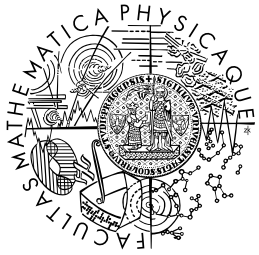


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Daniel Trlifaj

Graphlets in Complex Networks

Department of Applied Mathematics

Supervisor of the bachelor thesis: Ing. David Hartman, Ph.D.
Study programme: Computer Science
Study branch: General Computer Science

Prague 2023

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I would like to thank my supervisor, Ing. David Hartman, Ph.D., for introducing me to network science and for his patient guidance, kind support, and contagious excitement for the field.

I would also like to thank my family for their support and understanding on this eventful journey. My deepest gratitude also extends to the Tydlitat family who provided me with the most hospitable conditions for writing in the dauntingly hot weather of this summer.

Title: Graphlets in Complex Networks

Author: Daniel Trlifaj

Department: Department of Applied Mathematics

Supervisor: Ing. David Hartman, Ph.D., Computer Science Institute of Charles University

Abstract: Analyzing the characteristics of complex networks is a principal task of network science. In this thesis, we study graphets, small induces subgraphs rooted in a vertex, as a tool to describe and compare networks. First, we use graph theory to explore the theoretical properties of graphlets, propose a framework for studying them, and make novel observations. We discuss the link between graphlets and the Weisfeiler-Lehman isomorphism test and the reconstruction conjecture. We prove that the knowledge of graphlets of size $n-1$ for certain graphs is sufficient for their reconstruction. Second, we develop several graphlet-based metrics and apply them to real-world networks and their models. In accordance with prior literature, the results suggest that graphlets are potentially an excellent tool of characterizing networks. In contrast with prior literature, the results suggest that the Albert-Barabási model produces more realistic synthetic networks than other models.

Keywords: graphlets, complex networks, random network models, graph motifs

Contents

Introduction

Network science is a relatively new field of research that has seen a flurry of activity over the past two decades. It is characterized by abstracting a wide array of real-world phenomena, be it social connections (Wasserman and Faust [1994]; Zachary [1977]), road infrastructures (Reza et al. [2022]), or protein-protein interactions (Jeong et al. [2001]; Stelzl et al. [2005]), into the framework of graph theory, which enables researchers to trace universal patterns in otherwise exceedingly complex systems (Barabási [2013], Braha [2018]). This approach, on its own, is by no means new – it can be argued that Euler's approach to the Königsberg bridge problem from 1741 (Euler [1741]) or Milgram's and Travers' work on the small world problem[1] (1977) can be granted this characteristic. Nonetheless, two key developments from the 1990s helped establish network science as a distinct field independent of graph theory, mathematical sociology, or systems engineering. First, with the rapid development of a variety of computational technologies, especially the internet, we see the production of vast amounts of data from all walks of life that can be translated into the language of graphs. Second, in parallel, a number of key results and techniques[2] were developed, showing that approaching real-world phenomena through the framework of networks enables us to trace non-trivial structures in the world and even successfully grapple with dynamic processes on these networks (Newman [2003]). Network science, thus, appears as a promising path to engage with the data-saturated world of today.

A fundamental inquiry in the field of network science revolves around the development of a null model, a model that generates a class of graphs corresponding to real-world networks. The development of a null model is of great relevance since it would enable us to predict network structures in cases where the information about them is incomplete[3] and identify statistically significant properties of networks (Chen [2022]). Nevertheless, formulating a sensible model is a rather complicated matter as we need to enable the generation of a large enough class of graphs to capture the diversity of real-world networks, ideally specifying subclasses through the choice of parameters while maintaining their underlying structure and logic of formation[4].

There have been a number of attempts to create a null model, some more successful than others. These attempts range from appropriating Erdős-Rényi random graphs (Erdős et al. [1960]; Seshadhri et al. [2012]) to the attempt by Albert and Barabási to harness the properties of the degree distribution in real-world networks (Barabási and Albert [1999]; Albert [2005]) and an array of highly

[1] Milgram showed that individuals in the USA are connected by surprisingly short paths. Specifically, he showed that on average any two individuals in the country can be connected to each other through a chain of no more than six acquaintances. In the language of network science, we would say that the average path length between any two vertices in the social network of the USA is 6.

[2] Here, we refer to the scale-free and small-world properties, community structure, or cascading failures. For a comprehensive exposition of the results and methods of network science, see (Newman [2003]).

[3] An illustrative instance from the recent past is the transmission of COVID-19 within a specific community's social network. We do not have comprehensive information about individual contacts, and yet we would like to simulate the spread of infection.

[4] For a more detailed discussion, see chapter 1

specialized models based on the type of network under scrutiny (Haddadi et al. [2008]; Milenković et al. [2009]). However, the questions of how to construct a good model and what this *good* means remains open and a matter of active research in the field (Chen [2022]).

In this thesis, we approach these two questions through the lens of *graphlets*, small local graphs [5] initially introduced by Pržulj (2004a). The goal of this thesis is to explore the graph-theoretical properties of graphlets and demonstrate how they can be used in network modeling. The structure of the thesis corresponds to this effort.

In the first chapter, we attempt to shed some light on the thinking employed in network science and the way network modeling is approached. We present a simplified history of the development of network science to establish the context for our endeavor. Subsequently, we introduce the problem of network characterization and some of the commonly used tools and results tied to it. Eventually, we discuss classical models used for network modeling and outline the process of model creation in relation to the problem of network characterization.

In the second chapter, independently of the first, we turn to graphlets. We formally define graphlets and related concepts, discuss known results, and make a few observations about their properties. We also suggest connections with the Weisfeiler-Lehman isomorphism test and the reconstruction conjecture.

Finally, in the third chapter, we connect the previous two chapters, and we discuss how graphlets might be used for characterizing and comparing networks and how they might contribute to the effort of finding a better model for complex networks. We apply graphlet-based measures on real-world networks and, backed by both analytical and empirical results, suggest that graphlets can serve as a strong tool for network comparison and modeling with an abundance of untapped potential.

Terminology

In this thesis, we use standard terminology from graph theory and the following terminology from network science[6]:

- *network*: A graph (V, E) that is either constructed from or used for modeling real-world phenomena. These two concepts, graph and network, reference the same structure, a graph defined by vertices and edges; nevertheless, when we refer to a network, we emphasize the link with real-world phenomena observed through the lens of graphs. Although blurry, the distinction provides us with a better idea of the intentions that we have when working with these structures.

- *local topology*: The structure, meaning the way vertices are connected by edges, in local regions of a graph—for example, an induced subgraph rooted in a particular vertex. The definition is not related to the field of mathematical topology.

[5] You can find a more technical definition in 2.1.

[6] It should be noted that due to the relative novelty and interdisciplinary nature of the field of network science, the terminology is not clearly established.

- *complex local topology*: We say that a region of a graph has a complex local topology if it is locally highly connected.
- *null model*: A simplified model for synthetic network generation that is used as a baseline or reference point during the analysis of real-world networks. It aims to capture expected patterns in real-world networks as well as random phenomena. It serves as a null hypothesis against which novel network properties or behaviors can be evaluated.
- *k-neighborhood of v*: The set of vertices that are within a distance of k edges from the vertex v. We denote the k-neighborhood of v by $N_G^k(v)$.

Concepts relevant only to specific sections are introduced in appropriate parts of the thesis.
All figures, unless stated otherwise, are original.

1. Modeling of Complex Networks

The approach taken by network science to the real world is by no means self-evident. Why should translating real-world phenomena into the language of graph theory be a useful tool in understanding the reality of the world? What is the motivation for developing models of reality within network science? Neither of these questions has an obvious answer. In this section, we try to shed some light on the thinking within network science and, hopefully, at least partially answer these questions. We do this in three steps. First, we discuss the context and motivation for the development of network science. Second, we discuss classical tools used to describe real-world networks that help us capture universal patterns in the world. And, third, we discuss what the role that modeling plays in this, what are the classical models and how can we compare them to establish a good representation of reality.

Admittedly, throughout this chapter, the presentation of many things is one-sided and narrow in scope – some of the claims that we make in a single sentence would deserve chapter-long discussion. We acknowledge this fact and point to appropriate literature whenever needed to at least partially remedy this shortcoming. Nevertheless, we still find it important to provide at least a broad overview of motivation and thinking for network science and modeling of networks to better contextualize the usage of graphlets in the discipline.

1.1 Historical context and motivation

Throughout the second half of the 20^{th} century, in conjunction with the rise of positivist science in the natural sciences, there was a push toward the rationalization, instrumentalization, and quantification of various observed phenomena, processes, and activities (Ritzer [1996]; Gorman [2006]). In applied mathematics, we can observe this in the development of disciplines such as linear programming, which aims to optimize logistics in war and post-war efforts, and game theory, which attempts to model real-world behaviors of (mostly) rational agents. As a result of the spread of this kind of thinking and the development of specialized technologies, a significant amount of information about the world are captured and described in terms of quantifiable data and measurements (Bowker and Star [2000]). The proliferation of qualitative data and the advancement of technologies not only contributed to the progress of existing scientific disciplines from which the data originated but also stimulated the exploration of new approaches. These possibilities became even more relevant with the widespread use of fast computers, which enabled the production and processing of previously unimaginable amounts of information, and the internet, which facilitated the decentralized production of vast amounts of data. However, as the amount of detailed data increased, so did the complexity of the observed phenomena, surpassing the capabilities of classical approaches and structures. This raised the question of how we can effectively describe complexity in such intricate structures.

To respond to this challenge and possibility, researchers mobilized existing

work in the field of graph theory, which has proven immensely useful in describing relational structures. Representing data as graphs provides us with an unified framework that allows for comparable information about the observed relational structure. This enables us to describe, gain insights and analyze complex phenomena in a more comprehensive and integrated manner (Börner et al. [2007]). This led to the establishment of network science.

