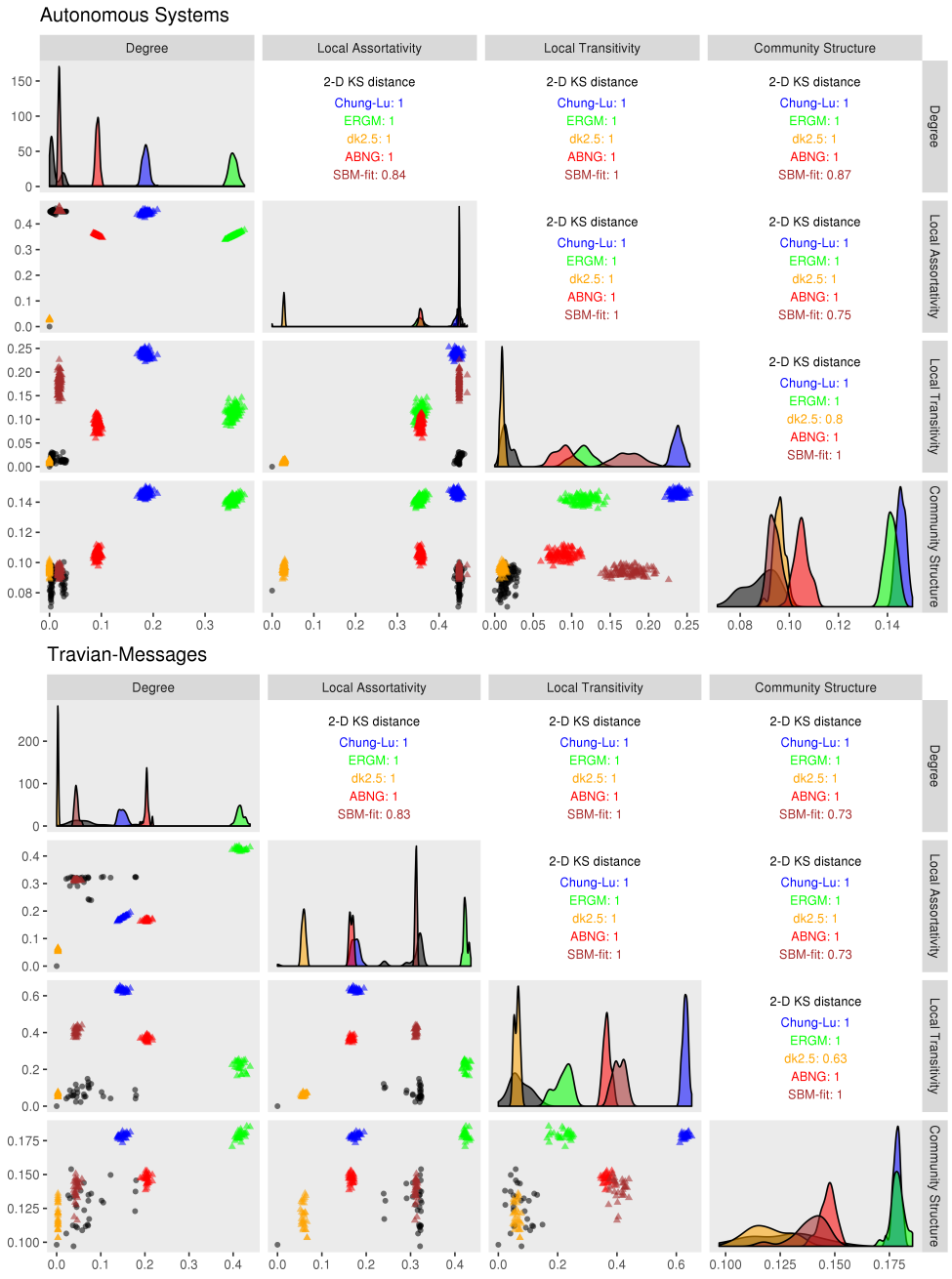


Figure 7.7. Empirical evaluation of the ability of network models to approximate the ground truth system based on observation of a single network. The ability of different models to replicate the community structure of networks of autonomous systems and Travian messages networks is tested.

7.4 Supplementary results

7.4.1 Choice of G^*

For all the experiments conducted in Section 4, the single observed network G^* is randomly chosen among the networks in dataset. However, the pairwise comparison relies on the chosen G^*, and arbitrary choice of G^*, e.g. an outlier, could cause misleading results. In this section, the impact of choice of G^* on the results presented in Section 4 are analyzed. In summary, for the 9 datasets except for autonomous systems, the three tested network metrics are consistent regardless of choice of G^*. For autonomous systems, the difference of variability between network populations generated by true process and generative models is more significant than the difference caused by choice of G^*.

Firstly, we compute the dissimilarity of degree distribution, local assortativity and local transitivity with respect to all networks $\{G \in \mathcal{G}_{A^*}\}$. Specifically, for each network G_i we compute the KS statistics D_i of the three metrics between G and rest of the networks in the population, which follows $\mathbb{P}_{\mathfrak{D}_G}(A^*)$ and check if these distributions are similar. In the context of Figures 4-8 in the main text, we test if the distributions of black dots are robust to the choice of G^*.

Figure 7.8 shows the KS statistic matrices $D(m)$ of the three network metrics m on the 9 network populations. Element $D_{ij}(m)$ represents the KS statistic of metric m between G_i and G_j. Based on the pairwise D statistics, we test if the distributions are similar for different G^*, i.e. if the rows in $D(m)$ are similar to each other.

Figure 7.9 shows the proportion of all tests that fail to reject the null hypothesis that the two samples are drawn from the same distribution. For most populations except autonomous systems, about half of the comparisons show no difference in dissimilarity distributions. In order to test this hypothesis together for all three metrics, multivariate (3 variables) two sample tests are performed with a kernel maximum mean discrepancy (kMMD) test. Similar to the KS test, the null hypothesis is that the two samples are drawn from the same distribution.

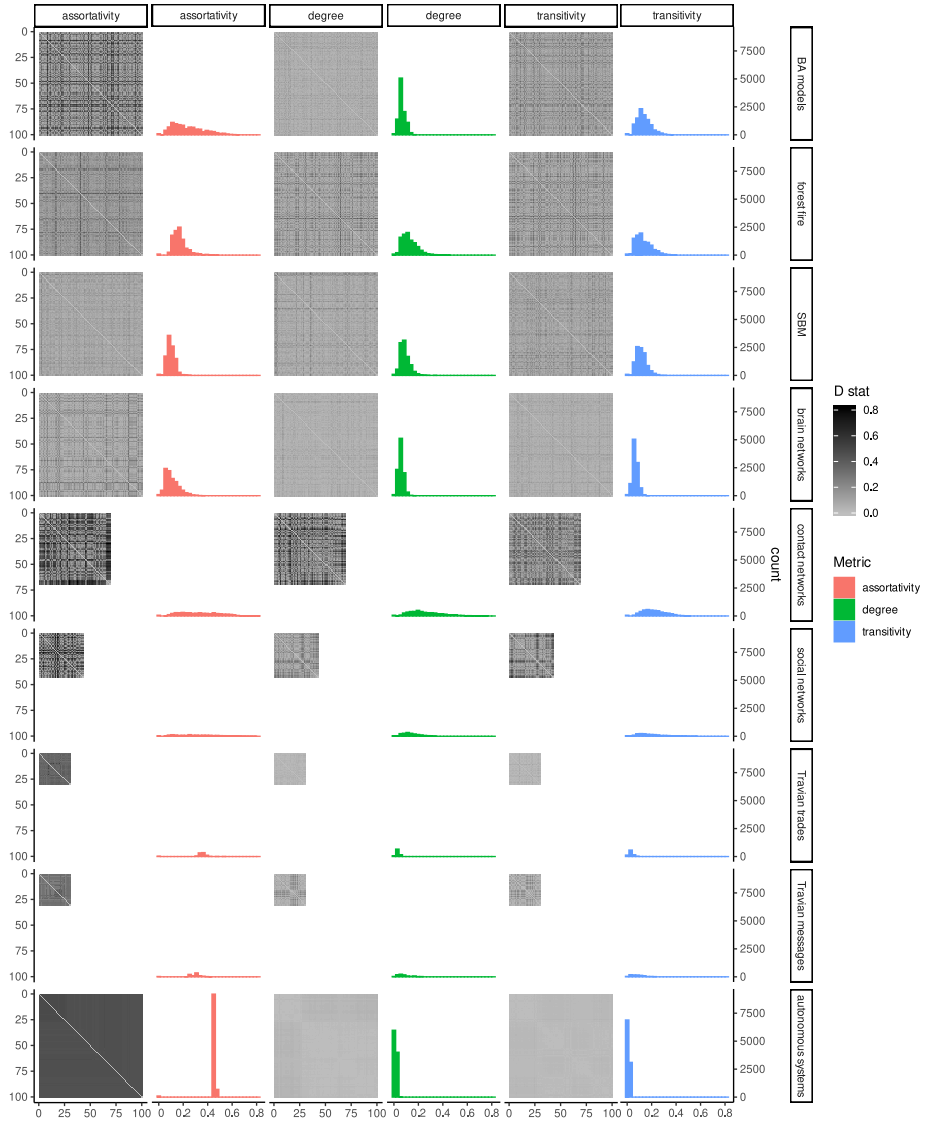


Figure 7.8. The pairwise KS statistic D and histogram of local assortativity, degree distribution and local transitivity on 9 network populations. 9 rows represent 9 sets of network population as labeled on the right. Columns 1, 3, and 5 represent the D statistic matrices of local assortativity, degree distribution and transitivity, respectively. X and Y axes in these columns indicate the number of networks in population. Columns 2, 4, and 6 show the histogram of the D statistic for all possible pairs of networks.

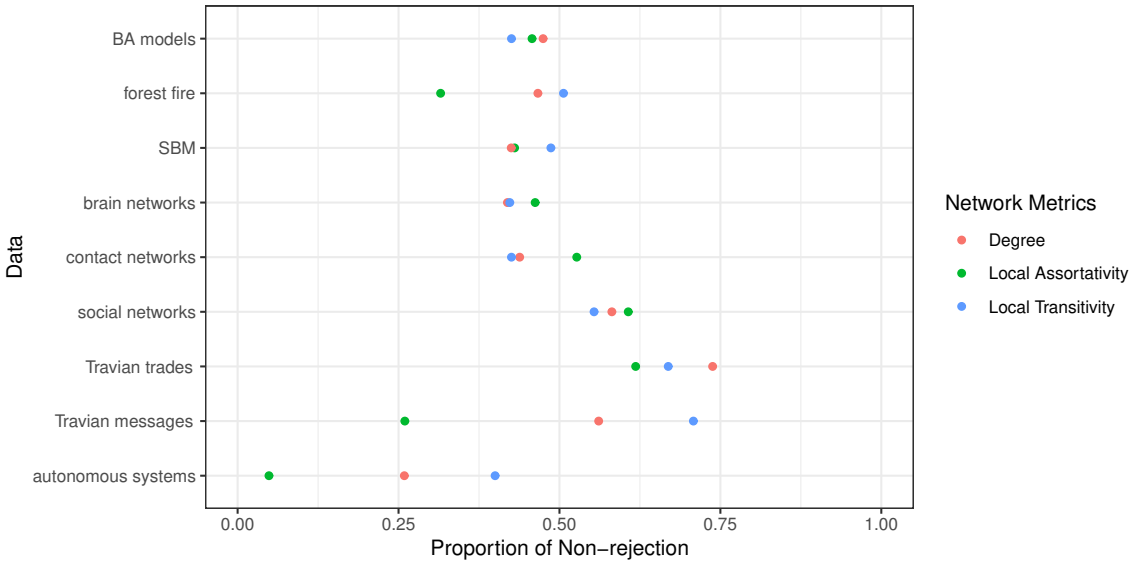


Figure 7.9. Proportion of non-rejection in KS test with 95% confidence level. Higher value indicates more pairs of G^* have same distribution.

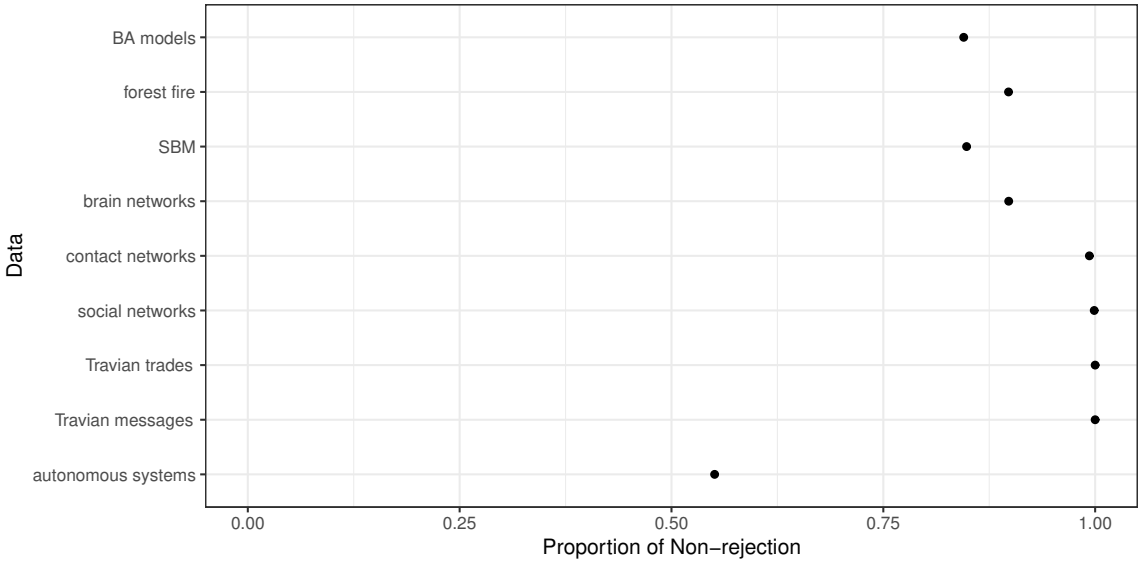


Figure 7.10. Proportion of non-rejection in kMMD test with 95% confidence level. Higher value indicates more pairs of G^* have same distribution.

Figure 7.10 shows the proportion of non-rejection in kMMD test. All populations have a non-rejection rate higher than 85% except autonomous systems. However, rejection in KS test or kMMD test does not revoke the result shown in Section 4 because the difference caused by choice of G^* is often negligible compared to the difference between true process and populations generated from generative models. For instance, the pair of networks in autonomous systems, #7 and #28, have the largest MMD among all pairs.

The results shown in Figure 7.11 implies that the synthetic networks are consistent for different G^*, and the difference between ground truth and model simulation is much larger than the difference caused by different G^*.

7.4.2 ABNG as the true model

Table 7.1.
The table shows action matrices used to test the ability of our action-based approach to reproduces its own variability. The following actions were used: Preferential attachment on - average neighbor degree (PAND), degree (PAD), PageRank (PAPR) and betweenness (PAB); Triadic closure (TC); Inverse log-weighted (SLW) and Jaccard similarities (SJ); and No action (NA).