As such, when we want to use a network science approach for analyzing reality, there are two principal steps that we must take – we first need to translate real-world phenomena into the framework of networks and, then, we can analyze some of their properties. This first step is often very complicated – in some cases, such as friendship networks on Facebook or emails between addresses, the translation into a network is relatively simple, but in most cases, such as neural networks, ecological food chains, and friendships in between people, the way we define vertices and edges are blurry at best and the information with which we work is undoubtedly incomplete. We do not intend on diving into the problems further, but it should be noted that we should not perceive real-world networks that we work with as a pure and accurate representation of the real world and this consideration should be at least noted when working and thinking about networks. In this text, we focus on the second step – working with premade networks representing real-world data.

In the next sections, we discuss how the framework of networks can be used to describe real-world phenomena and how we can model them.

1.2 Describing networks

The approach of network science enables us to, through large networks describe phenomena that otherwise are far beyond what we can perceive. However, making observations based on a large network alone and tracing universalities across a multitude of networks still remains beyond human abilities. For this reason, several ways to characterize networks were devised. In this section, we introduce some of them and showcase how they can be used to identify universal patterns across different networks:

1.2.1 Characteristics of networks

Generally speaking, there are two ways of characterizing a network – we can approach it through local properties of a network or through global properties [1]. In this subsection, we describe some of the most commonly used global and local characteristics, for a more complete list of network characteristics, see Zhou et al. (2004) or Newman (2003).

We always discuss properties on a connected graph G with $|V(G)| = n$ vertices and $|E(G)| = m$ edges.

[1] This distinction is taken from Newman (2010). There are also fundamentally different ways of characterizing a network such as the eigenvalue of incidence matrices or Laplace matrices but discussion of them is beyond the scope of this text. We refer to Boccaletti et al. (2006) for more information.

Global

Global characteristics consider the entire network all at once and calculate some information about the global behavior of vertices. Commonly, we obtain a single number characterizing the entire network.

- *average length of a path* : calculates the average shortest path between any two vertices in network – if we denote dist_{uv} the shortest path between u and v, we can define average length of a path as $\frac{1}{n}\sum_{u,v\in V(G)}\text{dist}_{uv}$.
- *diameter* : closely related to the average length of a path, it calculated the longest shortest path between any two vertices in a network, i.e. the diameter of a network is $\max_{u,v\in V(G)}\text{dist}_{uv}$.
- *associativity* : measures the tendency of vertices to connect with other vertices that have similar degrees. It assesses the mixing patterns of high-degree and low-degree vertices in the network. The measurement is carried out by the assortativity coefficient, and a variation on the Pearson correlation coefficient. For a thorough formal definition see Newman (2002).
- *graph motifs* : calculate the "patterns of interconnections occurring in complex networks at numbers that are significantly higher than those in randomized networks" (Milo et al. [2002], p.1). This can be used in tracing overrepresented and underrepresented structures in a network.
- *modularity* : quantifies the presence of community structure within a network. It evaluates the division of vertices into distinct modules or communities based on their connectivity patterns. The measurement of modularity is carried out by splitting the whole network into groups and subsequently comparing how many or fewer edges are between groups compared to a null model, commonly the Erdős-Rényi model (see section 1.3.2). For a technical description of the process, see Brandes et al. (Brandes et al. [2007]).
- *global clustering coefficient* : measures the degree to which vertices in a network tend to form clusters, tightly interconnected groups. It provides an overall measure of the extent of clustering in the entire network. It is commonly calculated based on the number of triangles in the network compared to the number of triangles that might occur in the network. Concretely: $C(G) = \frac{3*\text{ number of triangles in the network}}{\text{number of potential triangles in the networks}}$ where $C(G)$ is the global clustering coefficient.

Local

Local characteristics commonly consider individual vertices and determine their behavior in the network. Commonly, we obtain a distribution of values for each vertex or edge which subsequently needs to be further processed so that networks are comparable.

- *degree distribution* : calculates the degree of each vertex and produces a degree distribution of the network.

- *betweenness centrality* : quantifies the extent to which a vertex lies on the shortest paths between other vertices in the network which might be interpreted as its potential influence in information flow. Concretely, if we denote σ_{st} the total number of shortest path between $s, t \in V(G)$, $\sigma_{st}(v)$ the number of such paths that include vertex v, we can define the betweenness centrality of a vertex, v, as $g(v) = \sum_{v \neq s \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$ (Brandes [2001]).

- *vertex clustering coefficient* : measures the extent to which its neighboring vertices are connected, indicating the level of local clustering around the vertex. For a thorough formal definition see Newman (2002).

1.2.2 Properties of real-world networks

Based on these characteristics, network scientists were able to identify universal patterns that occur in most real-world networks. In this section, we introduce a few representative examples [2]:

- *small-world property* : when we measure the average length of the shortest path between any two vertices in a network (call it $L_{\text{avg}}(G)$ where G is a network on n vertices (Watts and Strogatz [1998])) in combination with their clustering coefficient (call it $C(G)$), it turns out that for the majority of real-world networks, the value of L_{avg} is proportional to the logarithm of the number of vertices in the network, i.e. $L_{\text{avg}} \propto n$ (Barrat and Weigt [2000]; Dorogovtsev and Mendes [2003]; Barmpoutis and Murray [2010]) and the clustering coefficient $C(G)$ is larger than that of a random network with the same degree on average (we usually use Erdős-Rényi model introduced in section 1.3.2).

- *scale-free property* : inspired by the research of Prince (1965), the scale-free property requires that a degree distribution of a network follows a power law, i.e. a few vertices have a large degree while the majority of vertices have a relatively low degree. It was shown in many networks that this distribution follows power-law distribution rather than the normal or Poisson distribution that one might expect (Barabási and Albert [1999]; Faloutsos et al. [1999]; Onnela et al. [2007]). Although it should be noted, researchers claim that the fact that we commonly observe degree distribution mimicking power-law is a product of the data sampling (translating data into the framework of networks as mentioned in 1.1) (Stumpf et al. [2005]; Haddadi et al. [2009]; Memišević et al. [2010]) and some claim that scale-free property is not as common as it has been assumed (Broido and Clauset [2019]).

- *community structure* : networks are said to have community structure if the graph can be divided into groups, communities, that are internally densely connected whereas connections between groups, communities, are sparse [3].

[2] We choose those that do not require an extensive discussion since the results are of not of central interest to this text. We provide them to give a sense of what describing real-world networks can be used for.

[3] Thorough formal definition is rather complicated as we need to define what a community can be, how dense/sparse connections can occur and whether communities can overlap. For a thorough treatment of this topic see Girvan and Newman (2002)

Communities are quite common in networks from real-world data (Porter et al. [2009]; Fortunato [2010]; Bedi and Sharma [2016]).

These characteristics occur in many seemingly unrelated networks which gives us empirical support for their soundness and the meaningfulness of network science more broadly.

1.3 Modeling a network

Beyond description, abstracting the world into the language of networks enables us to attempt to model reality that we can observe. Modeling reality can help us better understand complex network data (Memišević et al. [2010]) and potentially predict a variety of structures and analyze their functionality [4]. Researchers are actively trying to find a sensible well-fitting model provided by a number of vertices, n, and some parameter is capable of producing a graph with a structure similar to that of real networks – this effort, nonetheless, is no easy undertaking since we do not exactly know what is the structure that is supposed to be modeled and even the data themselves are sometimes of uncertain corresponding value to reality. The main challenge is to find a way to generate class of graphs that includes exactly real-world networks – large enough to include all possible variations (this is usually done by making the models random in core – in this way, they are theoretically able to generate any possible graph) but small enough to avoid generating networks that cannot be real-world networks (the idea behind this can be seen on the figure 1.1).

1.3.1 Heuristics

When we are trying to create a model, we have to establish what a good model is based on the real-world networks that we have. To do that, we have to somehow show which networks are "closer" to each other – to use the framing visualized in figure 1.1, we have to create a "metric" to show that the two sets truly overlap. How is it possible to compare such complex relational structures as graphs? To do that, it is common to use a set of heuristics that enable us to approximate the similarity of different networks and subsequently compare them. If the heuristics are good at characterizing real-world networks they can become the basis for a new model. In this section, we present models of real-world networks and highlight how heuristics are used for the evaluation of their quality.

Noted that, in the field of network science, heuristics used for network comparison are based on what we described as characteristics in section 1.2 used for a network description. Since the intention when using these methods differ when they are descriptors, we try to trace universal properties, and when they are heuristic, we are capturing the network as a whole, and we keep the linguistic distinction.

[4] This can help us better predict spreading phenomena, such as propagation of contagious diseases (Hiram Guzzi et al. [2022]), and analyze structure, for example when trying to find a new possible chemical (Hu et al. [2011]) or identify functional parts of the brain (Telesford et al. [2011]).

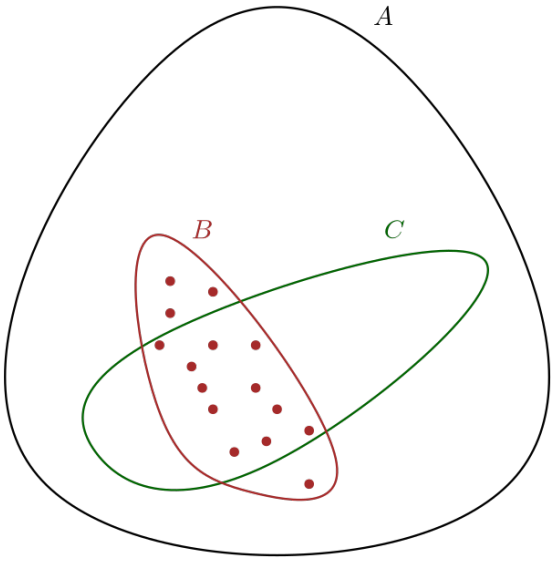


Figure 1.1: Problem of modeling networks. A is the set of all graphs on n vertices, B is the set of all real-world networks (dots inside symbolize networks about which we have data) and C is the set of graphs produced by a given model (note that most of the models are probabilistic so a better representation would be a heatmap). The goal of modeling is to make the set C match the set B.

1.3.2 Models of networks

In this subsection, we define some of the most commonly used models in network science and discuss their strengths and weaknesses. We do not discuss in depth their properties and function since that is beyond the scope of this text (we point to appropriate literature) and we do not judge which model is better than the other. Rather, by providing a coarse overview of the models we try to offer a window into the thinking and practices related to model-making.

Erdős-Rényi model

The simplest model in use is Erdős-Rényi random model (ER for short) (Erdős et al. [1960]), initially a mathematical experiment joining graph theory and probability theory. There are two ways how we can introduce ER model. First, denoted by $ER(n, m)$, we can consider two parameters $n, m \in \mathbb{N}^+$ corresponding to the number of vertices n and m edges that we connecte between randomly selected $2m$ vertices. Second, denoted by $ER(n, p)$, we can define the model by $n \in N^+$ and $p \in [0, 1]$ – than the model is created by considering all possible edges on n vertices and placing them in the final graph with probability p[5]. Clearly, $ER(n, m) \approx ER(n, p)$ if $p = \frac{2m}{n(n-1)}$ (m divided by the number of all possible edges on n vertices).

The Erdős-Rényi model has been extensively studied by graph theorists and lies at the core of the discipline of random graphs which led to a cornucopia of fascinating results (see Bollobas (1998)). From the perspective of network science, the model is characterized by relatively short average path lengths but limited

[5]This model is sometimes called Erdős-Rényi-Gilbert model Fienberg [2012] after Edgar Gilbert (1959) who proposed it at the same time as Erdős and Rényi.

community structure, small value of global clustering coefficient, and absence of scale-free property, its degree distribution does not follow power law.

It should be noted that a variant of the Erdős-Rényi (ER) model is the 2-configuration model (which we encounter in section 2.3.2).

Geometric model

Another model, the Geometric model (GEO for short) (Waxman [1988]), generates a graph by placing n vertices into a metric space and adding an edge between two vertices if and only if two vertices are closer to each other than some threshold r. The classical definition, using the notation established by Penrose (2003), is the following:

Given the number of vertices, n, and a parameter $r \in (0, 1)$, let us have a metric space $[0, 1)^d$ with Euclidean distance. We sample the values of n points in the space from a uniform distribution from the space $(0, 1]^d$ and connect any two points if and only if the distance between the two points, excluding loops, is less than r. This gives us a graph.

There are many modifications of the GEO model that for example also make the connection of edges probabilistic rather than making it determined by the threshold.

Geometric models prove to be a relatively good model based on most aforementioned heuristics – they exhibit a high clustering coefficient due to how they are constructed (vertices that have edges between each other need to be spatially close to each other which increases the probability that there will be an edge also between its neighbors). Furthermore, they can, but do not have to, exhibit scale-free property (Memišević et al. [2010]).

They exhibit a small world and scale-free properties whilst the exhibition wanted community structure. Some even heralded it as the most fitting model (Memišević et al. [2010]).

Albert-Barabási model

A slightly different approach was taken by Albert and Barabási (1999) who, focusing on the scale-free property, proposed the Albert-Barabási model (AB model for short) that produces networks with scale-free degree distribution. The main principle used in the generation of AB models is *preferential attachment* – when iteratively building a network model, we preferentially attach a new vertex to a vertex with a high degree. More exactly the algorithm is as follows:

Algorithm 1 Albert-Barabási model

Input: number of vertices n, number of edges m
Output: graph on n vertices and m edges

1: start with a graph, G, of one vertex
2: **while** there are less than n vertices in the graph **do**
3: take a new vertex v and consider the existing graph G
4: connect v to existing vertices in the graph with probability p_{uv} where $p_{uv} = \frac{deg_G(u)}{\sum_{w \in V(G) deg_G(w)}}$
5: **end while**

The scale-free model is based on the degree distribution characteristic of real-world networks and, thus, replicates this property well – generates networks whose degree distribution follows the power law (Albert and Barabási [2002]) which has been perceived as one of the most important aspects of real-networks (Fienberg [2012]). Furthermore, the AB model exhibits small-world property. Nevertheless, AB model, in most cases, fails to display community structure (Börner et al. [2007]) and, in recent years, the importance of scale-free property has been questioned – some argue that the strict scale-free property might be a side effect of data sampling (Memišević et al. [2010]; Haddadi et al. [2009]) whilst others claim that scale-free networks are not too common among real-world networks (Broido and Clauset [2019]) (it should be noted that both of these claims are a matter of active debate, see (Voitalov et al. [2019])).

There are many other models, such as GLP (Bu and Towsley [2002]), NLPA (Kunegis et al. [2013]), or Watts-Strogatz model (Watts and Strogatz [1998]) to name a few, discussion of which is beyond the scope of this text. Nonetheless, practically all of them are a modification of the three aforementioned models and, thus, they should provide a reasonable idea about how modeling is approached.

We can see that different models are good according to different heuristics and, thus, when we are trying to create a reasonable network model of real-world networks, it is essential to select a sensible heuristic to identify appropriate models. The question of which heuristic is the most appropriate one is opened in the next section.

1.3.3 Choice of a heuristic

There exist many heuristics characterizing networks from various perspectives. But, it is far from clear which ones we should prioritize. Ideally, we would like to satisfy all characteristics, but that is often impossible in practice. Those that are rooted in empirical observation should be taken more seriously, but they are insufficient to characterize networks on their own (Tanaka et al. [2005]). Beyond that, the opinions about which heuristics are appropriate for characterizing networks differ.

Some prioritize local heuristics over global ones, for example, Pržulj argues that "although global properties of large networks are easy to compute, they are inappropriate for use on incomplete networks because they can at best describe the structure produced by the [...] sampling techniques used to obtain the partial networks" (Pržulj [2007], p. e178). In a similar vein, Haddadi argues that global characteristics disregard the complexity of local structures that might be crucial to understand the behavior of networks (Haddadi et al. [2008]) and Memisević argues that "local properties [...] impose a larger number of constraints, thus reducing degrees of freedom in which networks being compared can differ" (Memišević et al. [2010], p. 3).

But for example, Tanaka et al. criticize the usage of degree distribution as the principal heuristic since "networks of vastly different structures could have the same degree distributions" (Tanaka et al. [2005], p. 5142).

Further, some researchers, for example, Menisevic et al., argue that "it might be difficult to assess the reliability of the fit of any particular network model to the data with respect to a single network property since different models might

be identified as optimal with respect to different properties" (Memišević et al. [2010]) and for that reason propose a mixture of different heuristics which might better capture what is happening in the network (Memišević et al. [2010]).

All in all, there is no clear consensus on which heuristic exactly might be best fitted for network comparison. It seems that, in the ideal case, we would like a metric that is capable of capturing the local topology of networks whilst being capable of distinguishing different networks that differ in global characteristics and that includes, at least implicitly, multiple different characteristics at once. In the next chapter, we introduce graphlets that appear to have exactly those characteristics.

2. Graphlets

Comparing two large-scale networks quickly becomes computationally untenable[1]. For this reason, researchers devised tools for characterizing properties of a network that can be viewed as heuristics in the process of comparing two networks Boccaletti et al. [2006] (as described in section 1.3). Generally, these heuristics focus on either global properties – such as the diameter of a network, clustering, or average path length – or local properties. The earliest notable local heuristic harnessing *network motifs* was proposed by Milo et al. (2002).[2] They focus on "patterns of interconnections occurring in complex networks at numbers that are significantly higher than those in randomized networks" (2002, p.1). Without going into the technical details, network motifs can only be partial subgraphs and are dependent on the random model used for motif detection in networks, which, in real networks, may be misleading Artzy-Randrup et al. [2004]. Inspired by network motifs but aware of their shortcomings, Pržulj (2004b) proposed *graphlets* that focus only on induced subgraphs and are independent of a null model, making them easier to work with and more flexible.

Alternatively, instead of approaching graphlets through network motifs in the whole network – as mentioned above – we can perceive them as a generalization of degree distribution (see definition 4), thus motivating their study as an exploration of the neighborhood of vertices, from a different perspective.

Thanks to this, graphlets are commonly used to describe the local topology of networks Espejo et al. [2020], Milenković et al. [2009], Hulovatyy et al. [2015].

In this chapter, we provide a formal definition of graphlets and related concepts, review existing literature on the properties of graphlets, and present new observations.

2.1 What are graphlets

Graphlets were first introduced by Pržulj as "a connected network with a small number of nodes[3]" (Pržulj et al. [2004b]). We failed to find a formal definition of graphlet in literature. In this section, we lay out one a possible definition of graphlets.

Here, we use the following definition – graphlet is an ordered pair $(G, b(G))$ where G is a graph and b is a function that assigns to each G certain index (for small values this is determined by Figure 2.1) and the index of any graph on n vertices is less than the index of any graph on n' vertices if $n < n'$. Concretely:

Definition 1. *(Graphlet G_i^n) Let $\mathcal{G}^n$ be set of all connected non-isomorphic graphs on $n \geq 2$ vertices. Let $\mathcal{G} = \bigcup_{n \in \mathbb{N}} \mathcal{G}^n$ be the union of all these sets. Let us consider*

[1] Although we have a quasipolynomial algorithm for isomorphism testing (Babai [2016]), in the case of large graphs, that are common in networks science, remains too computationally expensive.

[2] The reason why we explicitly mention network motifs, besides historical context and motivation, is that graphlets and motifs are occasionally used interchangeably, and we want to make a clear distinction between the two concepts.