	PAND	PAD	PAPR	PAB	TC	SLW	SJ	NA	$\bar{P}$
AM1	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	1
AM2	0	0.25	0	0.25	0	0.25	0	0.25	1
AM3	0.226	0.089	0.179	0.176	0.09	0.218	0.004	0.018	1
AM4	0.212	0.095	0.21	0.019	0.181	0.19	0.042	0.051	1
AM5	0.174	0.179	0.152	0.157	0.077	0.154	0.091	0.016	0.618
	0.161	0.125	0.153	0.081	0.088	0.161	0.060	0.171	0.382
AM6	0.241	0.058	0.157	0.216	0.012	0.078	0.143	0.095	0.267
	0.063	0.274	0.047	0.079	0.064	0.218	0.084	0.171	0.184
	0.203	0.166	0.038	0.068	0.037	0.207	0.132	0.148	0.549

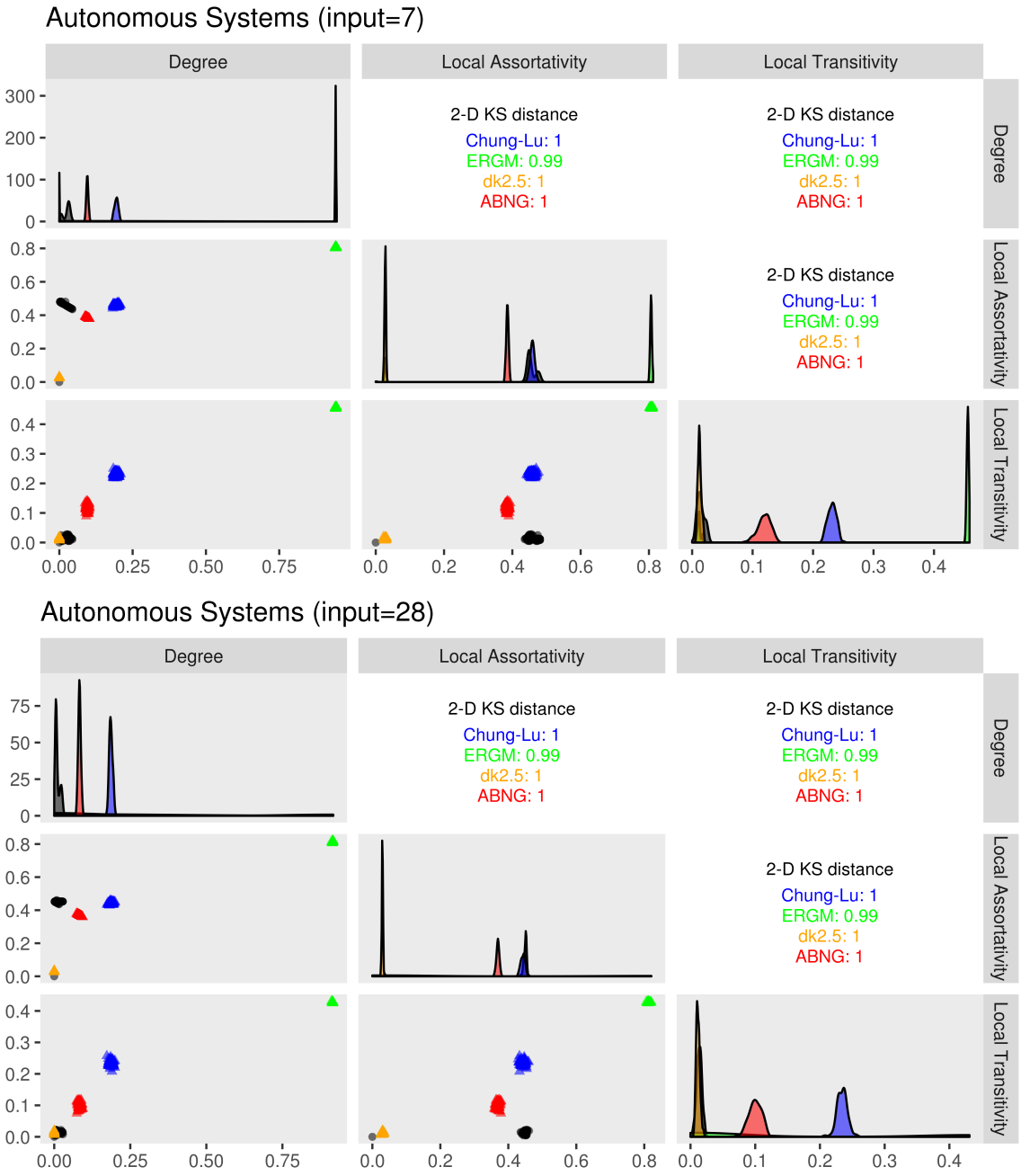


Figure 7.11. Empirical evaluation of the ability of network models to approximate the ground truth system based on observation of #3 network (left) and #28 network (right) in autonomous systems dataset.

The setup proposed in this Chapter raises an important question regarding the evaluation of network models and how they are trained to learn from a single observation. This raises some concerns about how we trained our action-based approach in Chapters 4–6, and used the fitted models to draw conclusions about the underlying network. One way of evaluating the validity of these conclusions is to subject the action-based approach to the analysis proposed in Section 7.2, where an arbitrary action matrix is used to synthesize an input network G^*, which is then used by the learning framework of the action-based approach introduced in Chapter 4 to parameterize the model. There are two questions one might ask: (i) can the fitted action-based model reproduce the variability in the networks produced using the original input action matrix, and (ii) can the learning framework recover the original action matrix using a single input network G^*.

In this section, we perform experiments with the goal of obtaining preliminary insights into the first question. We synthesized network populations using six different action matrices (see Table 7.1), randomly chose a network as G^* and then used it to learn action-based models. The results for these experiments are shown in Figure 7.12, and it is clear that ABNG can indeed reproduce the variability of the original population synthesized using a known action matrix.

7.5 Conclusions

Traditional approaches for evaluating the ability of a network model to synthesize networks exhibiting real-world characteristics have compared the similarity of the synthesized networks with the observed network. While this approach assumes that the particular observation is representative of the underlying process that created the observation, it does not account for the natural variability of the population from which the original network is sampled. Our empirical experiments have highlighted the importance of considering network populations for evaluating generative models. Although it is difficult to obtain data corresponding to network populations, we have

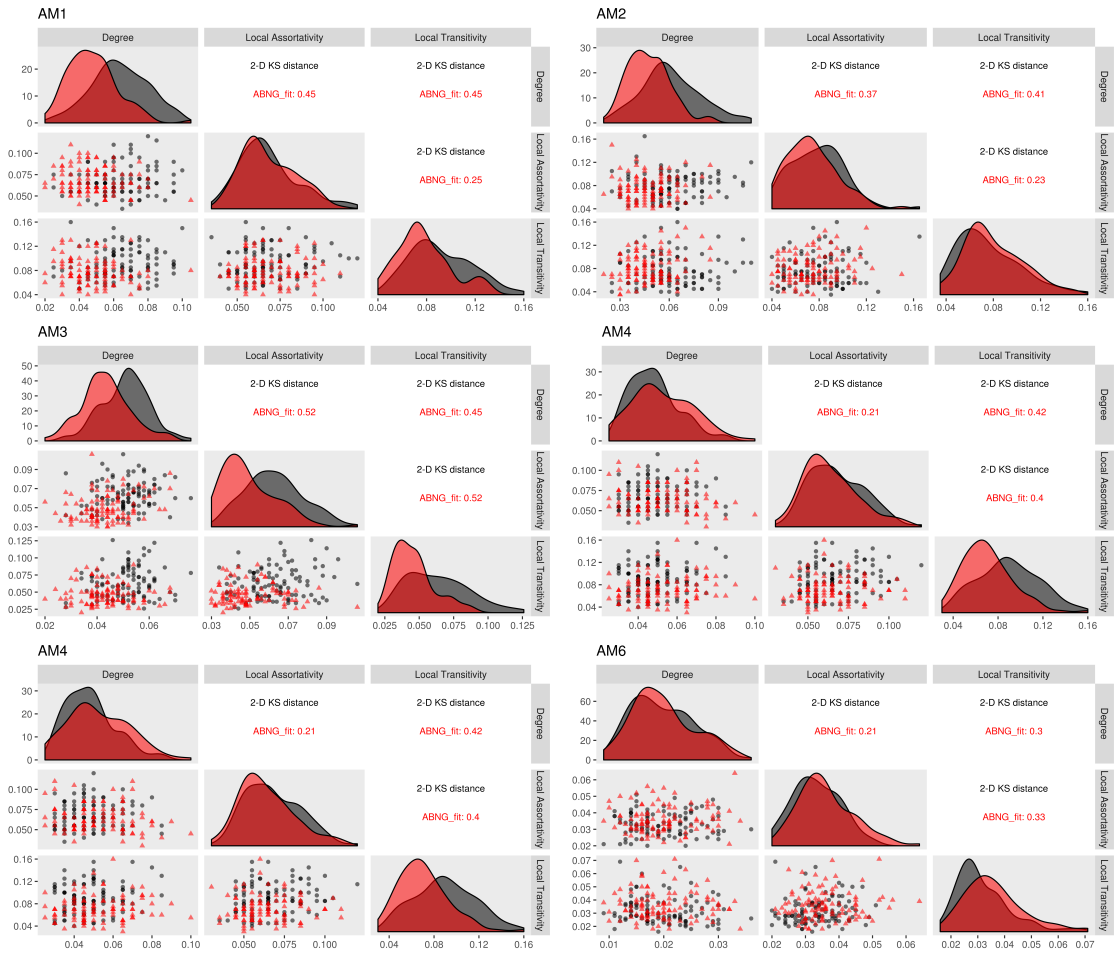


Figure 7.12. Empirical evaluation of the action-based approach to reproduce the variability of a network population synthesized using a know action matrix. We considered six different action matrices to create the network populations, and they are listed in Table 7.1.

shown that it is possible to establish a baseline test set to evaluate the ability of a network model to capture the distribution of network populations. This test set can then be used for preliminary validation of a network model before it is used for drawing conclusions about real-world networks.

The need for devising generative models for network populations is raised in this Chapter as the state-of-the-art network models are unable to replicate the properties of network populations. Instead of modifying and transforming existing network

generative models to fit the representative/average network of a network population, we need to develop models that can extract and quantify the variability existing in network populations. For example, Bayesian modeling frameworks are being successfully used to learn the joint distribution of edges in network population that shares the same set of vertices [288–290], though the topology of networks is not explicitly fitted. Future work on the development of unbiased measures of variability in network populations, accompanied with a quantitative analysis of the effect of network sample size and power can guide the sampling process of networks from a population and the development of improved generative network models.

8. CONCLUSIONS AND FUTURE WORK

Our efforts to develop a generalized representation for complex networks using the action-based approach has provided insights that can help researchers understand the principles by which a network is organized and shed some light on the mechanisms by which it grows and develops. In what follows, we present some conclusions that highlight the contributions of this work and discuss a few directions for future work.

8.1 Conclusions

We first demonstrated that by embracing the complexity in modern network data and utilizing the theoretical and empirical observations made by network scientists in last few decades, a mechanistic model can be extrapolated that provides a unique way of understanding potential processes (actions) that govern interactions in real-world networks. This led to the notion of node-type as a statistical unit for network data, which contains information about the data modeling process, while also providing an intuitive representation for networks data. Different actions can then be utilized within a forward operator (synthesis algorithm) that can be parameterized to synthesize networks exhibiting a wide variety of topologies. Given a set of actions, we found that choosing an appropriate forward operator can lead to creation of a model that is projective and exchangeable relative to a fixed structure that accounts for the node-type assignment and any potentially useful node attributes.

To exploit the capabilities of the action-based framework in empirical settings, we need an approach for estimating model parameters from a single network observation. Mechanistic models, such as the one proposed in this work, typically have intractable likelihoods but are easy to forward simulate. Consequently, the problem of estimating model parameters consists of two parts: (i) computation of dissimilarity using a user-

defined set of measures, and (ii) an optimization technique to learn parameters for a given target network. Due to lack of a subset of network properties that can capture the dissimilarity between networks, we provide the user with the flexibility of choosing these measures, and thus formulate the problem of parameterizing the model as a multi-objective optimization problem. Using this formulation, we were able to learn compressed models for networks synthesized using other models as well as real-world networks originating in different domains. We further demonstrated that the representation of a network using the action matrix can yield insights into the structural organization of these networks.

Following the demonstration of the action-based framework as a successful model for real-world networks, we decided to explore the possibility of incorporating domain knowledge under specific application contexts. We first considered the application of the action-based model to directed networks, particularly for the case of supply chain networks. Using domain specific constraints and relevant actions, we were able to devise a centralized approach for designing realistic supply chains that are resilient under attack. This can facilitate our understanding of critical infrastructure and help us make informed decisions regarding design of such systems. In Chapter 6, we concluded that combining actions with spatial information can help us learn better models for networks where the nodes are embedded in space. We applied this idea to structural brain networks and found that apart from learning better models that can capture the between-subject variability, the model parameters can provide useful insights about the cognitive ability of the subject. The model can also prove beneficial in other domains, which remains a subject for future research.

During our research on network models, we also realized the importance of analyzing network populations and the need for incorporating variability in generative models. Using our data-driven methods for quantifying the variability of network populations, we concluded that there is a need for devising generative models that can synthesize realistic network population, that is, a network model should be able to reproduce desired structural properties and the variability in these properties. In-

stead of modifying and transforming existing network models to fit a representative network from the population, we need to develop models that can extract and quantify the variability that exists in real-world network populations.

Overall, I firmly believe that ABM is a significant contribution in network science, and published work has statistically demonstrated its state-of-the-art performance, while also being easy for non-experts to interpret and draw hypotheses about the system being modeled. We believe that our research will motivate network scientists to think about processes governing network formation, thus providing a fresh perspective for the development of future network models. A few approaches have already been proposed that draw inspiration from the action-based model, and hopefully our efforts will inspire other researchers to develop fresh perspectives for understanding network data. For example, Netmix [301] combines different generative models (instead of actions) using evolutionary algorithms to synthesize networks, while [302] used random utility theory and discrete choice models (actions are used for making choices) to provide a framework for synthesizing temporal networks.

8.2 Future work

An important feature of most real-world networks is the existence of community structures [32, 33, 67, 136]. From a mechanistic perspective, one might be interested in understanding the processes that lead to formation of communities in real-world networks. Chapter 6 provided a way of modeling networks embedded in space using the concept of visibility in a continuous space. An obvious extension is to use the concept of visibility in discrete spaces using node attributes with the goal of explicitly modeling community structures. Such an endeavour would require careful examination of the node attributes and their relationship with the structure of the observed network.

Our work in Chapter 7 highlighted the lack of variability in populations synthesized by network models. To improve our models, we need to devise techniques that

can allow existing models to utilize additional information from the population. Recent research has focused on using multiple input networks in Bayesian frameworks to improve the capability of network models to synthesize realistic populations. While parameterizing models using multiple networks as input [288–291] can enable existing models to learn better representation of the population, such techniques fail to explicitly account for the natural variability in the population. Data-driven techniques to quantify the information content and variability in network populations can aid the rapid development of more explanatory network models that capture the distributional properties and variability of network populations.

Accurate measurement of the complexity of a complex system can uncover key insights about its behavior. Many approaches have been proposed, many that concentrate on estimating the complexity of a specific system observation in the form of a network. The action-based representation of networks provides a way for computing the generative potential of a system using Shannon-based entropy. Developing a measure for quantifying the generative entropy of a complex system using the action-based representation will possess some desirable properties, such as, scale invariance, and the ability to compare the complexity across different systems. Our preliminary work on this topic has shown promising results.