[3] Nodes is a term commonly used in network science for the concept of vertices in graph theory

a bijection $b : \mathcal{G} \mapsto \mathbb{N}$, such that 1) for $n \leq 5$, $b\restriction\mathcal{G}^n$ is defined as in Figure 2.1 below and 2) $\forall_{1<m<m';m,m'\in N}\forall G \in \mathcal{G}^m \forall G' \in \mathcal{G}^{m'} : b(G) < b(G')$.

Graphlet is a pair $(G, b(G))$ where $G \in \mathcal{G}$ is a connected graph and $b(G)$ is its index (natural number) assigned by the mapping b above. We denote this pair by G_i^n in the case when n is the number of vertices of the the graph G and $i = b(G)$.

We occasionally simplify our notation. When referring to a specific graphlet, G_i^n, we can omit the upper index indicating the number of vertices on which it is defined, as this number is uniquely determined by the index (because b is a bijection). Similarly, when referring to a general graphlet on n vertices, we use G^n without specifying its lower index.

By part 1) of definition 1, whenever we mention G_i with $i \leq 29$ or g_i with $i \leq 72$, we are using the convention from Figure 2.1. If referring to specific larger graphlets, the ordering should be explicitly specified.

The bijection b defines a total ordering of the set $\mathcal{G}$. For each $n \geq 2$, we will denote by $\mathbf{G}_n$ the restriction of this total order to $\mathcal{G}_n$.

Further, for each graphlet, we can consider automorphism orbits. The information about the neighborhood of a vertex then becomes much richer – we not only know in which graphlets a vertex participates, but also what is its position in the graphlet. This extension was introduced by Pržulj (2007). We use the following definition where graphlet orbits are defined using just the undelying graph structure of the graphlet, while their ordering refines the total ordering induced by the bijection b:

Definition 2. *(Graphlet orbit g_k) Let us fix a graphlet $G_i = (V_i, E_i)$ and its automorphism group $Aut(G_i)$. Orbit of a vertex $v \in V_i$ is $Orb(v) = \{u \in V_i | u = g(v)$ for some $g \in Aut(G_i)\}$. The equivalence classes of $=_{Orb(v)}$ are called* graphlet orbits*, their set is denoted by $Orb(G_i)$. We fix a one-to-one maping $d_i : Orb(G_i) \mapsto \mathbb{N}$. For $o \in Orb(G_i)$, we denote by $o_{i,j}$ the pair $(i, d_i(o))$.*

We define a total ordering of the set of all orbits $\mathcal{O} = \bigcup_{i\in\mathbb{N}} Orb(G_i)$ using the lexicographic ordering as follows: $o < o'$ whenever $o \in Orb(G_i), o' \in Orb(G_{i'})$, and either $i < i'$ or $i = i'$ and $d_i(o) < d_{i'}(o')$. The set of all graphlet orbits with this total lexicographic ordering is isomorphic, via an order isomorphism δ, to $\mathbb{N}$ with its natural ordering. We will moreover require the ordering of graphlet orbits to be compatible with Figure 2.1 below.

A graphlet orbital is a pair $(o, \delta(o))$ where $o \in Orb(G_i)$ and $\delta(o) = k$ is its index assigned by the isomorphism δ above. We denote this pair by g_k.

The ordering of graphlets and their orbits are based on a convention described in Figure 2.1 for graphlets on less than 6 vertices. For up to 5 vertices, there are exactly 30 graphlets with 73 different orbits.

When we refer to the position of v in a graphlet, we say that it *touches* a certain graphlet at a certain position. More concretely:

Definition 3. *(touches $g_j(G)\tilde{v}$) Let us have a graph $G = (V, E)$ and an induced subgraph $H = (V', E')$ such that H is isomorphic to the underlying graph of a graphlet G_i by an isomorphism f. If $v \in V'$ is such that $f(v) \in g_k$ for some orbit g_k of G_i, we say that v touches G_i at orbit g_k. Since G_i is uniquely determined by g_k, we can denote the relation of touching without G_i as $g_j(G)\tilde{v}$.*

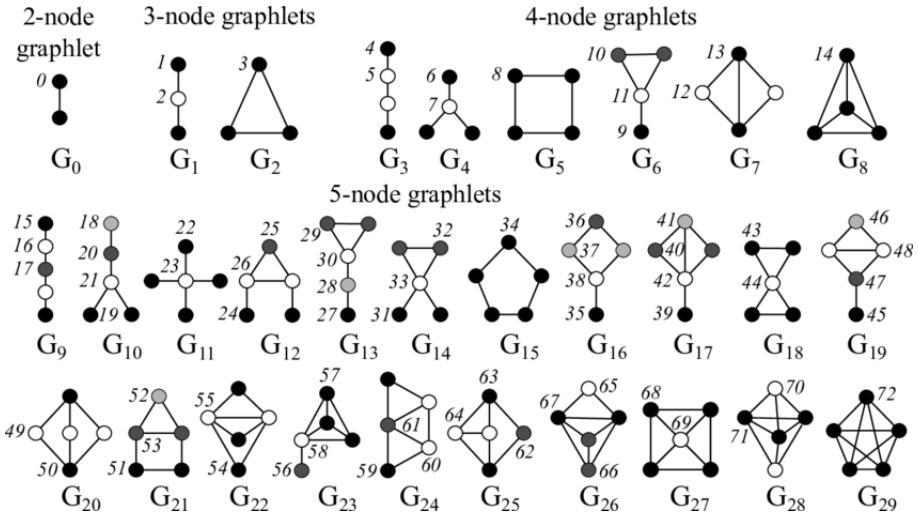


Figure 2.1: All possible graphlets for $n \in \{2, 3, 4, 5\}$. The ordering of graphlets is determined by G_i (below each graph). Graphlet orbitals are indicated by numbers by individual vertices in graphlets (for each graphlet, each orbital has its color). Orbit k than corresponds to g_k (from definition 2). For example, in G_1, we have orbitals g_1 (ending of a path) and g_2 (middle vertex). source of the image: Pržulj [2007]

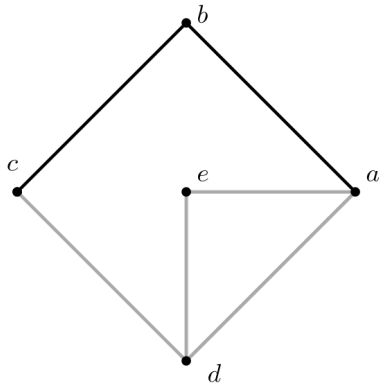


Figure 2.2: An example of vertex a touching G_1 (in bold) at g_1.

For example, consider the graph, G, in Figure 2.2 where vertex a touches G_1 at g_1, $g_1(G)\tilde{v}$:

If we consider a graph G and fix a vertex v, we can consider all graphlets up to a certain size in the neighborhood of v that v touches. We can summarise the information into a *graphlet degree vector*.

Definition 4. *(Graphlet degree vector $\{p, \ldots, q\}$-$gdd_G(v)$) Let us have a graph G on n vertices, the set of all graphlets induced on $2 \leq p \leq q < n$ vertices $\mathbf{G}_{\{p,\ldots,q\}} = \{\mathbf{G}_i | i \in \{p, \ldots, q\}\}$ and the set of orbits in these graphlets $Orb = \{g_k | \exists G_j \in \mathbf{G}_{\{p,\ldots,q\}} \exists v : v$ touches G_j at $g_k\}$, ordered as in definition 2.*

Let us denote by $n_i(G_j)_v$ the number of times that $g_i(G)\tilde{v}$ occurs, that is, the number of induced subgraphs H of G such that $v \in H$ and H is isomorphic to the underlying graph of the graphlet G_j whose orbit is g_k by an isomorphism f such that $f(v) \in g_k$.Graphlet degree vector is a vector whose values correspond to $n_i(G_j)_v$ for all possible orbitals in graphlets of sizes $\{p, \ldots, q\}$. Formally,

$\{p, \ldots, q\}$-$gdd_G(v) = (n_i(G)_v | g_i \in Orb)$.

For example for $\{2,3,4,5\}$-$\text{gdd}_G(v) = (n_0(G)_v, n_1(G)_v, \ldots n_{72}(G)_v)$.
For ilustration consider the graph G in Figure 2.2 and the vertex a. The graphlet degree vector is

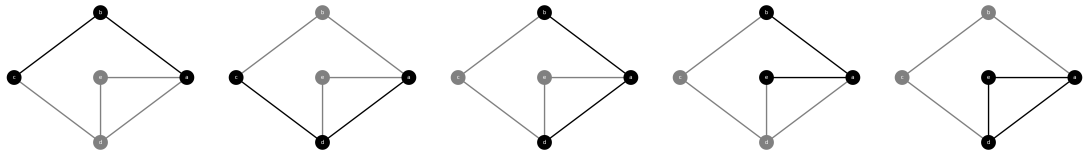
$$\{2,3\}\text{-gdd}_G(a) = (3,2,2,1) \tag{2.1}$$

if we present it in a table with comments about graphlet orbits for improved readability, we get

	g_0	g_1	g_2	g_3
$\{2,3\}$-$\text{gdd}_G(a)$	3	2	2	1

We get this because: There are three graphlets on $\{2,3\}$ vertices (G_0, G_1 and G_2) and four different orbitals that yield $Orb = (g_0, g_1, g_2, g_3)$. Vertex a touches G_0 as g_0 in G ($g_0(G)\tilde{a}$) three times which implies that $n_0(G)_a = 3$. Similarly, a touches G_1 twice as g_1 (graphs a and b in Figure 2.3) and twice as g_2 (graphs c and d in Figure 2.3) yielding $n_1(G)_a = 2 = n_2(G)_a$. Finally, a touches G_3 as g_3 in G ($g_3(G)\tilde{a}$, graph e in Figure 2.3) once which implies that $n_3(G)_a = 1$. This gives us:

$$\{2,3\}\text{-gdd}_G(a) = (n_0(G)_a, n_1(G)_a, n_2(G)_a, n_3(G)_a) = (3,2,2,1) \tag{2.2}$$



(a) (b) (c) (d) (e)

Figure 2.3: All 3-graphlets that vertex a touches in G

From this, it should become clear why the graphlet degree vector is sometimes considered a generalization of the vertex degree. We investigate the neighborhood of a vertex v up to a certain size and observe the topology described by the induced subgraphs that v touches. Furthermore, as a matter of fact, the vertex degree corresponds to $\{2\}$-$\text{gdd}_G(v)$. This is also why our notation for the graphlet degree distribution is inspired by the degree distribution, $deg_G(v)$.