The action-based approach can also be visualized as a probabilistic model that uses a portfolio of link prediction algorithms to learn a representation of networks. This can prove particularly useful for the problem of estimating the structure of a network using noisy measurements [303, 304]. Another often overlooked fact about network data is that most real-world interactions are much more complicated than simple pairwise interactions–for example, most systems originate from higher-order interactions between more than two individuals [305, 306] (paper co-authorship, communication within a group, etc.). Similarly, it is well known that complex systems do not work in isolation, and their interconnected and interdependent nature is better represented using a multilayer network framework [307–309]. A crucial next step would be to think of actions as process that can accommodate such higher-order

interactions, and help us understand the processes and mechanisms that drive the creation of the corresponding systems.

REFERENCES

REFERENCES

[1] Mark Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[2] Albert-Laszlo Barabasi. *Network science*. Cambridge University Press, 2016.

[3] Hiroki Sayama. *Introduction to the Modeling and Analysis of Complex Systems*. Open SUNY Textbooks, 2015.

[4] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, 1 2003.

[5] Anna Goldenberg, Alice X Zheng, Stephen E Fienberg, and Edoardo M Airoldi. A Survey of Statistical Network Models. *Foundations and Trends® in Machine Learning*, 2(3):129–233, 12 2009.

[6] Ljupco Kocarev and Visarath In. Network science: A new paradigm shift. *IEEE Network*, 24(6):6–9, 11 2010.

[7] Albert-Laszlo Barabasi. The network takeover. *Nature Physics*, 8(1):14–16, 1 2012.

[8] ULRIK BRANDES, GARRY ROBINS, ANN McCRANIE, and STANLEY WASSERMAN. What is network science? *Network Science*, 1(01):1–15, 4 2013.

[9] Mark Newman, Albert-Laszlo Barabasi, and Duncan J Watts. *The structure and dynamics of networks*. Princeton University Press, 2006.

[10] D Chakrabarti and C Faloutsos. Graph Mining: Laws, Generators, and Algorithms. *ACM Computing Surveys*, 38(1), 2006.

[11] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker Graphs: An Approach to Modeling Networks. *The Journal of Machine Learning Research*, 11:985–1042, 2010.

[12] Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 10 1999.

[13] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–2, 6 1998.

[14] Albert-Laszlo Barabasi. *Linked: The New Science of Networks*. Basic Books, 1st edition, 5 2002.

[15] David L. Alderson. OR FORUMCatching the Network Science Bug: Insight and Opportunity for the Operations Researcher. *Operations Research*, 56(5):1047–1065, 10 2008.

[16] Toni Vallès-Català, Tiago P Peixoto, Marta Sales-Pardo, and Roger Guimerà. Consistencies and inconsistencies between model selection and link prediction in networks. *Physical Review E*, 97(6):062316, 6 2018.

[17] Sixing Chen, Antonietta Mira, and Jukka-pekka Onnela. Flexible model selection for mechanistic network models. *Journal of Complex Networks*, pages 1–18, 8 2019.

[18] P D Hoff, A E Raftery, and M S Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

[19] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p*) models for social networks. *Social Networks*, 29(2):173–191, 5 2007.

[20] Eric D. Kolaczyk. *Statistical Analysis of Network Data*. Springer Series in Statistics. Springer New York, New York, NY, 2009.

[21] Eric D Kolaczyk. *Topics at the Frontier of Statistics and Network Analysis*. SemStat Elements. Cambridge University Press, Cambridge, 6 2017.

[22] P Erdös and A Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[23] P. Erdös and A. Rényi. On the Evolution of Random Graphs. *PUBLICATION OF THE MATHEMATICAL INSTITUTE OF THE HUNGARIAN ACADEMY OF SCIENCES*, 5(1):17–61, 1960.

[24] E N Gilbert. Random Graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[25] R Solomonoff and A Rapoport. Connectivity of Random Nets. *Bulletin of Mathematical Biology*, 13(2):107–117, 1951.

[26] Albert-Laszlo Barabasi. Scale-Free Networks: A Decade and Beyond. *Science*, 325(5939):412–413, 7 2009.

[27] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 27(1):415–444, 8 2001.

[28] M. E. J. Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, 2 2003.

[29] Sergei Maslov and Kim Sneppen. Specificity and Stability in Topology of Protein Networks. *Science*, 296(5569):910–913, 2002.

[30] Romualdo Pastor-Satorras, Alexei Vázquez, and Alessandro Vespignani. Dynamical and correlation properties of the internet. *Physical Review Letters*, 87(25):258701–1, 2001.

[31] R Milo, S Shen-Orr, S Itzkovitz, and N Kashtan. Network Motif: Simple Building Blocks of Complex Networks. *Science*, 298(5594):298., 2002.

[32] Gergely Palla, Imre Derényi, Ills Farkas, and Tams Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 6 2005.

[33] Jrg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1):016110, 7 2006.

[34] Mark Newman. The physics of networks. *Physics Today*, 61(11):33–38, 11 2008.

[35] S Milgram. The Small World Problem. *Psychology Today*, 2(1):60–67, 1967.

[36] J Travers and S Milgram. An Experimental Study of the Small World Problem. *Sociometry*, 32(4):425–443, 1969.

[37] A Barrat and M Weigt. On the properties of small-world network models. *The European Physical Journal B*, 13:547–560, 2000.

[38] B Bollobás. *Random Graphs*. Academic Press, 1985.

[39] J Kleinberg. The Small-world Phenomenon: An Algorithmic Perspective. In *Thirty-second Annual ACM Symposium on Theory of Computing*, pages 163–170, 2000.

[40] Udny G Yule. A Mathematical Theory of Evolution, Based on the Conclusions of Dr. J. C. Willis, F.R.S. *Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of a Biological Character*, 213:21–87, 1925.

[41] D De Solla Price. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science*, 27(5):292–306, 1976.

[42] B Bollobás, O Riordan, J Spencer, and G E Tusnády. The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, 18(3):279–290, 2001.

[43] M Boguñá and R Pastor-Satorras. Class of correlated random networks with hidden variables. *Physical Review E*, 68(3):36112, 2003.

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

[45] N C Wormald. *Some Problems in the Enumeration of Labelled Graphs*. PhD thesis, University of Newcastl, 1978.

[46] W Aiello, F Chung Graham, and L Lu. A Random Graph Model for Power Law Graphs. *Experimental Mathematics*, 10(1):53–66, 2001.

[47] F Chung and L Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002.

[48] F Chung and L Lu. Connected Components in Random Graphs with Given Expected Degree Sequences. *Annals of Combinatorics*, 6(2):125–145, 2002.

[49] Alexander Gutfraind, Lauren Ancel Meyers, and Ilya Safro. Multiscale Network Generation. *arXiv:1207.4266*, page 28, 7 2012.

[50] D Strauss. On a General Class of Models for Interaction. *SIAM Review*, 28(4):513–527, 1986.

[51] S Wasserman and P Pattison. Logit models and logistic regressions for social networks. *Psychometrika*, 60:401–425, 1996.

[52] Carolyn J Anderson, Stanley Wasserman, and Bradley Crouch. A p* primer: logit models for social networks. *Social Networks*, 21(1):37–66, 1999.

[53] J Leskovec, D Chakrabarti, J Kleinberg, and C Faloutsos. Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication. In *9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 133–145, 2005.

[54] J Leskovec, J Kleinberg, and C Faloutsos. Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. In *Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 177–187, 2005.

[55] J Leskovec and C Faloutsos. Scalable Modeling of Real Graphs Using Kronecker Multiplication. In *24th International Conference on Machine Learning*, pages 497–504, 2007.

[56] Alexander Bailey, Mario Ventresca, and Beatrice Ombuki-Berman. Genetic Programming for the Automatic Inference of Graph Models for Complex Networks. *IEEE Transactions on Evolutionary Computation*, 18(3):405–419, 6 2014.

[57] T Menezes and C Roth. Symbolic regression of generative network models. *Nature Scientific Reports*, 4, 2014.

[58] Aaron S. Pope, Daniel R. Tauritz, and Alexander D. Kent. Evolving random graph generators: A case for increased algorithmic primitive granularity. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 12 2016.

[59] Telmo Menezes and Camille Roth. Automatic Discovery of Families of Network Generative Processes. In *Dynamics On and Of Complex Networks III*, pages 83–111, 2019.

[60] Harry Crane and Walter Dempsey. A framework for statistical network modeling. *arXiv*, pages 1–33, 9 2015.

[61] N. Przulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 1 2007.

[62] S Roy. Systems biology beyond degree, hubs and scale-free networks: the case for multiple metrics in complex networks. *Systems and Synthetic Biology*, 6(1-2):31–34, 2012.

[63] mer Nebil Yaveroğlu, Nol Malod-Dognin, Darren Davis, Zoran Levnajic, Vuk Janjic, Rasa Karapandza, Aleksandar Stojmirovic, and Nataa Pržulj. Revealing the Hidden Language of Complex Networks. *Scientific Reports*, 4(1):4547, 5 2015.

[64] Abigail Z. Jacobs and Aaron Clauset. A unified view of generative models for networks: models, methods, opportunities, and challenges. In *NIPS Workshop on Networks: From Graphs to Rich Data*, pages 1–10, 2014.

[65] Victor Veitch and Daniel M. Roy. The Class of Random Graphs Arising from Exchangeable Random Measures. *arXiv preprint*, math.ST, 12 2015.

[66] Duncan J. Watts. A twenty-first century science. *Nature*, 445(7127):489, 2007.

[67] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2 2010.

[68] Garry Robins and Martina Morris. Advances in exponential random graph (p*) models. *Social Networks*, 29(2):169–172, 2007.

[69] George E P Box and Norman R Draper. *Empirical model-building and response surfaces.* John Wiley \& Sons, 1987.

[70] S. Havlin, D. Y. Kenett, E. Ben-Jacob, A. Bunde, R. Cohen, H. Hermann, J. W. Kantelhardt, J. Kertész, S. Kirkpatrick, J. Kurths, J. Portugali, and S. Solomon. Challenges in network science: Applications to infrastructures, climate, social systems and economics. *European Physical Journal: Special Topics*, 214(1):273–293, 2012.

[71] James Ladyman, James Lambert, and Karoline Wiesner. What is a complex system? *European Journal for Philosophy of Science*, 3(1):33–67, 1 2013.

[72] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 5 2007.

[73] P. Erdős and A. Rényi. On the strength of connectedness of a random graph. *Acta Mathematica Academiae Scientiarum Hungaricae*, 12(1-2):261–267, 3 1964.

[74] Bla Bollobás. The diameter of random graphs. *Transactions of the American Mathematical Society*, 267(1):41–41, 1 1981.

[75] Reuven Cohen and Shlomo. Havlin. *Complex networks : structure, robustness, and function*. Cambridge University Press, 2010.

[76] Alessandro Vespignani. Twenty years of network science. *Nature*, 2018.

[77] Anna D Broido and Aaron Clauset. Scale-free networks are rare. *Nature Communications*, 10(1):1017, 12 2019.

[78] Ivan Voitalov, Pim van der Hoorn, Remco van der Hofstad, and Dmitri Krioukov. Scale-free networks well done. *Physical Review Research*, 1(3):033034, 10 2019.

[79] Petter Holme. Rare and everywhere: Perspectives on scale-free networks. *Nature Communications*, 10(1):1016, 12 2019.

[80] Carlos Herrera and Pedro J. Zufiria. Generating scale-free networks with adjustable clustering coefficient via random walks. In *2011 IEEE Network Science Workshop*, pages 167–172. IEEE, 6 2011.

[81] Z Wu, G Menichetti, C Rahmede, and G Bianconi. Emergent complex network geometry. *Sci Rep*, 5:10073, 2015.

[82] Remco van der Hofstad. Random graphs and complex networks. *Lecture notes, available online at: http://www. win. tue. nl/ rhofstad/NotesRGCN2011. pdf*, 2, 2011.

[83] Sergio Currarini, Matthew O. Jackson, and Paolo Pin. An Economic Model of Friendship: Homophily, Minorities, and Segregation. *Econometrica*, 77(4):1003–1045, 2009.

[84] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets*. Cambridge University Press, Cambridge, 2010.

[85] Leto Peel, Jean-Charles Delvenne, and Renaud Lambiotte. Multiscale mixing patterns in networks. *Proceedings of the National Academy of Sciences*, 115(16):4057–4062, 4 2018.

[86] Paolo Pin and Brian Rogers. Stochastic Network Formation and Homophily. *The Oxford Handbook of the Economics of Networks*, 1(May):1–35, 6 2016.

[87] Gueorgi Kossinets and Duncan J. Watts. Empirical Analysis of an Evolving Social Network. *Science*, 311(5757):88–90, 1 2006.

[88] Peter D. Hoff. Modeling homophily and stochastic equivalence in symmetric relational data. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pages 1–8, 11 2007.

[89] Yuxiao Dong, Reid A. Johnson, Jian Xu, and Nitesh V. Chawla. Structural Diversity and Homophily: A Study Across More than One Hundred Large-Scale Networks. *arXiv*, page 11, 2 2016.

[90] Erzsbet Ravasz and Albert-Lszl Barabási. Hierarchical organization in complex networks. *Physical Review E*, 67(2):026112, 2 2003.

[91] Herbert A. Simon. The Architecture of Complexity. *American Philosophical Society*, 106(6):467–482, 1962.

[92] Bernat Corominas-Murtra, J. Goni, R. V. Sole, and C. Rodriguez-Caso. On the origins of hierarchy in complex networks. *Proceedings of the National Academy of Sciences*, 110(33):13316–13321, 8 2013.

[93] Aaron Clauset, Cristopher Moore, and M. E J Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

[94] Alexei Vázquez, Romualdo Pastor-Satorras, and Alessandro Vespignani. Large-scale topological and dynamical properties of the Internet. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 65(6):1–12, 2002.

[95] E Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabasi. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555, 9 2002.

[96] Marta Sales-Pardo, Roger Guimera, Andr A. Moreira, and Lus A. Nunes Amaral. Extracting the hierarchical organization. *Proceedings of the National Academy of Science*, 104(39):15224–15229, 2007.

[97] Luciano Da Fontoura Costa. The hierarchical backbone of complex networks. *Physical Review Letters*, 93(9):1–4, 2004.

[98] Ala Trusina, Sergei Maslov, Petter Minnhagen, and Kim Sneppen. Hierarchy measures in complex networks. *Physical Review Letters*, 92(17):2–5, 2004.

[99] Enys Mones, Lilla Vicsek, and Tams Vicsek. Hierarchy Measure for Complex Networks. *PLoS ONE*, 7(3):e33799, 3 2012.

[100] E Bullmore and O Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

[101] F Papadopoulos, M Kitsak, M A Serrano, M Boguna, and D Krioukov. Popularity versus similarity in growing networks. *Nature*, 489(7417):537–540, 2012.

[102] Chaoming Song, Shlomo Havlin, and Hernn a. Makse. Origins of fractality in the growth of complex networks. *Nature Physics*, 2(4):275–281, 4 2006.

[103] G Bianconi, R K Darst, J Iacovacci, and S Fortunato. Triadic closure as a basic generating mechanism of communities in complex networks. *Physical Review E*, 90(4):42806, 2014.

[104] Alexei Vázquez. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E*, 67(5):056104, 2003.

[105] Jun Wu, Yue-Jin Tan, Hong-Zhong Deng, and Da-Zhi Zhu. A new measure of heterogeneity of complex networks based on degree sequence. In *Unifying Themes in Complex Systems*, pages 66–73. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[106] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2-3):161–180, 3 1995.

[107] Michael Molloy and Bruce Reed. The Size of the Giant Component of a Random Graph with a Given Degree Sequence. *Combinatorics, Probability and Computing*, 7(3):S0963548398003526, 9 1998.

[108] Sergei Maslov, Kim Sneppen, and Alexei Zaliznyak. Detection of topological patterns in complex networks: correlation profile of the internet. *Physica A: Statistical Mechanics and its Applications*, 333(1-4):529–540, 2 2004.

[109] Ali Pinar, C. Seshadhri, Tamara G. Kolda, Ali Pinar, and Tamara G. Kolda. The Similarity Between Stochastic Kronecker and Chung-Lu Graph Models. In Mohammed Zaki, Zoran Obradovic, Pang Ning Tan, Arindam Banerjee, Chandrika Kamath, and Srinivasan Parthasarathy, editors, *Twelfth SIAM International Conference on Data Mining*, pages 1071–1082, Philadelphia, PA, 4 2012. Society for Industrial and Applied Mathematics.

[110] Chiara Orsini, Marija M. Dankulov, Pol Colomer-de Simón, Almerima Jamakovic, Priya Mahadevan, Amin Vahdat, Kevin E. Bassler, Zoltn Toroczkai, Marin Boguñá, Guido Caldarelli, Santo Fortunato, and Dmitri Krioukov. Quantifying randomness in real networks. *Nature Communications*, 6(May):8627, 2015.

[111] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. Systematic topology analysis and generation using degree correlations. *ACM SIGCOMM Computer Communication Review*, 36(4):135, 8 2006.

[112] Isabelle Stanton and Ali Pinar. Constructing and sampling graphs with a prescribed joint degree distribution. *Journal of Experimental Algorithmics*, 17(1):3.1, 2012.

[113] Tiago A. Schieber, Laura Carpi, Albert Díaz-Guilera, Panos M. Pardalos, Cristina Masoller, and Martn G. Ravetti. Quantification of network structural dissimilarities. *Nature Communications*, 8(May 2016):13928, 1 2017.

[114] Viplove Arora and Mario Ventresca. Action-based Modeling of Complex Networks. *Scientific Reports*, 7(1):6673, 12 2017.

[115] Balint Tillman, Athina Markopoulou, Minas Gjoka, and Carter T. Buttsc. 2K+ Graph Construction Framework: Targeting Joint Degree Matrix and Beyond. *IEEE/ACM Transactions on Networking*, 27(2):591–606, 4 2019.

[116] Garry Robins, Tom Snijders, Peng Wang, Mark Handcock, and Philippa Pattison. Recent developments in exponential random graph (p*) models for social networks. *Social Networks*, 29(2):192–215, 5 2007.

[117] Juyong Park and M. E. J. Newman. Statistical mechanics of networks. *Physical Review E*, 70(6):066117, 12 2004.

[118] M S Handcock. Assessing degeneracy in statistical models of social networks. Technical report, Center for Statistics and the Social Sciences, 2003.

[119] Cosma Rohilla Shalizi and Alessandro Rinaldo. Consistency under sampling of exponential random graph models. *The Annals of Statistics*, 41(2):508–535, 4 2013.

[120] Sebastian Moreno, Jennifer Neville, and Sergey Kirshner. Tied Kronecker Product Graph Models to Capture Variance in Network Populations. *ACM Transactions on Knowledge Discovery from Data*, 20(3):1–40, 3 2018.

[121] Eric Parsonage, Hung X. Nguyen, Rhys Bowden, Simon Knight, Nickolas Falkner, and Matthew Roughan. Generalized graph products for network design and analysis. *Proceedings - International Conference on Network Protocols, ICNP*, pages 79–88, 2011.

[122] D Chakrabarti, Y Zhan, and C Faloutsos. R-MAT: A Recursive Model for Graph Mining. In M W Berry, U Dayal, C Kamath, and D B Skillicorn, editors, *Fourth SIAM International Conference on Data Mining*, pages 442–446, 2004.

[123] Rohan Sharma and Bibhas Adhikari. Self-Coordinated Corona Graphs: a model for complex networks. *arXiv preprint*, pages 1–21, 9 2015.

[124] Rohan Sharma, Bibhas Adhikari, and Abhishek Mishra. Structural and spectral properties of corona graphs. *Discrete Applied Mathematics*, 228:14–31, 2017.

[125] S Moreno, S Kirshner, J Neville, and S V N Vishwanathan. Tied Kronecker product graph models to capture variance in network populations. In *48th Annual Allerton Conference on Communication, Control, and Computing*, pages 1137–1144, 2010.

[126] C. Seshadhri, Ali Pinar, and Tamara G Kolda. An in-depth analysis of stochastic Kronecker graphs. *Journal of the ACM*, 60(2):1–32, 4 2013.

[127] Tamara G. Kolda, Ali Pinar, Todd Plantenga, and C Seshadhri. A Scalable Generative Graph Model with Community Structure. *SIAM Journal on Scientific Computing*, 36(5):C424–C452, 1 2014.

[128] A Sala, L Cao, C Wilson, R Zablit, H Zheng, and B Y Zhao. Measurement-calibrated Graph Models for Social Network Experiments. In *19th International Conference on World Wide Web*, pages 861–870, 2010.

[129] M Barthélemy. Spatial networks. *Physics Reports*, 499(1-3):1–101, 2011.

[130] Avanti Athreya, Donniell E. Fishkind, Keith Levin, Vince Lyzinski, Youngser Park, Yichen Qin, Daniel L. Sussman, Minh Tang, Joshua T. Vogelstein, and Carey E. Priebe. Statistical Inference on Random Dot Product Graphs: A Survey. *J. Mach. Learn. Res.*, 18(January 2017):8393–8484, 9 2017.

[131] Stephen J. Young and Edward R. Scheinerman. Random Dot Product Graph Models for Social Networks. In *Algorithms and Models for the Web-Graph*, volume 4863, pages 138–149. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[132] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marin Boguñá. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 9 2010.

[133] Ginestra Bianconi and Christoph Rahmede. Emergent Hyperbolic Network Geometry. *Scientific Reports*, 7(October 2016):1–9, 2017.

[134] Alessandro Muscoloni, Josephine Maria Thomas, Sara Ciucci, Ginestra Bianconi, and Carlo Vittorio Cannistraci. Machine learning meets complex networks via coalescent embedding in the hyperbolic space. *Nature Communications*, 8(1):1615, 12 2017.

[135] Alessandro Muscoloni and Carlo Vittorio Cannistraci. Minimum curvilinear automata with similarity attachment for network embedding and link prediction in the hyperbolic space. *arXiv preprint*, 2 2018.

[136] Mason A. Porter, Jukka-Pekka Onnela, and Peter J. Mucha. Communities in Networks. *Physics Reports*, 486(3-5):75–174, 2 2009.

[137] S Fortunato and C Castellano. Community Structure in Graphs. In R A Meyers, editor, *Encyclopedia of Complexity and Systems Science*, pages 1141–1163. Springer, 2009.

[138] Peter Csermely, Andrs London, L.-Y. Wu, and Brian Uzzi. Structure and dynamics of core/periphery networks. *Journal of Complex Networks*, 1(2):93–123, 12 2013.

[139] M Puck Rombach, Mason A Porter, James H Fowler, and Peter J Mucha. Core-Periphery Structure in Networks. *SIAM Journal on Applied Mathematics*, 74(1):167–190, 1 2014.

[140] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, 11 2016.

[141] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[142] Stanley Wasserman and Carolyn Anderson. Stochastic a posteriori blockmodels: Construction and assessment. *Social Networks*, 9(1):1–36, 1987.

[143] Katherine Faust and Stanley Wasserman. Blockmodels: Interpretation and evaluation. *Social Networks*, 14(1-2):5–61, 1992.

[144] Brian Karrer and M E J Newman. Stochastic blockmodels and community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 83(1):1–11, 2011.

[145] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed Membership Stochastic Blockmodels. *Jmlr*, 9(2008):1981–2014, 2008.

[146] Emmanuel Abbe. Community Detection and Stochastic Block Models. *Foundations and Trends® in Communications and Information Theory*, 14(1-2):1–162, 3 2018.

[147] R Lukeman, Y.-X. Li, and L Edelstein-Keshet. Inferring individual rules from collective behavior. *Proceedings of the National Academy of Sciences*, 107(28):12576–12580, 2010.

[148] Yoshihiko Kayama. Complex networks derived from cellular automata. *arXiv*, 9 2010.

[149] X S Yang and Y Z L Yang. Cellular automata networks. In *Proceedings of Unconventional Computing*, pages 280–302, 2007.

[150] A Bailey, M Ventresca, and B Ombuki-Berman. Automatic Generation of Graph Models for Complex Networks by Genetic Programming. In *14th Annual Conference on Genetic and Evolutionary Computation*, pages 711–718, 2012.

[151] A Bigdeli, A Tizghadam, and A Leon-Garcia. Comparison of Network Criticality, Algebraic Connectivity, and Other Graph Metrics. In *1st Annual Workshop on Simplifying Complex Network for Practitioners*, pages 4:1–4:6. ACM, 2009.

[152] Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42–65, 8 1982.

[153] Damien Fay, Andrew W. Moore, Ken Brown, Michele Filosi, and Giuseppe Jurman. Graph metrics as summary statistics for Approximate Bayesian Computation with application to network model parameter estimation. *Journal of Complex Networks*, 3(1):52–83, 3 2015.

[154] J. Goni, Martijn P. van den Heuvel, Andrea Avena-Koenigsberger, Nieves Velez de Mendizabal, Richard F. Betzel, Alessandra Griffa, Patric Hagmann, Bernat Corominas-Murtra, J.-P. Thiran, and Olaf Sporns. Resting-brain functional connectivity predicted by analytic measures of network communication. *Proceedings of the National Academy of Sciences*, 111(2):833–838, 1 2014.

[155] K R Harrison, M Ventresca, and B Ombuki-Berman. Investigating Fitness Measures for the Automatic Construction of Graph Models. In A M Mora and G Squillero, editors, *EvoApplications*, volume 9028 of *Lecture Notes in Computer Science*, pages 189–200. Springer, 2015.

[156] Sadegh Aliakbary, Sadegh Motallebi, Sina Rashidian, Jafar Habibi, and Ali Movaghar. Distance metric learning for complex networks: Towards size-independent comparison of network structures. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(2):023111, 2 2015.

[157] Sucheta Soundarajan, Tina Eliassi-Rad, and Brian Gallagher. A Guide to Selecting a Network Similarity Method. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, number 1, pages 1037–1045, Philadelphia, PA, 4 2014. Society for Industrial and Applied Mathematics.

[158] Frank Emmert-Streib, Matthias Dehmer, and Yongtang Shi. Fifty years of graph matching, network alignment and network comparison. *Information Sciences*, 346-347:180–197, 6 2016.

[159] Claire Donnat and Susan Holmes. Tracking network dynamics: a survey of distances and similarity metrics. *arXiv preprint*, 1 2018.

[160] A. Avena-Koenigsberger, J. Goni, R. Sole, and O. Sporns. Network morphospace. *Journal of The Royal Society Interface*, 12(103):20140881–20140881, 12 2014.

[161] Viplove Arora and Mario Ventresca. Evaluating the Natural Variability in Generative Models for Complex Networks. In *Studies in Computational Intelligence*, pages 743–754. Springer International Publishing, 2019.

[162] Niousha Attar and Sadegh Aliakbary. Classification of complex networks based on similarity of topological network features. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(9):091102, 9 2017.

[163] Kyle Robert Harrison, Mario Ventresca, and Beatrice M Ombuki-Berman. A meta-analysis of centrality measures for comparing and generating complex network models. *Journal of Computational Science*, 17:205–215, 11 2016.

[164] Michele Berlingerio, Danai Koutra, Tina Eliassi-Rad, and Christos Faloutsos. NetSimile: A Scalable Approach to Size-Independent Network Similarity. *arXiv preprint*, 9 2012.

[165] James P. Bagrow and Erik M. Bollt. An information-theoretic, all-scales approach to comparing networks. *Applied Network Science*, 4(1):45, 12 2019.

[166] Manlio De Domenico and Jacob Biamonte. Spectral Entropies as Information-Theoretic Tools for Complex Network Comparison. *Physical Review X*, 6(4):041062, 12 2016.

[167] Yutaka Shimada, Yoshito Hirata, Tohru Ikeguchi, and Kazuyuki Aihara. Graph distance for complex networks. *Scientific Reports*, 6(1):34944, 12 2016.

[168] J Shore and B Lubin. Spectral goodness of fit for network models. *Social Networks*, 43(0):16–27, 2015.

[169] N Przulj, D. G. Corneil, and I. Jurisica. Modeling interactome: scale-free or geometric? *Bioinformatics*, 20(18):3508–3515, 12 2004.

[170] W Hayes, K Sun, and Nataa Pržulj. Graphlet-based measures are suitable for biological network comparison. *Bioinformatics*, 29(4):483–491, 2013.

[171] Waqar Ali, Anatol E. Wegner, Robert E. Gaunt, Charlotte M. Deane, and Gesine Reinert. Comparison of large networks with sub-sampling strategies. *Scientific Reports*, 6(June):24–29, 2016.

[172] X Gao, B Xiao, D Tao, and X Li. A Survey of Graph Edit Distance. *Pattern Analysis and Applications*, 13(1):113–129, 2010.

[173] Dena Asta and Cosma Rohilla Shalizi. Geometric Network Comparison. *arXiv preprint*, 11 2014.

[174] S Moreno and J Neville. Network Hypothesis Testing Using Mixed Kronecker Product Graph Models. In H Xiong, G Karypis, B M Thuraisingham, D J Cook, and X Wu, editors, *13th International Conference on Data Mining*, pages 1163–1168, 2013.

[175] Cedric E. Ginestet, Jun Li, Prakash Balachandran, Steven Rosenberg, and Eric D. Kolaczyk. Hypothesis testing for network data in functional neuroimaging. *The Annals of Applied Statistics*, 11(2):725–750, 6 2017.

[176] Petter Holme and Fredrik Liljeros. Mechanistic models in computational social science. *Frontiers in Physics*, 3:1–14, 9 2015.

[177] Ruth E. Baker, Jose Maria Peña, Jayaratnam Jayamohan, and Antoine Jérusalem. Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biology Letters*, 14(5):1–4, 2018.

[178] Peter Machamer, Lindley Darden, and Carl F Craver. Thinking about Mechanisms. *Philosophy of Science*, 67(1):1–25, 3 2000.

[179] Benjamin M Bolker. *Ecological models and data in R*. Princeton University Press, 2008.

[180] Carl F. Craver. When mechanistic models explain. *Synthese*, 153(3):355–376, 11 2006.

[181] Viplove Arora and Mario Ventresca. A Multi-objective Optimization Approach for Generating Complex Networks. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion - GECCO '16 Companion*, pages 15–16, New York, New York, USA, 2016. ACM Press.