Finally, if we consider graphlet degree vectors for all vertices in V from G, we get *graphlet degree distribution.*

Definition 5. *(Graphlet degree distribution ($\{p, \ldots, q\}$-gdd_G)) Graphlet degree distribution is* $\{p, \ldots, q\}$-$gdd_G = (\{p, \ldots, q\}\text{-}gdd_G(v) | v \in V)$

Graphlet degree distribution is usually presented in the form of a matrix, which is a convention that we uphold in this text. Furthermore, if we consider the entire graphlet degree distribution of graphlets on $p, \ldots, q$ vertices for a specific graph G, we use the notation $\{p, \ldots, q\}$-gdd(G). When the graph in question is

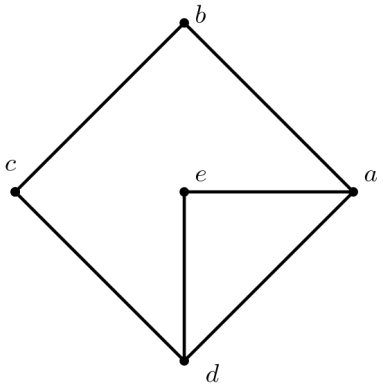


Figure 2.4: Graph on which we count graphlets for all vertices.

clear from context, we resort to simplified notation $\{p, \ldots, q\}$-gdd. For the graph G (shown in Figure 2.4), we obtain the following graphlet degree distribution:

$$\{2,3\}\text{-gdd}_G = \begin{pmatrix} 3 & 2 & 2 & 1 \\ 2 & 3 & 1 & 0 \\ 2 & 3 & 1 & 0 \\ 3 & 2 & 2 & 1 \\ 2 & 2 & 0 & 1 \end{pmatrix} \tag{2.3}$$

and in the form of table for better readability

	g_0	g_1	g_2	g_3
$\{2,3\}$-gdd$_G(a)$	3	2	2	1
$\{2,3\}$-gdd$_G(b)$	2	3	1	0
$\{2,3\}$-gdd$_G(c)$	2	3	1	0
$\{2,3\}$-gdd$_G(d)$	3	2	2	1
$\{2,3\}$-gdd$_G(e)$	2	2	0	1

There are two things to mention:

Remark. The initial definition of graphlets by Pržulj (Pržulj et al. [2004b]) did not consider orbitals but only entire graphlets that v touches. Therefore, the entire graphlet degree distribution also consisted only of counts of different graphlets that v touched. This definition is occasionally used interchangeably with our definition of graphlets and is at the heart of much misunderstanding in discussions about graphlets (compare Aziz et al. [2020], Zhang et al. [2013] and Pržulj [2007]). When we refer to it later on in the text, we use the notation $\{p, \ldots, q\}$-gdd$^\circ$.

Remark. The concept of graphlets was extended beyond the aforementioned definition – from vertices to edges (Solava et al. [2012]), directed graphs (Sarajlić et al. [2016]), and temporal graphs (Yoon et al. [2023]). However, these extensions are beyond the scope of this text.

2.2 Existing results

Despite their widespread usage, we found little theoretical treatment of the properties of graphlets. Furthermore, most of the existing results and observations

do not stem from an interest in graphlets on their own but rather from their applications, which in turn shed some light on their theoretical properties. In this section, we summarize a selection of approaches and results from existing literature that utilize graphlets to uncover their properties beyond mere application. By doing so, we simultaneously examine the existing literature for theoretical results and further motivate the study of graphlets based on their application.

2.2.1 Linear dependence between graphlets

Some graphlet orbits are linearly interdependent.

Calculating $\{p, \ldots, q\}$-gdd$_G$ by brute force is computationally extremely demanding, even for small values of q. If we fix a vertex v, we have to find all the graphlets spanned by neighbors up to a distance of q, establish their isomorphic classes, and determine the orbit of v in these graphlets. In the search for a more efficient graphlet counting algorithm, researchers started noticing interdependencies between the counted graphlets.

For example, if we denote $c(u, v)$ as the number of vertices that are connected to both u and v, denote $n_j(G)_v$ as the number of times that $g_j(G)\tilde{v}$, number of times v touches a graphlet as g_j and consider a graph G, we can deduce the following about G_1, a 3-path, and G_2, a 3-cycle:

$$2n_2(G)_v = \sum_{u \in N_G(v)} c(v, u) \tag{2.4}$$

$$n_1(G)_v = \sum_{u \in N_G(v)} deg(u) - 1 - c(v, u) = \sum_{u \in N_G(v)} n_0(G)_u - 1 - c(v, u) \tag{2.5}$$

This follows from the observation that if we fix an edge vu between a vertex v and its neighbor, the number of triangles (3-cycles), denoted as G_2, that the edge vu touches depends on the number of neighbors of u that are also neighbors of v, which, by definition, corresponds to $c(v, u)$. Therefore, if we calculate this for every neighbor of v, we obtain $2n_2(G)_v$ because we count the same triangle twice for the two neighbors of v that participate in it. By the same logic, the number of 3-paths, denoted as G_1, that v touches, denoted as g_1, depends on the number of vertices that are connected to u, a neighbor of u, but not to v. This corresponds to $deg(u) - 1 - c(v, u)$, as we need to subtract one for the edge vu.

Likewise, if we focus on the graphlets g_9 and g_{12}, we can analyze their occurrence, as noted by Hocevar and Demesar (2014), using Figure 2.5. Nodes on x, y, and z induce G_1, and we will be adding a fourth vertex, z. The number of vertices connected to both y and z is, by definition, $c(y, z)$ (represented by w_1, w_2, and w_3 in the figure). If this fourth vertex, w, is connected to x, it touches G_7 in g_{12} (as with w_3). If it is not connected to x, x touches G_6 in g_9 (as with w_1 and w_2).

Since all $c(y, z)$ are either in G_6 or in G_7, it must hold that for every specific trio of x, y, z, the orbits of x give us $n_9(G)_x + n_{12}(G)_x = c(y, z)$. Therefore, if we sum over all possible trios that form G_1, where x is at g_1, and take into consideration the repeated selection of concrete trios (every trio will be selected twice in our counting), we obtain the following for a specific vertex x:

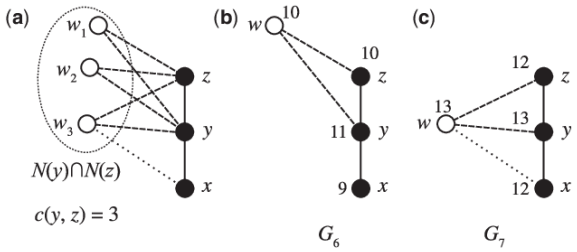


Figure 2.5: Relation between orbits g_9 and g_{12}. Filled edges are in the G^3 being extended. Dashed edges exist by definition and are optional – their presence makes the resulting G^4 isomprphic to G_6 or G_7. source of the image and the example: Hočevar and Demšar [2014]

$$2n_9(G)_x + 2n_{12}(G)_x = \sum_{\substack{y,z:x,z\in N_G(y)\\ G(x,y,z)\cong G_1}} c(y,z) \tag{2.6}$$

Hocevar and Demesar (2014) sucesfully constructed sets of 57 equations relating 58 orbits of graphlets on 5 vertices and the counts of neighboring pairs, triples and quadruplets under certain conditions of isomorphism.

These approaches that interrelate graphlet orbits together make the basis for most effective algorithms for precise enumeration of graphles and their orbits. For a thorough discussion of this approach and its usage, see Ribero et al. (2021).

2.2.2 Redundant graphlets and graphlet dependencies

> Interdependence of graphlets can be exploited in the effort to uncover organizational principles of networks.

In a similar vein of thinking, Yaveroğlu et al. (2014) identify direct interdependence between graphlet counts and subsequently exploit them for the analysis of networks. First, they identify redundant graphlet orbits, which are graphlets that can be linearly determined by the remaining graphlets. For example, if we fix a vertex v and consider its $g_0(G)_v$, we can observe that every pair of its neighbors is either in G_1 or G_2, which v touches as either g_2 or g_3. From this observation, we can deduce the relationship $\binom{n_0(G)_v}{2} = n_2(G)_v + n_3(G)_v$, which means that if we know either two, we can calculate the third. Yaveroğlu et al. follow through with this logic and identify 11 non-redundant graphlet orbits among the 15 possible ones in graphlets on 4 vertices. Let us call the set of these non-redundant 11 orbits R_{11}.