[182] HARRY CRANE and HENRY TOWSNER. RELATIVELY EXCHANGEABLE STRUCTURES. *The Journal of Symbolic Logic*, 83(2):416–442, 6 2018.

[183] Harry Crane and Walter Dempsey. Edge Exchangeable Models for Interaction Networks. *Journal of the American Statistical Association*, 113(523):1311–1326, 7 2018.

[184] Alexander P. Kartun-Giles, Dmitri Krioukov, James P. Gleeson, Yamir Moreno, and Ginestra Bianconi. Sparse power-law network model for reliable statistical predictions based on sampled data. *Entropy*, 20(4):1–17, 2018.

[185] Olav Kallenberg. *Foundations of modern probability*. Springer Science \& Business Media, 2006.

[186] Harry Crane. *Probabilistic foundations of statistical network analysis*. Chapman and Hall/CRC, 2018.

[187] Peter Orbanz and Daniel M Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):437–461, 2015.

[188] Bruno De Finetti. La prévision: ses lois logiques, ses sources subjectives. In *Annales de l'institut Henri Poincaré*, volume 7, pages 1–68, 1937.

[189] Douglas N Hoover. Relations on probability spaces and arrays of random variables. *Preprint, Institute for Advanced Study, Princeton, NJ*, 2, 1979.

[190] David J Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.

[191] Olav Kallenberg. Symmetries on random arrays and set-indexed processes. *Journal of Theoretical Probability*, 5(4):727–765, 1992.

[192] Persi Diaconis, Susan Holmes, and Svante Janson. Threshold Graph Limits and Random Threshold Graphs. *Internet Mathematics*, 5(3):267–320, 1 2008.

[193] David J Aldous. More uses of exchangeability: representations of complex random structures. In N. H. Bingham and C. M. Goldie, editors, *Probability and Mathematical Genetics*, pages 35–63. Cambridge University Press, Cambridge, 2010.

[194] Harry Crane and Walter Dempsey. Relational exchangeability. *Journal of Applied Probability*, 56(01):192–208, 3 2019.

[195] Franois Caron and Emily B. Fox. Sparse graphs using exchangeable random measures. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 79(5):1–44, 1 2017.

[196] Harry Crane and Walter Dempsey. Edge Exchangeable Models for Interaction Networks. *Journal of the American Statistical Association*, 113(523):1311–1326, 2018.

[197] Lszl Lovász. *Large networks and graph limits*, volume 60. American Mathematical Soc., 2012.

[198] Peter J Bickel and Aiyou Chen. A nonparametric view of network models and Newman-Girvan and other modularities. *Proceedings of the National Academy of Sciences of the United States of America*, 106(50):21068–21073, 2009.

[199] Christian Borgs, Jennifer T. Chayes, Henry Cohn, and Nina Holden. Sparse exchangeable graphs and their limits via graphon processes. *J. Mach. Learn. Res.*, 18(1):1–71, 1 2017.

[200] Diana Cai, Trevor Campbell, and Tamara Broderick. Edge-exchangeable graphs and sparsity. In *Advances in Neural Information Processing Systems 29*, pages 4249–4257, 3 2016.

[201] Sinead A Williamson. Nonparametric Network Models for Link Prediction. *Journal of Machine Learning Research*, 17:1–21, 2016.

[202] Harry Crane and Henry Towsner. Relative exchangeability with equivalence relations. *Archive for Mathematical Logic*, 57(5-6):533–556, 8 2018.

[203] Stephen Wolfram. *A new kind of science*. Wolfram Media, Champaign, IL, 2002.

[204] Melanie Mitchell. *Complexity: A guided tour*. Oxford University Press, 2009.

[205] B Zheng, H Wu, L Kuang, J Qin, W Du, J Wang, and D Li. A simple model clarifies the complicated relationships of complex networks. *Sci Rep*, 4:6197, 2014.

[206] Viplove Arora and Mario Ventresca. The inverse problem of discovering complex network generators. In *Proceedings of the 9th International Conference on Inverse Problems in Engineering (ICIPE). Journal of Physics*, 2017.

[207] Albert-Laszlo Barabasi. Network science: Luck or reason. *Nature*, 489(7417):507–508, 9 2012.

[208] Susan R. Hunter, Eric A. Applegate, Viplove Arora, Bryan Chong, Kyle Cooper, Oscar Rincón-Guevara, and Carolina Vivas-Valencia. An Introduction to Multiobjective Simulation Optimization. *ACM Transactions on Modeling and Computer Simulation*, 29(1):1–36, 1 2019.

[209] Piotr Czyzak and Andrzej Jaszkiewicz. Pareto Simulated Annealing–A Metaheuristic Technique for Multiple-Objective Combinatorial Optimization. *Journal of Multi-Criteria Decision Analysis*, 7(1):34–47, 1998.

[210] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[211] Jean Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

[212] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph Evolution: Densification and Shrinking Diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1):2–es, 3 2007.

[213] Viplove Arora and Mario Ventresca. Action-Based Model for Topologically Resilient Supply Networks. In *Studies in Computational Intelligence*, volume 689, pages 658–669, 2018.

[214] Viplove Arora and Mario Ventresca. Modeling topologically resilient supply chain networks. *Applied Network Science*, 3(1):19, 12 2018.

[215] Alexandra Brintrup, Anna Ledwoch, and Jose Barros. Topological robustness of the global automotive industry. *Logistics Research*, 9(1):1, 12 2016.

[216] Alexandra Brintrup, Yu Wang, and Ashutosh Tiwari. Supply Networks as Complex Systems: A Network-Science-Based Characterization. *IEEE Systems Journal*, PP(99):1–12, 2015.

[217] Marcus A. Bellamy and Rahul C. Basole. Network analysis of supply chain systems: A systematic review and future research. *Systems Engineering*, 16(2):235–249, 6 2013.

[218] Thomas Y Choi, Kevin J Dooley, and Manus Rungtusanatham. Supply networks and complex adaptive systems: control versus emergence. *Journal of Operations Management*, 19(3):351–366, 5 2001.

[219] Amit Surana, Soundar Kumara, Mark Greaves, and Usha Nandini Raghavan. Supply-chain networks: a complex adaptive systems perspective. *International Journal of Production Research*, 43(20):4235–4265, 10 2005.

[220] Surya D. Pathak, David M. Dilts, and Gautam Biswas. On the evolutionary dynamics of supply network topologies. *IEEE Transactions on Engineering Management*, 54(4):662–672, 2007.

[221] Yusoon Kim, Thomas Y Choi, Tingting Yan, and Kevin Dooley. Structural investigation of supply networks: A social network analysis approach. *Journal of Operations Management*, 29(3):194–211, 2011.

[222] Stephen P. Borgatti and Xun Li. On Social Network Analysis In A Supply Chain Context. *Journal of Supply Chain Management*, 45(2):5–22, 4 2009.

[223] Alessandro Lomi and Philippa Pattison. Manufacturing Relations: An Empirical Study of the Organization of Production Across Multiple Networks. *Organization Science*, 17(3):313–332, 6 2006.

[224] Wang Keqiang Wang Keqiang, Zeng Zhaofeng Zeng Zhaofeng, and Sun Dongchuan Sun Dongchuan. Structure Analysis of Supply Chain Networks Based on Complex Network Theory. *2008 Fourth International Conference on Semantics, Knowledge and Grid*, pages 493–494, 12 2008.

[225] Supun S Perera, Michael G H Bell, and Michiel C J Bliemer. Modelling Supply Chains as Complex Networks for Investigating Resilience : An Improved Methodological Framework. *Australasian Transport Research Forum (ATRF)*, pages 1–16, 2013.

[226] Patrick Alcantara, Gianluca Riglietti, and Lucila Aguada. BCI Supply Chain Resilience Report. Technical report, Business Continuity Institute, 2017.

[227] Walid Klibi, Alain Martel, and Adel Guitouni. The design of robust value-creating supply chain networks: A critical review. *European Journal of Operational Research*, 203(2):283–293, 6 2010.

[228] H.P. Thadakamalla, U.N. Raghavan, Soundar Kumara, and A. Albert. Survivability of Multiagent-Based Supply Networks: A Topological Perspective. *IEEE Intelligent Systems*, 19(5):24–31, 9 2004.

[229] Kang Zhao, Akhil Kumar, Terry P. Harrison, and John Yen. Analyzing the Resilience of Complex Supply Network Topologies Against Random and Targeted Disruptions. *IEEE Systems Journal*, 5(1):28–39, 3 2011.

[230] Wenjun Wang, W. Nick Street, and Renato E. DeMatta. Topological Resilience Analysis of Supply Networks under Random Disruptions and Targeted Attacks. *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM '15*, pages 250–257, 2015.

[231] A.P. Barroso, V.H. Machado, H. Carvalho, and V. Cruz Machado. Quantifying the Supply Chain Resilience. In *Applications of Contemporary Management Approaches in Supply Chains*, pages 13–32. InTech, 4 2015.

[232] Jianxi Gao, Baruch Barzel, and Albert-Lszl Barabási. Universal resilience patterns in complex networks. *Nature*, 530(7590):307–312, 2 2016.

[233] Anand Nair and Jose M. Vidal. Supply network topology and robustness against disruptions - an investigation using multi-agent model. *International Journal of Production Research*, 49(5):1391–1404, 2011.

[234] Lei Wen, MingFang Guo, and LiJun Wang. Statistic Characteristics Analysis of Directed Supply Chain Complex Network. *International Journal of Advancements in Computing Technology*, 4(21):84–91, 11 2012.

[235] Sonia Irshad Mari, Young Hae Lee, Muhammad Saad Memon, Young Soo Park, and Minsun Kim. Adaptivity of Complex Network Topologies for Designing Resilient Supply Chain Networks. *The International Journal of Industrial Engineering: Theory, Applications and Practice*, 22:102–116, 2015.

[236] Gang Li, Yong Gen Gu, and Zhi Huan Song. Evolution of cooperation on heterogeneous supply networks. *International Journal of Production Research*, 51(13):3894–3902, 2013.

[237] Qi Xuan, Fang Du, Yanjun Li, and Tie-jun Wu. A Framework to Model the Topological Structure of Supply Networks. *IEEE Transactions on Automation Science and Engineering*, 8(2):442–446, 4 2011.

[238] Supun Perera, H. Niles Perera, and Dharshana Kasthurirathna. Value chain approach for modelling resilience of tiered supply chain networks. In *2017 Moratuwa Engineering Research Conference (MERCon)*, pages 159–164, 2017.

[239] Supun Perera, H Niles Perera, and Dharshana Kasthurirathna. Structural characteristics of complex supply chain networks. In *2017 Moratuwa Engineering Research Conference (MERCon)*, pages 135–140. IEEE, 5 2017.

[240] Supun Perera, Michael G.H. Bell, and Michiel C.J. Bliemer. Network science approach to modelling the topology and robustness of supply chain networks: a review and perspective. *Applied Network Science*, 2(1):33, 12 2017.

[241] D L Alderson. Catching the Network Science Bug: Insight and Opportunities for the Operations Researchers. *Operations Research*, 56(5):1047–1065, 2009.

[242] Surya D. Pathak, David M. Dilts, and Sankaran Mahadevan. Investigating population and topological evolution in a complex adaptive supply network. *Journal of Supply Chain Management*, 45(3):54–67, 2009.

[243] Edward J.S. Hearnshaw and Mark M.J. Wilson. A complex network approach to supply chain network theory. *International Journal of Operations & Production Management*, 33(4):442–469, 3 2013.

[244] Yusoon Kim, Yi Su Chen, and Kevin Linderman. Supply network disruption and resilience: A network structural perspective. *Journal of Operations Management*, 33-34(September 2015):43–59, 2015.

[245] Vipul Jain, S. Wadhwa, and S. G. Deshmukh. Select supplier-related issues in modelling a dynamic supply chain: Potential, challenges and direction for future research. *International Journal of Production Research*, 47(11):3013–3039, 2009.

[246] Michael Bell, Supun Perera, Mahendrarajah Piraveenan, Michiel Bliemer, Tanya Latty, and Chris Reid. Network growth models: A behavioural basis for attachment proportional to fitness. *Scientific Reports*, 7(October 2016):42431, 2 2017.

[247] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage*, 52(3):1059–1069, 9 2010.

[248] L. da F. Costa, Francisco A. Rodrigues, Gonzalo Travieso, and P. R. Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242, 1 2007.

[249] Kang Zhao, Akhil Kumar, and John Yen. Achieving High Robustness in Supply Distribution Networks by Rewiring. *IEEE Transactions on Engineering Management*, 58(2):347–362, 5 2011.

[250] G. Li, P. Ji, L.Y. Sun, and W.B. Lee. Modeling and simulation of supply network evolution based on complex adaptive system and fitness landscape. *Computers & Industrial Engineering*, 56(3):839–853, 4 2009.

[251] Gang Li, Hongjiao Yang, Linyan Sun, Ping Ji, and Lei Feng. The evolutionary complexity of complex adaptive supply networks: A simulation and case study. *International Journal of Production Economics*, 124(2):310–330, 2010.

[252] P. E. Vertes, A. F. Alexander-Bloch, N. Gogtay, J. N. Giedd, J. L. Rapoport, and E. T. Bullmore. Simple models of human brain functional networks. *Proceedings of the National Academy of Sciences*, 109(15):5868–5873, 4 2012.

[253] Richard F Betzel, Andrea Avena-Koenigsberger, Joaqun Goñi, Ye He, Marcel A. de Reus, Alessandra Griffa, Petra E Vértes, Bratislav Mišic, Jean-philippe Thiran, Patric Hagmann, Martijn van den Heuvel, Xi-nian Zuo, Edward T Bullmore, and Olaf Sporns. Generative models of the human connectome. *NeuroImage*, 124:1054–1064, 1 2016.

[254] Olaf Sporns, Giulio Tononi, and Rolf Kötter. The human connectome: A structural description of the human brain. *PLoS Computational Biology*, 1(4):0245–0251, 2005.

[255] O Sporns. *Networks of the Brain*. MIT Press, 2010.

[256] Santiago Ramn Cajal. *Histology of the nervous system of man and vertebrates*, volume 1. Oxford University Press, USA, 1995.

[257] Larry W Swanson. *Brain architecture: understanding the basic plan*. Oxford University Press, 2012.

[258] Danielle S Bassett, Perry Zurn, and Joshua I Gold. On the nature and use of models in network neuroscience. *Nature Reviews Neuroscience*, pages 1–13, 7 2018.

[259] Olaf Sporns. The human connectome: a complex network. *Annals of the New York Academy of Sciences*, 1224(1):109–125, 4 2011.

[260] Qawi K. Telesford, Sean L. Simpson, Jonathan H. Burdette, Satoru Hayasaka, and Paul J. Laurienti. The Brain as a Complex System: Using Network Science as a Tool for Understanding the Brain. *Brain Connectivity*, 1(4):295–308, 2011.

[261] Alex Fornito, Andrew Zalesky, and Michael Breakspear. Graph analysis of the human connectome: Promise, progress, and pitfalls. *NeuroImage*, 80:426–444, 10 2013.

[262] Olaf Sporns. Contributions and challenges for network models in cognitive neuroscience. *Nature Neuroscience*, 17(5):652–660, 2014.

[263] Olaf Sporns. Graph theory methods: applications in brain networks. *Dialogues in Clinical Neuroscience*, 20:111–120, 2018.

[264] Danielle S Bassett and Olaf Sporns. Network neuroscience. *Nature Neuroscience*, 20(3):353–364, 3 2017.

[265] Richard F. Betzel and Danielle S. Bassett. Generative models for network neuroscience: prospects and promise. *Journal of The Royal Society Interface*, 14(136):20170623, 11 2017.

[266] Bernadette C. M. van Wijk, Cornelis J. Stam, and Andreas Daffertshofer. Comparing Brain Networks of Different Size and Connectivity Density Using Graph Theory. *PLoS ONE*, 5(10):e13701, 10 2010.

[267] Florian Klimm, Danielle S Bassett, Jean M Carlson, and Peter J Mucha. Resolving Structural Variability in Network Models and the Brain. *PLoS Computational Biology*, 10(3):e1003491, 3 2014.

[268] Sean L Simpson, Malaak N Moussa, and Paul J Laurienti. An exponential random graph modeling approach to creating group-based representative whole-brain connectivity networks. *NeuroImage*, 60(2):1117–1126, 4 2012.

[269] David C. Van Essen, Stephen M. Smith, Deanna M. Barch, Timothy E.J. Behrens, Essa Yacoub, and Kamil Ugurbil. The WU-Minn Human Connectome Project: An overview. *NeuroImage*, 80:62–79, 10 2013.

[270] Evan M. Gordon, Timothy O. Laumann, Babatunde Adeyemo, and Steven E. Petersen. Individual Variability of the System-Level Organization of the Human Brain. *Cerebral Cortex*, 27(1):386–399, 10 2015.

[271] R. Edward Roberts, Elaine J. Anderson, and Masud Husain. White Matter Microstructure and Cognitive Function. *The Neuroscientist*, 19(1):8–15, 2 2013.

[272] J A Peacock. Two-dimensional goodness-of-fit testing in astronomy. *Monthly Notices of the Royal Astronomical Society*, 202(3):615–627, 1983.

[273] Max Hinne, Tom Heskes, Christian F. Beckmann, and Marcel A.J. van Gerven. Bayesian inference of structural brain networks. *NeuroImage*, 66:543–552, 2013.

[274] John D. Medaglia, Mary-Ellen Lynall, and Danielle S. Bassett. Cognitive Network Neuroscience. *Journal of Cognitive Neuroscience*, 27(8):1471–1491, 8 2015.

[275] Aron K. Barbey. Network Neuroscience Theory of Human Intelligence. *Trends in Cognitive Sciences*, 22(1):8–20, 2018.

[276] Douglas H. Schultz and Michael W. Cole. Higher Intelligence Is Associated with Less Task-Related Brain Network Reconfiguration. *The Journal of Neuroscience*, 36(33):8551–8561, 8 2016.

[277] M. P. van den Heuvel, C. J. Stam, R. S. Kahn, and H. E. Hulshoff Pol. Efficiency of Functional Brain Networks and Intellectual Performance. *Journal of Neuroscience*, 29(23):7619–7624, 6 2009.

[278] Yonghui Li, Yong Liu, Jun Li, Wen Qin, Kuncheng Li, Chunshui Yu, and Tianzi Jiang. Brain Anatomical Network and Intelligence. *PLoS Computational Biology*, 5(5):e1000395, 5 2009.

[279] Hae Jeong Park and Karl Friston. Structural and functional brain networks: From connections to cognition. *Science*, 342(6158), 2013.

[280] S L Simpson, S Hayasaka, and P J Laurienti. Exponential random graph modeling for complex brain networks. *PLoS ONE*, 6(5):e20039, 2011.

[281] Richard F. Betzel, John D. Medaglia, and Danielle S. Bassett. Diversity of meso-scale architecture in human and non-human connectomes. *Nature Communications*, 9(1):346, 12 2018.

[282] Sebastian Moreno and Jennifer Neville. An Investigation of the Distributional Characteristics of Generative Graph Models. In *Proceedings of the 1st Workshop on Information in Networks*, 2009.

[283] Frank Emmert-Streib and Matthias Dehmer. Exploring Statistical and Population Aspects of Network Complexity. *PLoS ONE*, 7(5):e34523, 5 2012.

[284] Matthias Dehmer and Abbe Mowshowitz. A history of graph entropy measures. *Information Sciences*, 181(1):57–78, 2011.

[285] Kartik Anand and Ginestra Bianconi. Entropy measures for networks: Toward an information theory of complex topologies. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 80(4):1–4, 2009.

[286] Ginestra Bianconi. The entropy of randomized network ensembles. *EPL (Europhysics Letters)*, 81(2):28005, 1 2008.

[287] Tiago P Peixoto. Entropy of stochastic blockmodel ensembles. *Physical Review E*, 85(5):056122, 5 2012.

[288] Isabella Gollini and Thomas Brendan Murphy. Joint Modeling of Multiple Network Views. *Journal of Computational and Graphical Statistics*, 25(1):246–265, 2016.

[289] Daniele Durante, David B. Dunson, and Joshua T. Vogelstein. Nonparametric Bayes Modeling of Populations of Networks. *Journal of the American Statistical Association*, 112(520):1516–1530, 2017.

[290] Simn Lunagómez, Sofia C. Olhede, and Patrick J. Wolfe. Modeling Network Populations via Graph Distances. *arXiv preprint*, pages 1–26, 2019.

[291] Tracy M. Sweet, Andrew C. Thomas, and Brian W. Junker. Hierarchical Network Models for Education Research. *Journal of Educational and Behavioral Statistics*, 38(3):295–318, 2012.

[292] Tiago P. Peixoto. Model selection and hypothesis testing for large-scale network models with overlapping groups. *Physical Review X*, 5(1):1–20, 2015.

[293] Giona Casiraghi, Vahan Nanumyan, Ingo Scholtes, and Frank Schweitzer. Generalized Hypergeometric Ensembles: Statistical Hypothesis Testing in Complex Networks. *arXiv*, pages 1–5, 7 2016.

[294] Tom A B Snijders, Philippa E Pattison, Garry L Robins, and Mark S Handcock. New specifications for exponential random graph models. *Sociological Methodology*, page 44, 2004.

[295] David R. Hunter. Curved exponential family models for social networks. *Social Networks*, 29(2):216–230, 5 2007.

[296] M. Piraveenan, M. Prokopenko, and A. Y. Zomaya. Local assortativeness in scale-free networks. *EPL (Europhysics Letters)*, 84(2):28002, 10 2008.

[297] M. Piraveenan, M. Prokopenko, and A. Y. Zomaya. ADDENDUM: Local assortativeness in scale-free networks. *EPL (Europhysics Letters)*, 89(4):49901, 2 2010.

[298] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 1 2008.

[299] Leon Danon, Albert Díaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008–P09008, 9 2005.

[300] Tiago P. Peixoto. Nonparametric Bayesian inference of the microcanonical stochastic block model. *Physical Review E*, 95(1):1–21, 2017.

[301] Niousha Attar and Sadegh Aliakbary. Automatic Generation of Adaptive Network Models based on Similarity to the Desired Complex Network. *arXiv*, 10 2018.

[302] Jan Overgoor, Austin Benson, and Johan Ugander. Choosing to Grow a Graph: Modeling Network Formation as Discrete Choice. In *The World Wide Web Conference on - WWW '19*, pages 1409–1420, New York, New York, USA, 11 2019. ACM Press.

[303] M. E. J. Newman. Network structure from rich but noisy data. *Nature Physics*, 14(6):542–545, 6 2018.

[304] M E J Newman. Estimating network structure from unreliable measurements. *Physical Review E*, 98(6):062321, 12 2018.

[305] Austin R. Benson, David F. Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.

[306] Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115(48):E11221–E11230, 11 2018.

[307] Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A Porter, Sergio Gómez, and Alex Arenas. Mathematical Formulation of Multilayer Networks. *Physical Review X*, 3(4):041022, 12 2013.

[308] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[309] S Boccaletti, G Bianconi, R Criado, C.I. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 11 2014.

[310] Mark E J Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):36104, 2006.

[311] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 US election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

[312] Donald Ervin Knuth. *The Stanford GraphBase: A platform for combinatorial computing*, volume 37. Addison-Wesley Reading, 2009.

[313] M Girvan and M E J Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[314] Pablo M Gleiser and Leon Danon. Community Structure in Jazz. *Advances in Complex Systems*, 6(4):565–573, 2003.

[315] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 9 2003.

[316] Helen M Berman, Tammy Battistuz, Talapady N Bhat, Wolfgang F Bluhm, Philip E Bourne, Kyle Burkhardt, Zukang Feng, Gary L Gilliland, Lisa Iype, Shri Jain, and others. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):899–907, 2002.

[317] Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Mikio Yoshida, Masahira Hattori, and Yoshiyuki Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences*, 98(8):4569–4574, 2001.

[318] C Stark, B.-J. Breitkreutz, A Chatr-aryamontri, L Boucher, R Oughtred, M S Livstone, J Nixon, K V Auken, X Wang, X Shi, T Reguly, J M Rust, A G Winter, K Dolinski, and M Tyers. The BioGRID Interaction Database: 2011 update. *Nucleic Acids Research*, 39(Database-Issue):698–704, 2011.

[319] Vittoria Colizza, Romualdo Pastor-Satorras, and Alessandro Vespignani. Reaction–diffusion processes and metapopulation models in heterogeneous networks. *Nature Physics*, 3(4):276, 2007.

[320] Cathrine Seierstad and Tore Opsahl. For the few not the many? The effects of affirmative action on presence, prominence, and social capital of women directors in Norway. *Scandinavian Journal of Management*, 27(1):44–54, 2011.

[321] Thomas Rolland, Murat Taşan, Benoit Charloteaux, Samuel J. Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, Atanas Kamburov, Susan D. Ghiassian, Xinping Yang, Lila Ghamsari, Dawit Balcha, Bridget E. Begg, Pascal Braun, Marc Brehme, Martin P. Broly, Anne-Ruxandra Carvunis, Dan Convery-Zupan, Roser Corominas, Jasmin Coulombe-Huntington, Elizabeth Dann, Matija Dreze, Amlie Dricot, Changyu Fan, Eric Franzosa, Fana Gebreab, Bryan J. Gutierrez, Madeleine F. Hardy, Mike Jin, Shuli Kang, Ruth Kiros, Guan Ning Lin, Katja Luck, Andrew MacWilliams, Jrg Menche, Ryan R. Murray, Alexandre Palagi, Matthew M. Poulin, Xavier Rambout, John Rasla, Patrick Reichert, Viviana Romero, Elien Ruyssinck, Julie M. Sahalie, Annemarie Scholz, Akash A. Shah, Amitabh Sharma, Yun Shen, Kerstin Spirohn, Stanley Tam, Alexander O. Tejeda, Shelly A. Trigg, Jean-Claude Twizere, Kerwin Vega, Jennifer Walsh, Michael E. Cusick, Yu Xia, Albert-Lszl Barabási, Lilia M. Iakoucheva, Patrick Aloy, Javier De Las Rivas, Jan Tavernier, Michael A. Calderwood, David E. Hill, Tong Hao, Frederick P. Roth, and Marc Vidal. A Proteome-Scale Map of the Human Interactome Network. *Cell*, 159(5):1212–1226, 11 2014.

[322] Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.

[323] Xilin Shen, Fuyuze Tokoglu, Xenios Papademetris, and R Todd Constable. Groupwise whole-brain parcellation from resting-state fMRI data for network node identification. *Neuroimage*, 82:403–415, 2013.

[324] Sean P Willems. Data Set Real-World Multiechelon Supply Chains Used for Inventory Optimization. *Manufacturing & Service Operations Management*, 10(1):19–23, 1 2008.

[325] Enrico Amico and J. Goni. Mapping hybrid functional-structural connectivity traits in the human connectome. *Network Neuroscience*, 2(3):306–322, 9 2018.

[326] Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-Franois Pinton, and Wouter den Broeck. What's in a crowd? Analysis of face-to-face behavioral networks. *Journal of theoretical biology*, 271(1):166–180, 2011.

[327] Abhijit Banerjee, Arun G Chandrasekhar, Esther Duflo, and Matthew O Jackson. The diffusion of microfinance. *Science*, 341(6144):1236498, 2013.

[328] Alireza Hajibagheri, Kiran Lakkaraju, Gita Sukthankar, Rolf T Wigand, and Nitin Agarwal. Conflict and communication in massively-multiplayer online games. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 65–74. Springer, 2015.

[329] Route Views. University of oregon route views project, 2000.

APPENDICES

APPENDIX A
EXTRAPOLATING THE ACTION-BASED MODEL TO LARGER NETWORKS

A.1 Introduction

For a given target network, the action-based model learns a model that can explain the formation of links between various nodes in the target network. It thus might be possible to use an action-based model optimized for a small target network for synthesizing larger networks. It is reasonable to assume that real-world networks arising in similar domains might be topologically similar and arise from similar interaction processes. Under this assumption, an action-based model learnt for a small Facebook network can be used to predict the topology of a much larger Facebook network. This way, we can scale the action-based model by training it on small target networks and using the learnt action matrices to extrapolate the structure to larger networks.

To test the ability of ABNG to extrapolate to larger networks, we decided to test on two real-world network datasets:

1. Five power networks listed in Table A.1, where the first network, 662-bus, is used as the target network and the learnt model is used for synthesizing the other four networks.
2. Sixteen Facebook networks listed in Table A.2, where the first four networks are separately used as the target networks and the learnt models are used for synthesizing the other twelve networks. This way we can also compare the relative performance of using different target networks.

We present some brief results for these experiments in the following section. The results are promising and warrant further investigations.

A.2 Results

The five small networks (one power and four Facebook) were used as the target networks, and ABNG was used for learning models using the procedure described in

Table A.1.
List of power grid networks along with some network properties: number of vertices n; number of edges m; mean degree z; average path length l; clustering coefficient c; and degree correlation coefficient r.

Network Name	n	m	z	l	c	r
662-bus	662	1568	4.737	10.2445	0.077	0.319
1138-bus	1138	1458	2.562	12.724	0.093	-0.080
bcspwr10	5300	8271	3.121	20.846	0.094	-0.053
eris1176	1176	8688	14.776	12.059	0.94	0.891
US-Grid	4941	6594	2.669	18.989	0.103	0.003

Table A.2.
List of Facebook networks along with some network properties: number of vertices n; number of edges m; mean degree z; average path length l; clustering coefficient c; and degree correlation coefficient r.

Network Name	n	m	z	l	c	r
Caltech36	769	16656	43.319	2.338	0.291	-0.065
Haverford76	1446	59589	82.419	2.228	0.251	0.067
Reed98	962	18812	39.11	2.461	0.221	0.023
Simmons81	1518	32988	43.46	2.57	0.212	-0.062
Brown11	8600	384526	89.425	2.696	0.145	0.069
Carnegie49	6637	249967	75.325	2.738	0.185	0.122
CMU	6621	249959	75.505	2.738	0.185	0.122
Mich67	3748	81903	43.705	2.839	0.194	0.142
MIT	6402	251230	78.485	2.72	0.180	0.120
Pepperdine86	3445	152007	88.248	2.50	0.206	0.056
Rice31	4087	184828	90.447	2.468	0.203	0.065
UC64	6833	155332	45.465	3.015	0.191	0.125
UChicago30	6591	208103	63.148	2.808	0.155	0.018
UMass92	16516	519385	62.895	2.934	0.123	-0.001
USFCA72	2682	65252	48.659	2.691	0.191	0.092
Williams40	2790	112986	80.994	2.416	0.207	0.040

Chapter 4. Degree distribution (DD), local assortativity (LA), and local transitivity (LT) were used as the three objectives in the optimization problem. As in previous Chapters, the action matrix closest to the origin was chosen as the representative solution for each of the networks and are shown in Table A.3. From the optimized action matrices we see that as in previous cases 'no action' gets a high probability for all the networks. Preferential attachment on neighbor degree is a dominant action for the power network. As expected, the social networks use actions based on similarity with high probability, but some preferential attachment actions are also used. The Simmons81 network produces a counter-intuitive result as the similarity-based actions have zero probability.