This enables Yaveroğlu et al. to develop a system for tracing structural behavior of concrete networks. To achieve this, they take the only the 11 orbits, selected 11 columns by the conventional ordering, of R_{11} from $\{2,3,4\}$-gdd, matrix of n rows and 11 columns, and compute Spearman's Correlation coefficients between all pairs of columns in this (n x 11) matrix and present them in an (11 x 11) symmetric matrix that they call *the Graphlet Correlation Matrix* (GCM). By minimizing structural dependencies between graphlets through the removal of redundant orbits, they are able to determine dependencies between graphlets

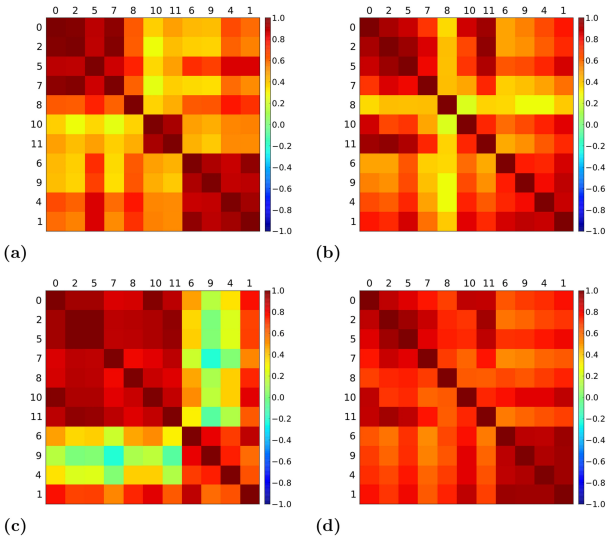


Figure 2.6: Graphlet Correlation Matrix calculated for two models of networks (mentioned in section 1.3.2): a) a scale-free Barabasi-Albert network with 500 nodes and 1% edge-density; b) a geometric random network of the same size and density; and two real-world networks: c) the world trade network of 2010; d) the human metabolic network. source of the image: Yaveroğlu et al. [2014]

that occur in the data and shed a novel light on the organizational principles of networks.

They apply this matrix to data about trading relations between a group of countries in 2010, analyze them using the GCM, and determine the function of certain groups of countries in the trading relations. The application of the GCM on real-world data, as well as classical models in network science [4], can be seen in Figure 2.6. For a thorough discussion of the technique and analysis, see Yaveroğlu et al. (2014).

2.2.3 Graph isomorphism and graphlet kernel

> Graphlets are used for isomorphism testing and are tied to the reconstruction conjecture.

Graph isomorphism is an important and widely studied open problem in graph theory (Kobler et al. [2012]) – given two graphs, G and H, can we determine if they are isomorphic, i.e., $G \cong H$? It has important practical ramifications in diverse fields of research, and as such, several approaches have been suggested for at least partial solutions. One well-received approach, proposed by Shervashidze et al. (2009), is the computationally efficient and well-performing *graphlet kernel* method. In this method, we first compute the number of graphlets on $2, 3, 4, 5$ vertices subinduced in the compared graphs [5], normalize their count by the total number of encountered graphlets, and calculate the transpose product between

[4] We will discuss models further in Chapter 2.

[5] It should be noted that, in contrast to our definition in this text, graphlets in the work by Shervashidze et al. refer to the frequency of occurrence of non-isomorphic connected graphs on $2, 3, 4, 5$ vertices throughout the whole graph, which results in a frequency vector.

the graphlet counts of the two compared graphs. This method is both well-performing and computationally efficient, successfully balancing out both aspects of isomorphism testing (Shervashidze et al. [2009]).

A key question in this approach is the chance of correctly identifying the isomorphism between two graphs. First, the authors resort to analytical results and draw connections between subgraph enumeration and the reconstruction conjecture, which states that every graph is uniquely determined by its subgraphs. Concretely,

Theorem 1. *(Reconstruction conjecture (Kelly [1957])) Let $G = (V, E)$ and $G' = (V', E')$ be two graphs of size $n > 2$, $\forall v \in V$, let G_v be a node deleted subgraph of G and $\forall v' \in V'$ let $G'_{v'}$ be a node deteled subgraph of G. Let $g : V \mapsto V'$ be an isomorphism function such that G_v is isomorphic to $G'_{g(v)}$ $\forall v \in V$. Then G is isomorphic to G'.*

They proceed to link reconstruction conjecture with their proposed kernel and to the computational and probabilistic aspects of these considerations, description of which is beyond the scope of this thesis. The link itself is nevertheless important and is further discussed in subsection 2.3.2

Besides the aforementioned three results, graphlets have been used in the field of Graph Neural Networks (GNN for short) as a way to capture and understand structural characteristics of nodes and their neighborhood on graph-structured data (Morris et al. [2019]). In this way, GNNs can learn representations that encode the local graph topology, enabling them to capture fine-grained structural information which is otherwise hard to trace (Guo et al. [2019]). Nevertheless, within the field of GNN, the definition and approach to graphlets substantially differ from how we treat them in this text – their definition puts emphasis on the concept of substructures and, thus, results are often hard to translate to our research (notable exception is harnessed in Section 2.3.2). For this reason, we do not explore this topic further. For more information, see Bronstein et al. (2017).

2.3 Exploring characteristic of graphlet distribution

Based on the previous section, one can argue that graphlets, as well as subgraph counting more generally, are encountered from different perspectives in a variety of research fields. Nonetheless, interest in their theoretical properties is often peripheral, and existing results are scattered across disciplines. In this section, we aim to address this by constructing an admittedly simplistic framework for studying graphlets and exploring their expressive power. Results in this section are original unless explicitly stated otherwise.

The key questions we seek to answer are: What can knowledge about the presence of graphlets in a graph tell us about the graph itself? In other words, what is the expressive power of $\{p, \ldots, q\}$-gdd? Additionally, what can we deduce about the graphlets in a graph given certain information about the graph, such as its class?

These questions, along with the inquiries posed by researchers in Section 2.2, all revolve around the relationship between a graph and its graphlet degree distribution. We can think of this relationship as a mapping between the

set of non-isomorphic connected graphs on n vertices and the set of all possible $\{2,\dots,\gamma\}$-gdd, where $\gamma \leq n$.

Before we do that, we have to establish what are the possible values of a $\{2,\dots,\gamma\}$-gdd from a graph on n vertices. Let us denote the largest value of $\{2,\dots,\gamma\}$-gdd$_G$ for a graph G on n vertices as $\overline{\{2,\dots,\gamma\}\text{-gdd}(G)}$, concretely:

$$\overline{\{2,\dots,\gamma\}\text{-gdd}(G)} = \max_{i,j}\{2,\dots,\gamma\} - gdd(G)_{i,j} \tag{2.7}$$

The maximal possible value of $\overline{\{2,\dots,\gamma\}\text{-gdd}(G)}$ is reached in the complete graph.

Lemma 2. *Let $\mathcal{G}_n$ be the set of all non-isomorphic connected graphs on n vertices. Then $max_{G\in\mathcal{G}_n}\overline{\{2,\dots,\gamma\}\text{-}gdd(G)}$ is $\binom{n}{\gamma}$, if $\gamma \leq \frac{n}{2}$, and $\binom{n}{\lfloor\frac{n}{2}\rfloor}$, if $\gamma > \frac{n}{2}$, and it occurs when the graph is complete, C_n.*

Proof. Let us have a graph G and fixed vertex v. When we look into the neighborhood of v intending to maximalize one value in $\{2,\dots,\gamma\}$-gdd$_G(v)$, we would like this vertex v to participate as many times as possible in one concrete graphlet and we want v to touch this graphlet at one specific orbit. This is exactly the case for a complete graph – μ-neighborhood of v for $\mu \leq \gamma$ always contains all vertices and selection of any μ vertices neighboring v gives us a complete graph on these vertices which means that we have $\binom{n}{\mu}$ different graphlets. Now we only need to select μ such that the number of graphlets is maximalized. This happens at $\mu = \lfloor\frac{n}{2}\rfloor$ since that is where the function $\binom{n}{\mu}$ for fixed n attains its maximum value on $\mathbb{N}$ and the function is monotonically increasing until that point. This gives us the statement. □

The minimal maximum value of $\overline{\{2,\dots,\gamma\}\text{-gdd}(G)}$ is

Lemma 3. *Let $\mathcal{G}_n$ be the set of all non-isomorphic connected graphs on n vertices. The minimal maximal possible value in $\overline{\{2,\dots,\gamma\}\text{-}gdd(G)}$, i.e. $min_{G\in\mathcal{G}_n}\overline{\{2,\dots,\gamma\}\text{-}gdd(G)}$, for any graph on $n \geq 3$ vertices is 2 and it occurs when the graph is a path or a cycle.*

Proof. Every induced subgraph of a path or a cycle is a path. If we fix a vertex v, the only graphlets in a path that have nonzero entries will be paths shorter than γ. In a given path, the number of mutually automorphic vertices is, at most, 2 since the position of a vertex is determined by its distance from both ends of the path. For this reason, the number of times v touches any orbit is at most 2 in a path. This value is necessarily minimal, it cannot be smaller, since we consider only connected graphs.