Table A.3.
The table shows optimized action matrices for the five different real-world networks that were extrapolated to larger networks. The following actions were used: Preferential attachment on - average neighbor degree (PAND), degree (PAD), PageRank (PAPR) and betweenness (PAB); Triadic closure (TC); Inverse log-weighted (SLW) and Jaccard similarities (SJ); and No action (NA).

Network↓ |Action→	PAND	PAD	PAPR	PAB	TC	SLW	SJ	NA	$\bar{P}$
662-bus	0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.31	0.98
	0.07	0.26	0.17	0.02	0.16	0.05	0.26	0.00	0.02
Caltech36	0.00	0.01	0.00	0.00	0.00	0.49	0.00	0.50	1.00
Haverford76	0.00	0.06	0.06	0.10	0.08	0.22	0.00	0.48	0.58
	0.16	0.41	0.08	0.07	0.01	0.20	0.07	0.01	0.42
Reed98	0.00	0.00	0.01	0.00	0.00	0.36	0.00	0.63	0.86
	0.00	0.00	0.25	0.25	0.00	0.01	0.46	0.03	0.14
Simmons81	0.00	0.03	0.04	0.34	0.00	0.00	0.00	0.60	1.00

Figures A.1–A.2d summarize the results of our extrapolation experiments. In each of the heat maps, the action matrices learnt for the target network were used as models for the larger networks, and the mean dissimilarity values (of 20 synthesized networks) of the optimization objectives are recorded. To provide a baseline, we also show the corresponding mean dissimilarity values for the target network and highlight it using a red box.

It is evident from the global network properties of the power networks listed in Table A.1 that the topologies are not very similar. Thus, we would expect that ABNG won't perform well in these extrapolation experiments, especially for the eris1176

Figure A.1. An action-based model learnt for the 662-bus network is used for synthesizing the rest of the power networks listed in Table A.1. The mean dissimilarity values of the optimization objectives are recorded in the heat maps for the original network (highlighted in red box) as well as the extrapolated networks. The lower the value, better is the ability to extrapolate.

network. Overall, it seems that ABNG does reasonably well for the power networks, and it would be interesting to compare the performance of the extrapolated networks with a model that is optimized for the same network.

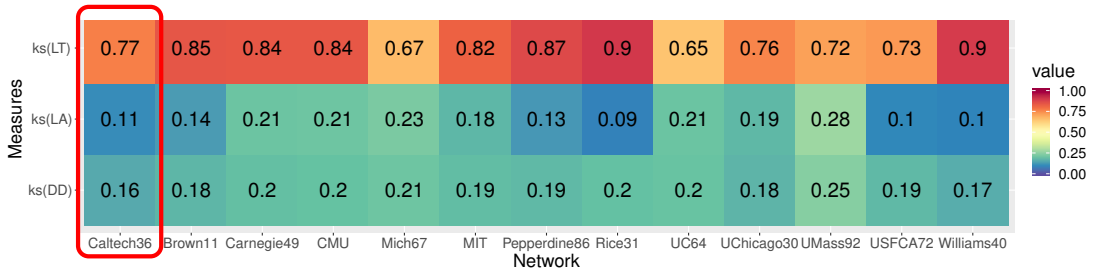
We now focus on the Facebook networks. The global network properties in Table A.2 show that most of the networks have similar clustering and degree correlation coefficients. It seems likely that ABNG should be able to extrapolate from smaller Facebook networks to larger ones. Our results in Figure A.2 show similar trends.

In all four cases, we see that if ABNG was able to reproduce network properties in the target network, it was also able to do so when the model was extrapolated to the larger networks. While these results seem promising, they need to be subjected to close examination on other measures. In Figures A.2a–A.2c, we see that ABNG is able to match the degree distribution and local assortativity for all the networks, but fails to perform well on local transitivity. On the contrary, the Simmons81 network performs better on local transitivity, and the effect is also seen on the results for the extrapolated networks. It might be useful to note that the Simmons81 network produced a radically different action matrix (see Table A.3), which might be a reason

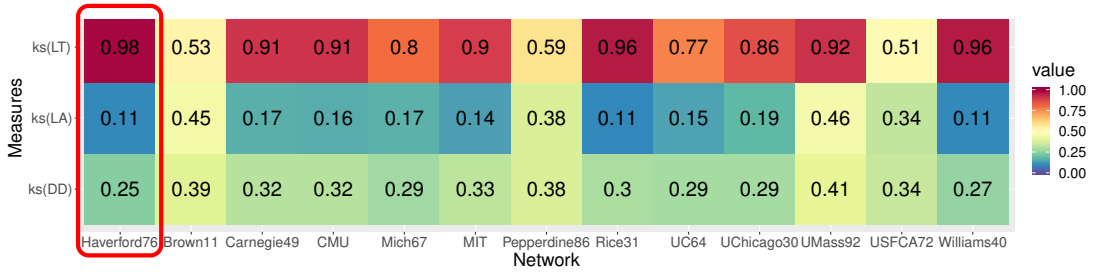
behind the different results for this particular network, but other possibilities need to be explored.

Finally, we also provide a 3D visualization of the Pareto optimal solutions for Simmons81 and two extrapolated networks, Williams40 and Carnegie49, in Figure A.3. 20 representative Pareto optimal solutions for Simmons81 were chosen and the action matrices were used to extrapolate to the two larger networks. Each solution is labelled using a unique number and the corresponding map in the objective space are shown. Using the plot we can get an idea of the spread of the Pareto optimal solutions, and how the corresponding solutions behave for the extrapolated networks.

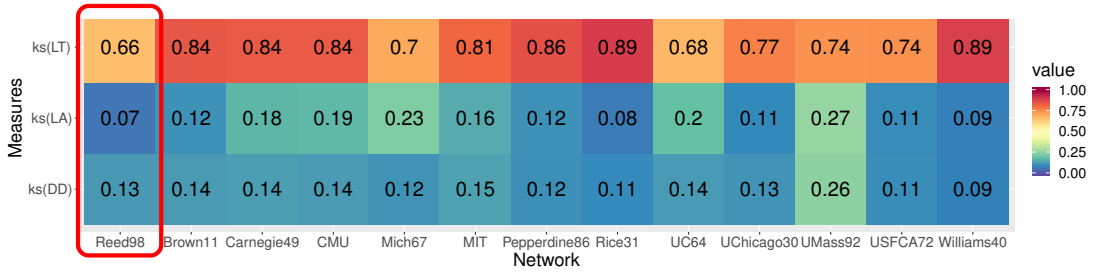
To conclude, our experimental results have highlighted that this is a promising application of ABNG. This paves the path for using the action-based model for very large networks by circumventing the problem of optimizing model parameters. Alternatively, one can use the solutions for a smaller network to find better solutions for larger networks.



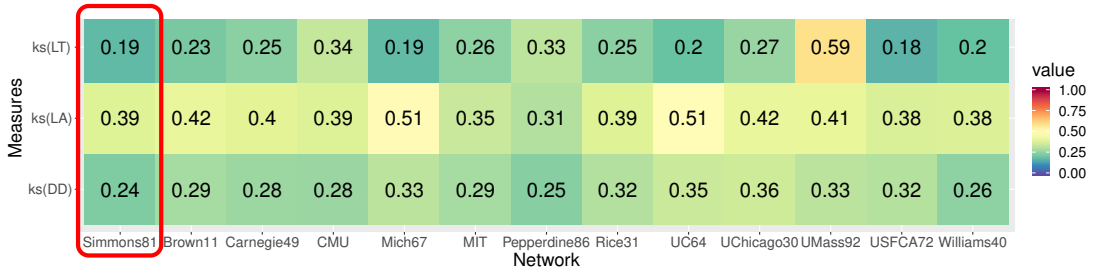
(a) Caltech36 social network used for extrapolation.



(b) Haverford76 social network used for extrapolation.



(c) Reed98 social network used for extrapolation.



(d) Simmons81 social network used for extrapolation.

Figure A.2. An action-based model is learnt for the four Facebook networks are used for synthesizing the larger Facebook networks listed in Table A.2. The mean dissimilarity values of the optimization objectives are recorded in the heat maps for the original network (highlighted in red box) as well as the extrapolated networks. The lower the value, better is the ability to extrapolate.

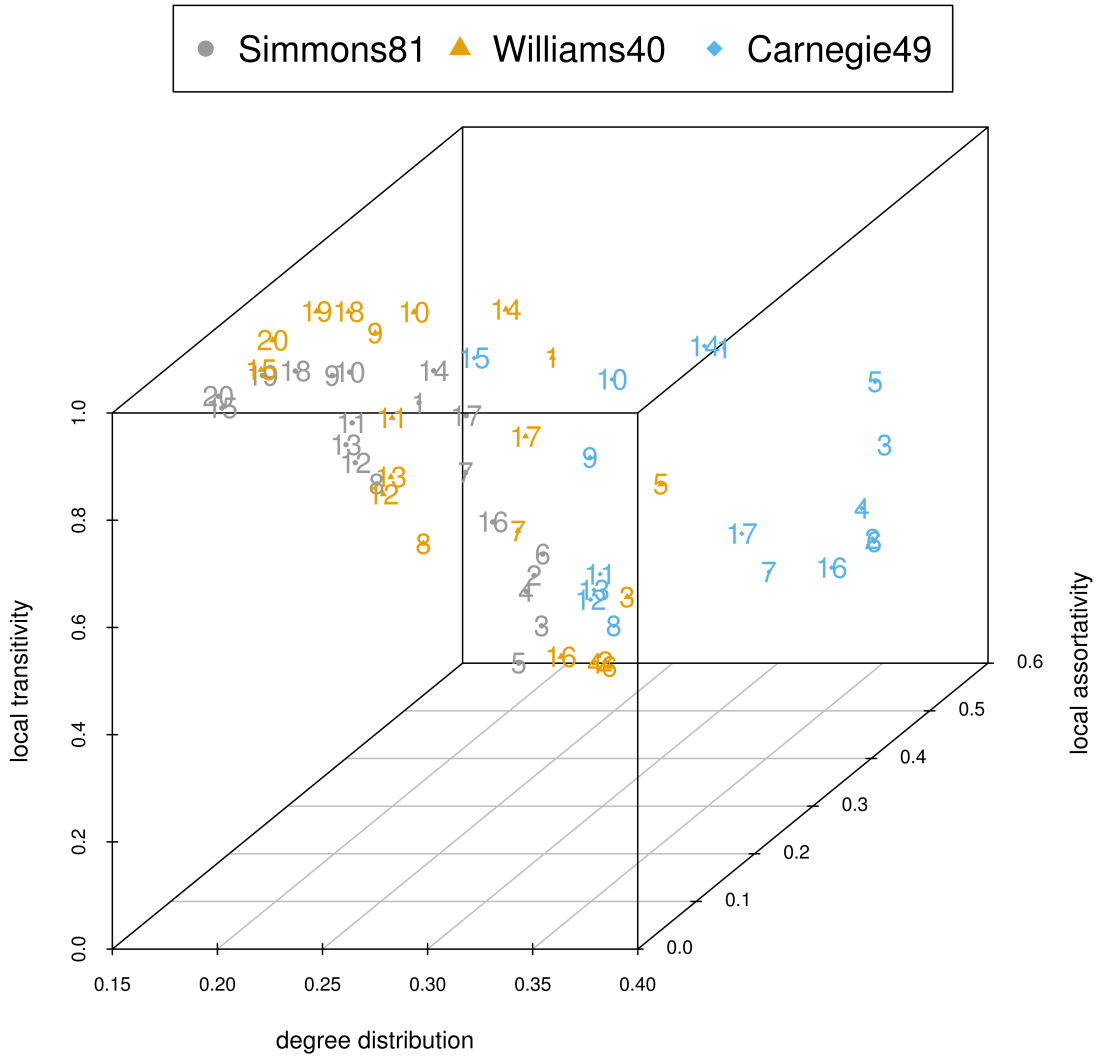


Figure A.3. A 3D visual comparison of 20 Pareto optimal solutions for Simmons81 Facebook network when they were used for extrapolating to Williams40 and Carnegie49.

APPENDIX B
DATASETS AND PACKAGES

Some useful websites for downloading network data:

- Network repository: `http://networkrepository.com/networks.php`
- Index of Complex Networks: `https://icon.colorado.edu/`
- UCI Network Data Repository: `https://networkdata.ics.uci.edu/`
- KONECT database: `http://konect.cc/networks/`

B.1 Data used in Chapter 4

Following real world networks were used: network of word adjacencies [310], US politics books sold on Amazon [311], a network of co-appearances [312], network of American football games [313], collaboration network between Jazz musicians [314], a social network of dolphins [315], brain networks (with different correlation cutoffs) [269], two protein networks [316], a network of yeast protein interactome [317], three networks obtained from the Biogrid repository [318], US Airport network [319], Norwegian boards network [320], human protein interaction network [321], social network from an online community for students at University of California, Irvine [322], and a network representation of the topology of the Western States Power Grid of the United States [13].

The protein networks 1php and 1qop were obtained from the Protein Data Bank [316]. The pdb files obtained from this database contains information about all atoms composing a given protein. This can be used to obtain contact maps containing key relations from the protein structure. The C-alpha atoms were chosen from the pdb files and a contact map was obtained using a threshold of 8 Å, i.e. if the distance between two atoms i and j is less than 8 Å, then the undirected link (v_i, v_j) exists. The networks Biogrid FRET, Far Western and Dosage Lethality were obtained from the biogrid database available at [318]. Brain fMRI data was obtained from Human Connectome Project dataset [269]. The parcellation scheme described in [323] was used to define nodes for network analysis. Correlation matrix for patient 1019436 was used at correlation cutoffs 0.7, 0.6 (brain 2), and 0.55 (brain 3) to generate the respective

Table B.1.
List of target networks along with some network properties: number of vertices n; number of edges m; mean degree z; average path length l; clustering coefficient c; and degree correlation coefficient r.