Now we show that $min_{G\in\mathcal{G}_n}\overline{\{2,\dots,\gamma\}\text{-gdd}(G)} = 2$ happens exactly when G is a path or a cycle. This follows from the fact that if the given graph G included any branching (shown in Figure 2.7), the vertex v in the middle of the branching would touch graphs G_1 at orbit g_2 three times (in the paths $u'vu'', u''vu'''$ and $u'vu'''$) which would lead to higher $\overline{\{2,\dots,\gamma\}\text{-gdd}(G)}$ than in a path or a cycle. The only graphs that do not include branching are paths and cycles which concluded the proof

□

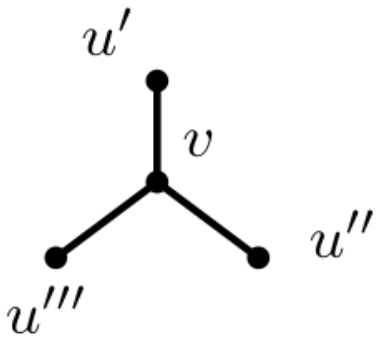


Figure 2.7: example of a branching

Now, we can return to the idea of mapping between graphs and possible graphlet distributions and define the following:

Definition 6. *(graphlet mapping ($\mathcal{G}_\gamma^n$)) Let $2 < n \in \mathbb{N}$ and $\gamma < n$. Let $\mathcal{G}_n$ be a set of non-isomorphic labeled connected graphs on n vertices. Let GDD_γ^n be a set of all matrices of dimensions (nx k) where k is the number of orbits in all graphlets on $\{2, \ldots, \gamma\}$ vertices. The values of, $(GDD_\gamma^n)_{i,j}$, are bounded by $0 \leq (GDD_\gamma^n)_{i,j} < \binom{n}{\lfloor \frac{n}{2} \rfloor}$ and the rows correspond to vertices in lexicographical order. Graphlet mapping, $\mathcal{G}_\gamma^n : \mathcal{G}_n \mapsto GDD_\gamma^n$, is a function that maps a given graph on its graphlet distribution.*

We show that the definition is correct and make some observations about the mapping

Claim 4. *The mapping $\mathcal{G}_\gamma^n$ described in definition 6 is well-defined, in general, non-surjective and non-injective for any choice of $\gamma \leq n$.*

Proof. The number of possible graphs on n vertices is 2^n and the number of connected and mutually non-isomorphic graphs is smaller since we only add constraints. Therefore, the set of connected non-isomorphic graphs on n vertices is finite and the domain is well-defined. From lemma 2 and lemma 3, we get the bounds $2 \leq \overline{\{2, \ldots, \gamma\}\text{-gdd}(G)} \leq \binom{n}{\lfloor \frac{n}{2} \rfloor}$. Therefore, the range of the mapping is finite and well-defined. Further, since we use the labels to mark vertices and their corresponding counterparts in rows of GDD_γ^n, there is a unique projection for each G $\mathcal{G}_\gamma^n$. Therefore, the mapping is well-defined.

We can see that the mapping is not surjective by constructing a matrix that, although bounded and fitting criteria of the definition, contradicts internal dependencies of graphlets discussed in 2.2.2, e.g. we can fix a vertex v and the values of $n_2(G)_v$ and $n_3(G)_v$, but choose $n_0(G)_v$ so that $\binom{g_0(G)_v}{2} \neq g_2(G)_v + g_3(G)_v$ (using the result discussed in section 2.2.1).

The non-injectivity can be proved by the following counterexample: consider graph G and graph H on Figure 2.8. Since they both have the same degree distribution $(1 - 2 - 2 - 2 - 3)$, $\{2\}\text{-gdd}_G = \{2\}\text{-gdd}(H)$.

□

2.3.1 Graph to graphlet degree distribution

We can continue our exploration by tracing how $\mathcal{G}_\gamma^n$ behaves on certain types of graphs. This part of our exploration of the properties of graphlets should give us

(a) graph G (b) graph H

Figure 2.8: counter example to injectivity of $\mathcal{G}_v$

a better sense of what they capture and, as it turns out, establish some of their limitations. Some of the results below might seem inappropriately placed, but this section intends to get our hands on graphlets, play around with them, and, along the way, establish facts about graphlets that will be useful later on.

Lemma 5. *Let v and u be two vertices in G such that there exists an automorphism that maps v on u, i.e. they are contained in the same orbit. Then $\forall \gamma \leq n : \{2, \ldots, \gamma\}\text{-}gdd_G(v) = \{2, \ldots, \gamma\}\text{-}gdd_G(u)$.*

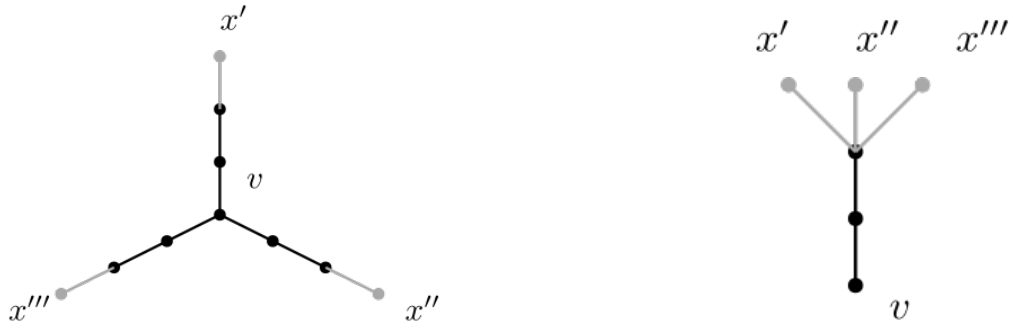
Proof. By contradiction, let i be the index of the graphlet for which $n_i(G)_v \neq n_i(G)_u$ (if there is no such index, we are done). Consider all the subsets of the neighborhood of u and v that induce a graphlet such that u and v are touching it at orbital g_i. Since the number of such graphlets differs for u and v, there must be an edge in the neighborhood of u (wlog) that is not present in the neighborhood of v. But than, there cannot exist an automorphism that maps u on v. This gives us the contradiction. □

Theorem 6. *Consider a connected graph, G, on n vertices and its graphlet degree distribution of graphlets up to the size of $n-1$, $\{2, \ldots, n-1\}$-gdd. If G is 2-vertex-connected, then we can uniquely determine $\{2, \ldots, n-2\}$-gdd from $\{n-1\}$-gdd.*

Proof. Let us have a graph G and fix a vertex v. If we know $\{n-1\}\text{-gdd}_G(v)$ and want to correctly determine $\{2, \ldots, n-2\}\text{-gdd}_G(v)$, we have to ensure two things : 1. which graphlets on $l \in \{2, \ldots n-2\}$ vertices does v touch and 2. that the number of times that we take these graphlets into account truly corresponds the to number of times v touches them.

To approach the first, we can make a small observation – every graphlet on $l \in \{2, \ldots n-2\}$ vertices that v touches must be induced in a graphlet on $n-1$ vertices that v touches. If it was not the case and we had a graphlet on l vertices that v touches, call it G_l, but is not induced in any graphlet on $l+1$ vertices that touch v, we could find a vertex u connected to G_l and, thus, a graphlet on $l+1$ vertices that induce G_l. From this, it follows that in order to establish which graphlets on $l \leq n-1$ vertices v touch, it is sufficient to investigate induced graphlets in every graphlet on $n-1$ vertices that v touches. From this it follows that from $\{n-1\}\text{-gdd}_G(v)$, we can find out which values in $\{2, \ldots, n-2\}\text{-gdd}_G(v)$ are nonzero, i.e. which graphlets on $l \in \{2, \ldots, n-2\}$ vertices can we expect, and where does v touch them.

To approach the second, we must find out how many times each of these graphlets that v touches occur in the graph since certain graphlets can be present in some of the larger graphlets but not in others and we want to include each exactly once. Illustration of the problem can be seen in the following Figure 2.9:



(a) graph G (b) graph H

Figure 2.9: The graphs G and H demonstrate a problematic case when induced graphlet (in these cases of P_3, path on 3 vertices, rooted in v) can be counted once (as in the case of H where paths from v to x', x'' and x''' all induce the same specimen) or three times (as in the case of G where paths from v to x', x'' and x''' induce different specimens of P_3). We show the case on smaller graphlets than $n-1$ for illustration.

To resolve this, let us consider a graphlet G_i on $k < n-1$ vertices that v touches at orbit g_j. From 2-vertex-connectedness, we know that v touches exactly $n-1$ graphlets on $n-1$ vertices – we can remove any vertex but v from the G and get a graphlet on $n-1$ vertices that v touches. If the graph was not 2-vertex-connected, upon removing a certain vertex x, the graph would split into two disconnected components, v would be in one of them and, thus, there would have to be less than $n-1$ graphlets on $n-1$ vertices.

Therefore, if we have a graphlet G_i on k vertices touching v, there are exactly $n-k$ graphlets on $n-1$ vertices that v touches and that include G_i as an induced subgraph. Based on this observation we can establish the number of times a graphlet on k vertices that touch v at g_j occurs by summing all the occurrences in all graphlets on $n-1$ vertices that v touches – from 2-vertex-connectedness, there is exactly $n-1$ of them – and dividing the number by $n-k$ as each of these graphlets is accounted for in exactly $n-k$ graphlets on $n-1$ vertices (we can remove any vertex but those in the graphlet on k vertices, one of which is v, and there is $n-k$ of those). □

From this it follows that for a 2-vertex connected graph, the information included in $\{2,\ldots,n-1\}$-gdd$_G$ is the same as the information included in $\{n-1\}$-gdd$_G$. It should be noted that the process of calucating the rest of graphlet counts can be as difficult as computing the entire graphlet degree distribution since we need to take into account isomorphisms of all subinduced graphlets. Nevertheless, the result might be interesting from the perspective of expressivity of GNN.

Remark. The case when G is only 1-connected is more difficult. Even though we can establish the types of graphlets that are subinduced in the known graphlets

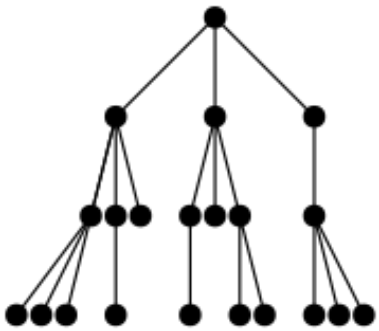


Figure 2.10: Example for graph G for computing nested degrees.

(e.g. those on $n-1$ vertices), by the same argument as given in the proof of Theorem 6, it is difficult to establish how many times is a graphlet induced. The reason for that is that we do not have $n-1$ graphlets on $n-1$ vertices – since the graph is not 2-vertex connected, there is at least one vertex that, upon removing, will split the graph into two disconnected components. Therefore, not all graphlets on less than $n-1$ vertices participate in exactly $n-1$ graphlets on $n-1$ vertices which makes counting difficult. An example of the problem can be seen in Figure 2.9. From this, it follows that in order to establish the number of times a graphlet is induced, we would have to aggregate information from multiple vertices at once, which, if possible, would require an exploration of a number of nontrivial combinatorial possibilities.

Now, we can focus on specific classes of graphs and see how graphlet behaves on those.