Network Name	n	m	z	l	c	r
Erdős-Rényi	100	500	10	2.236	0.103	0.041
Power Law	100	500	10	2.225	0.165	-0.089
Small World	100	500	10	2.383	0.275	-0.085
Barabási-Albert	100	485	10	2.208	0.177	-0.102
Forest Fire	100	311	6.22	3.089	0.361	-0.071
Stochastic Block	100	430	8.6	2.700	0.140	0.27
Word Adjacencies	112	425	7.59	2.535	0.157	-0.129
Political Books	105	441	8.4	3.078	0.348	-0.128
Co-appearances	77	254	6.59	2.641	0.499	-0.165
Jazz Collaborations	199	2742	27.56	2.235	0.520	0.02
Football Games	115	616	10.71	2.508	0.407	0.142
Network of Dolphins	62	159	5.13	3.357	0.309	-0.044
Brain (cor=0.7)	129	327	5.07	7.57	0.512	0.552
Brain (cor=0.6)	239	1039	8.69	4.87	0.542	0.577
Brain (cor=0.55)	252	1499	11.89	3.98	0.557	0.574
Biogrid FRET	987	1747	3.54	6.76	0.013	0.40
Biogrid Far Western	622	1073	3.45	5.27	0.010	-0.12
Biogrid Dosage Lethality	994	1780	3.58	3.41	0.002	-0.36
Protein 1php	394	1256	6.38	6.34	0.17	0.286
Protein 1qop	655	2243	6.85	6.82	0.193	0.35
Yeast Protein	426	521	2.45	6.02	0.021	-0.195
US Airports	500	2980	11.92	2.99	0.351	-0.268
Norwegian Boards (Aug. 2011)	854	2745	6.43	6.66	0.624	0.052
Human Protein	4100	13358	6.52	4.06	0.033	-0.216
Social Network	1893	13835	14.62	3.06	0.057	-0.188
US Power Grid	4941	6594	2.669	18.989	0.103	0.003

networks (largest component of the network was used as the target network). The Norwegian boards network [320] shows interaction among board members of public companies in Norway as obtained from the data in August 2011. The human protein interaction network was obtained from [321] and the largest connected component was used as the target network. The US Airport network [319] shows connections between airports (nodes) if there is a direct flight between them.

B.2 Supply chain data

We begin by describing the real-world supply chain dataset that was used in this research. Supply chain data provided by [324] and analyzed in [239] has been used to investigate the applicability of the proposed framework on different supply chain networks. The dataset contains 38 multiechelon supply chains used for inventory optimization purposes. The supply chains consist of firms with five different roles, namely, parts (suppliers), manufacturers, transportation, distributors and retailers. Tier information (the dataset used the term relative depth) is also available, and different supply chains have between 2 and 10 tiers. The SCNs described in this paper comprise actual supply chain maps created by either company analysts or consultants. This makes the dataset a perfect test bed for validating the efficiency and effectiveness of supply chain models. 10 among the 38 were selected based on network density and size, and they are listed in Table B.2 along with eight relevant SCN properties. Two of the SCNs are also shown in Figure B.1 for visual representation. The SCNs shown in Figure B.1 both posses a tiered structure, but the interconnectedness among various tiers and the number of nodes in each tier is very different in the two networks.

A key limitation of the SCN dataset is the absence of data on geographical locations of individual firms. This information was not provided in the original dataset due to confidentiality reasons. As discussed earlier, geographical location might play a significant role in linking decisions of firms and its unavailability might significantly limit our understanding of various structural features. Furthermore, this empirical study does not explore the dynamic nature of the SCNs since the dataset does not provide any information pertaining to temporal changes in the SCN topology. Lastly, the dataset does not provide information regarding amount of material flow between connected firms. Although specific production capabilities of firms within each tier are known, no information is available in relation to how much each upstream firm supplies to the downstream firms. Nevertheless, the size of the dataset, both in terms

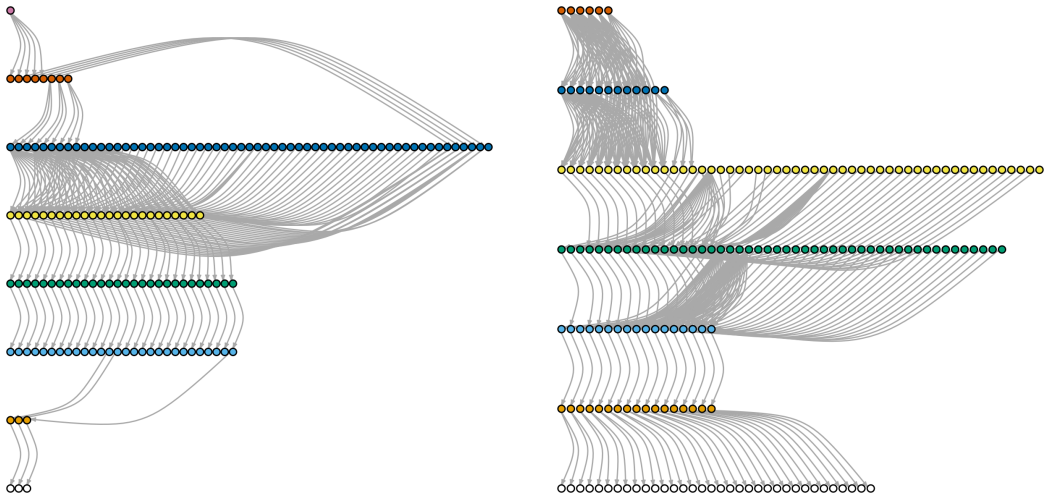


Figure B.1. A visual representation of the tiered structure of the real-world supply networks obtained from the dataset. The images correspond to supply chains of Computer Peripheral Equipment (left) and Perfumes, Cosmetics, and Other Toilet Preparations (right)

of the number of networks available and in terms of number of nodes in each network, make this a very attractive dataset to study.

Table B.2.

List of real-world SCNs used for modeling along with relevant network properties of the both the target G^* and synthesized G networks: total number of vertices n; total number of edges m; mean degree $\bar{k}$; average path length (apl); network connectivity (NC); network heterogeneity (NH); fitted power law coefficients for in (α-in) and out (α-out) degree distributions.

					apl		NC		NH		α-in		α-out	
Ref #	SIC Description	n	m	$\bar{k}$	G^*	G	G^*	G	G^*	G	G^*	G	G^*	G
13	Semiconductors and Related Devices	108	452	8.37	1	1	0.74	0.59	1.84	1.52	2.43	2.34	11.61	7.64
18	Computer Peripheral Equipment	154	224	2.91	2.62	2.32	0.067	0.074	1.02	0.79	3.02	4.41	3.68	5.96
21	Perfumes, Cosmetics, and Other Toilet Preparations	186	359	3.86	2.82	3.03	0.082	0.087	1.06	0.91	5.44	3.28	3.18	3.46
24	Power-Driven Handtools	334	1245	7.45	2.70	2.70	0.18	0.43	1.83	2.05	2.07	1.95	2.15	5.19
27	Electromedical and Electrotherapeutic Apparatus	482	941	3.90	1.96	2.32	0.18	0.42	2.68	3.07	2.66	2.09	3.08	3.36
28	Computer Storage Devices	577	2262	7.84	2.56	2.48	0.34	0.14	2.47	1.50	2.11	2.40	3.14	10.88
32	Perfumes, Cosmetics, and Other Toilet Preparations	844	1685	3.99	1.96	2.38	0.074	0.027	1.56	1.10	2.42	6.99	2.59	4.36
34	Telephone and Telegraph Apparatus	1206	4063	6.74	1.07	1.29	0.12	0.33	2.75	3.94	2.88	3.17	2.32	7.43
36	Farm Machinery and Equipment	1451	4812	6.63	1.66	1.77	0.40	0.19	2.68	2.38	10.47	2.29	2.06	5.37
38	Aircraft Engines and Engine Parts	2025	16225	16.02	2.38	2.47	0.11	0.10	2.45	2.18	2.86	21.21	1.9	16.76

B.3 Network population data

To examine the ability of existing generative models to approximate the ground truth process using a single network observation (assuming it is representative of the true process with respect to the measures of interest), we propose two different experiments: (i) a controlled experiment where the true process is known, and (ii) set of real-world networks that have most likely evolved from a common generative process (for example, social interaction networks of different villages).

Table B.3.
Statistics and network metrics of network populations (standard deviation in parentheses), where n is the number of nodes in the network and APL is the averaged shortest path length between all pairs of nodes.

Name	Sample Size	n	Density	Transitivity	Assortativity	APL
Barabási-	100	100	0.098	0.173	-0.082	2.22
Albert		(0)	(0)	(0.0069)	(0.036)	(0.016)
Forest	100	200	0.058	0.406	-0.032	3.102
Fire		(0)	(0.017)	(0.043)	(0.098)	(0.401)
SBM	100	150	0.072	0.177	-0.512	2.72
		(0)	(0.0025)	(0.0097)	(0.054)	(0.0527)
Brain	100	360 (0)	0.032	0.422	0.141	3.73
Networks		(0)	(0.001)	(0.011)	(0.042)	(0.129)
Contact	69	167.21	0.045	0.470	0.362	4.104
Networks		(64.71)	(0.012)	(0.143)	(0.215)	(1.091)
Social	43	212.23	0.048	0.198	-0.078	2.77
Networks		(53.54)	(0.013)	(0.037)	(0.054)	(0.0207)
Travian	30	1144.5	0.0039	0.019	-0.055	4.35
Trades		(123.22)	(0.00055)	(0.003)	(0.036)	(0.134)
Travian	30	1722.2	0.0026	0.108	-0.513	2.72
Messages		(180.76)	(0.00023)	(0.022)	(0.034)	(0.244)
Autonomous	100	3196.3	0.001099	0.015	-0.221	3.77
Systems		(101.7)	(0.000027)	(0.0013)	(0.004)	0.014

- Barabási-Albert: The Barabási-Albert model [12] was used to synthesize networks with each arriving node adding 5 edges using the linear preferential attachment mechanism.

- Forest Fire: Network populations synthesized using the Forest Fire model [212] used a forward burning probability $p = 0.38$.
- Stochastic Block Model (SBM): The Stochastic Block Model was used to synthesize networks with 3 assortative communities of sizes 30, 70 and 50.
- Brain Networks: We used DWI data from the 100 unrelated subjects of the HCP 900 subjects data release [269] to get the structural brain networks. The preprocessing of the DWI data to get the corresponding networks is described in [325]. One network represents the abstracted brain structure of one subject. Nodes in network represent regions of interest (ROIs) in brain and edges represent the density of connecting fibers. All networks share the same set of nodes since brain images of different subjects are regularized into a common template of ROIs. This data was also used for the experiments outlined in Chapter 6.
- Contact Networks: 69 daily cumulated networks where nodes represent visitors of the Science Gallery while the edges represent close-range face-to-face proximity between the concerned persons [326]. Since visitors showing up on different days are different, the node sets are not fixed for the 69 networks.
- Social Networks in Indian Villages: Data from a survey of social networks in 75 villages in rural southern Karnataka, a state in India [327]. One network represents the social network of one village. Nodes in the network represent individuals and edges represent different social interactions.
- Travian Network Datasets: Data collected over 30 days for real-time strategy game Travian. The message network contains links for messages sent between players, while the trade network represents trading relations [328].
- Autonomous Systems: The graph of routers comprising the Internet can be organized into sub-graphs called Autonomous Systems (AS). The dataset [329] contains 733 daily instances spanning an interval of 785 days from November 8 1997 to January 2 2000. The first 100 networks were used in this study.

Table B.3 shows the statistics and some common network metrics of listed datasets. First three populations are generated from parameterized network models, and the the rest six are networks obtained from real-world interactions. Among the real-world populations, the Travian and contact networks are created from interactions among different sets of individuals across different days, but the underlying systems that supports these interactions remains the same. The networks are thus different instances of interaction processes happening on a fixed system, and can thus be hypothesized to

have a common generative process. The structural organization of the human brain is controlled by the human genome, and it is safe to assume that the network representation of different individuals belongs to a population. Similarly, it is reasonable to assume that social interactions between individuals in different villages arise from similar generative mechanisms. Daily instances of subgraphs of Autonomous Systems can again be assumed to have a common underlying generative process.

B.4 Packages and implementations

The implementation of ABNG has been done using the `igraph` package in R. `igraph` was also used for synthesizing target networks from other human-devised network generators (listed in Table B.1) and performing statistical analysis on various networks. The following other packages and implementations were used:

- The `parallel` package in R was used to parallelize the implementation wherever possible.
- The `ergm` package in R was used to synthesize networks using the exponential random graph model.
- Networks from the Chung-Lu model were synthesized using the implementation in the `igraph` package.
- The `spectralGOF` package in R was used to compute SGOF values.
- The implementation of *dk*-random graph generator available at `https://github.com/polcolomer/RandNetGen` was used.
- The implementation of the Symbolic regression based approach available at `https://github.com/telmomenezes/synthetic-old/` was used.
- The implementation of the D-measure is available at `https://github.com/tischieber/Quantifying-Network-Structural-Dissimilarities`.
- The implementation of NSGA-II in the R package `mco` was used for parameter optimization.
- The implementation of the microcanonical SBM in `graph-tool` is available at `https://graph-tool.skewed.de/`.

VITA

VITA

Viplove Arora received his Bachelor of Technology degree in Production and Industrial Engineering from the Indian Institute of Technology, Delhi, India in 2014. After completing his bachelor degree, Viplove moved to Purdue University, West Lafayette, Indiana in 2014 to pursue a masters degree in Industrial Engineering. In 2015, Viplove decided to pursue a Ph.D. degree under the supervision of Dr. Mario Ventresca in the School of Industrial Engineering at Purdue University. During the Ph.D., Viplove has worked on the development of conceptual, mathematical, and computational tools for modeling complex systems. His research interests include network science, complex systems, algorithm design, simulation, multi-objective optimization, machine learning, and mechanism design.

LIST OF RESEARCH PAPERS

Journal Publications

1. **V. Arora**, D. Guo, K. D. Dunbar, M. Ventresca, *Quantifying the Variability in Network Populations and its role in Generative Models*, (Accepted: Network Science)
2. S. R. Hunter, E. A. Applegate, **V. Arora**, B. Chong, K. Cooper, O. Rincon-Guevara, C. Vivas-Valencia, *An Introduction to Multi-Objective Simulation Optimization*, ACM Transactions on Modeling and Computer Simulation (2019)
3. **V. Arora**, M. Ventresca, *Modeling topologically resilient supply chain networks*, Applied Network Science (2018)
4. **V. Arora**, M. Ventresca, *Action-based Modeling of Complex Networks*, Scientific Reports (2017)

Conference Proceedings

1. **V. Arora**, M. Ventresca, *Evaluating the Natural Variability in Generative Models for Complex Networks*, International Workshop on Complex Networks and their Applications, 2018
2. **V. Arora**, M. Ventresca, *Action-Based Model for Topologically Resilient Supply Networks*, International Workshop on Complex Networks and their Applications, 2017

3. D. Guo, **V. Arora**, E. Amico, J. Goñi, M. Ventresca, *Dynamic Generative Model of the Human Brain in Resting-State*, International Workshop on Complex Networks and their Applications, 2017
4. **V. Arora**, M. Ventresca, *The inverse problem of discovering complex network generators*, ICIPE, 2017
5. **V. Arora**, M. Ventresca, *A Multi-objective Optimization Approach for Generating Complex Networks*, GECCO, 2016

In Preparation

1. **V. Arora**, M. Ventresca, *Action-based Representation of Complex Networks: Theory and Inference*
2. **V. Arora**, M. Ventresca, *Action-based model for network data with errors*
3. **V. Arora**, E. Amico, J. Goñi, M. Ventresca, *Investigating cognitive ability using action-based models of structural brain networks*
4. M. Ventresca, **V. Arora**, *Quantifying complex system entropy: an action-based perspective*