Let $N^{\mu}_{\text{deg}}(v)$ be the *nested degrees* of a vertex v to the depth of μ – by this, we mean nested listing of all degrees of a vertex and all of its neighbors up the distance of μ. For example in the graph in Figure 2.10, we get the nested degrees $N^{2}_{\text{deg}}(v) = (3 : (4 : ((4)(2)(1)))(4 : (2)(1)(3))(2 : (4)))$ – this stems from the fact that vertex v has three neighbors: 1) first has degree 4 and three children with degrees 4,2 and 1; 2) second has degree 4 and three children with degrees 2,1 and 3; 3) third has degree 2 and one child with degree 4. $N^{\mu}_{\text{deg}}(v)$ stops working in graphs where there are cycles – since it does not account for which vertices we visited, it can continue in cycles. Nevertheless, it is sufficient for the following claim:

Lemma 7. *Let us have a graph G. If G is a tree, than for each v its $\{2,\ldots,\mu\}$-gdd$_G(v)$ is determined by its $N^{\mu}_{deg}(v)$.*

Proof. Let us fix a vertex v and consider its $N^{\mu}_{\text{deg}}(v)$. Since the all subgraphs of a tree are trees, there are no cycles in the graph, we can reconstruct the whole neighborhood of v based on this sequence. Therefore, we have more than enough information for calculating $\{2,\ldots,\mu\}$-gdd$_G(v)$ since, by definition, all graphlets on μ vertices are in a distance of less than μ. □

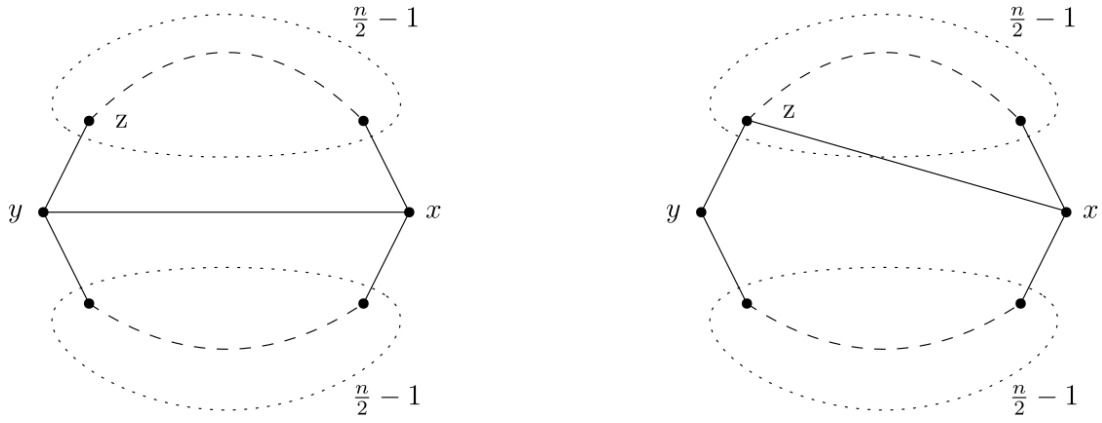
Theorem 8. *Every two regular graphs on n vertices, G and H, with grith, the length of the smallest induced cycle, of $\mu \leq n-1$ or smaller, have the same $\{2,\ldots,\mu\}$-gdd.*

Proof. Since there is no cycle of length μ or smaller, the only graphlets that can occur are acyclic and connected – trees. We can determine $\{2,\ldots,\mu\}$-gdd(G) and

$\{2, \ldots, \mu\}$-gdd(G). Since all vertices have the same degree, $\{2, \ldots, \mu\}$-gdd(G) is equal to $\{2, \ldots, \mu\}$-gdd(G) by lemma 7. □

From this, it follows, that graphlets are not particularly useful on graphs that have a local topology similar to trees, especially to those with uniform degree distribution.

Finally, consider the following two graphs:



(a) graph G : cycle on n vertices with additional edge (x, y)

(b) graph H : cycle on n vertices with additional edge (x, z)

Figure 2.11: Two graphs exposing limitations of graphlets

Lemma 9. *Graph G and H in Figure 2.11, have equal* $\{2, \ldots (\frac{n}{2} - 1)\} - GDD$.

Proof. The two graphs are nonisomorphic – the smallest induced cycle in G is on $\frac{n}{2} + 1$ vertices, whereas in H, it is on $\frac{n}{2}$. □

Which further strengthens the claim about the non-injectivity of $\mathcal{G}_\gamma^n$ in claim 4 for any $\gamma \leq \frac{n}{2} - 1$, instead of just the simple case of degree distribution.

To conclude, in this subsection, we have shown that larger graphlets can uniquely determine smaller ones under certain circumstances, contributing to the existing work on the interdependence of graphlets (see subsection 2.2.1). Furthermore, we have demonstrated that the amount of information that graphlets can provide about graphs with tree-like local topology is limited, especially when the degree distribution of the graph in question is relatively uniform.

Connection to Weisfeiler-Lehman isomorphism test

To demonstrate the potential strength of graphlets, we can contrast them with a well-explored area of research in theoretical computer science – the Weisfeiler-Lehman isomorphism test (WL for short). First, we describe what this test entails and its variants [6]. Then, we establish a link between the k-dim WL and graphlets. Finally, we harness some of the literature around the test to explore the strength of graphlets.

In the search for a fast test for isomorphism, Weisfeiler and Lehman (1968) proposed an algorithm that colors all vertices according to their degrees and then

[6] The notation and exposition of WL are based on Cai et al. (1992)

iteratively updates the coloring of all vertices. In each iteration, the label of every vertex is extended by the multiset of colors of its neighbors. Afterward, any two vertices have the same color if and only if their multiset is equal. The iteration stops when the labels stabilize, i.e. there is no difference in coloring between iterations. We refer to this test as the 1-dim Weisfeiler-Lehman test (1-dim WL for short), as it operates on sets of size 1, namely vertices and their degrees.

Although seemingly simple and despite some obvious drawbacks, such as the fact that 1-dim WL is clearly useless for regular graphs, it turns out that even 1-dim WL is surprisingly effective in identifying non-isomorphic graphs. Babai et al. (1980) proved that the probability that 1-dim WL produces a normal form of the graph, i.e., a coloring of the graph according to the orbits in the graph, for random graphs on n vertices is $1 - n^{-1/7}$. Additionally, Babai and Kučera (1979) showed that just two iterations of 1-dim WL result in a canonical form of a random graph with probability $1 - \exp(-cn)$, where c is a constant and n is the number of vertices in the graph in question.

As it turns out, graphlets behave differently than 1-dim WL. Consider the graphs in Figure 2.12:

(a) graph G (b) graph H

Figure 2.12: Two graphlets distinguishable by 2 iterations of 1-dim WL but by a $\{2,3,4\}$-gdd(v). The numbers to the right of the graphs indicate the sum of degrees of vertices in the given layer of the graphs.

The two graphs are distinguishable by 1-dim WL already after 2 iterations – the vertices in the distance 2 from v have different degrees, $(2-1-1-1-1)$ vs $(2-1-2-1)$ – whereas $\{2,3,4\}$-gdd$_G(v) = (4,6,4,0,6,18,6,4,0,0,0,0,0,0,0,0,0)$ $= \{2,3,4\}$-gdd$_H(v)$.

Remark. The example above suggests that distinguishing two graphs based on graphlet degree vector of individual vectors is not easy – although graphlets provide us with information about all the graphlets in a neighborhood of a given vertex, they overlap and, thus, two different graphs can provide equal graphlet distribution despite being different from the perspective of isomorphisms. Still, it is worthwhile to explore the links between WL and graphlet distributions since, if we managed to show that graphlets are at least as strong as 2 iterations of 1-dim WL, we could make use of the theorem of Babai and Kučera (1979) and show that graphlet degree distribution can be used to produce the canonical form of a random graph with probability $1 - \exp(-cn)$, where c is a constant and n is the number of vertices in the graph in question. Such a result would anchor the power of graphlets to distinguish non-isomorphic graphs.

The example in Figure 2.12 further suggests that in order to make such a link, we would have to consider the graphlet degree distribution as a whole rather than

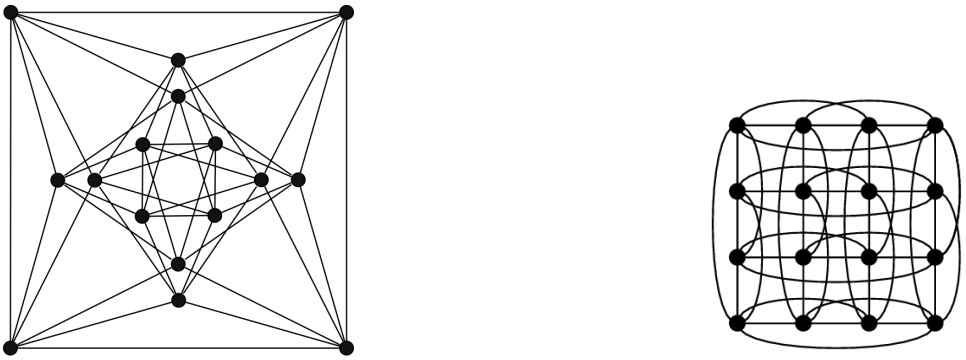
focusing on individual vertices.

Although from the above, it might appear that graphlets are not as powerful in isomorphism testing, in some cases, graphlets perform better in distinguishing different graphs. Take for example the following classical counterexample for 1-dim WL two C_3 and C_6[7] that are not distinguishable by 1-dim WL after any number of iterations whereas graphlets can easily tell them apart (it is sufficient to consider the total number of g_3, triangles, in $\{2,3\}$-gdd(G) and $\{2,3\}$-gdd(H)).

(a) graph G (b) graph H

Figure 2.13: Classical example of graphs indistinguishable by 1-dim WL but distinguishable by $\{2,3\}$-gdd

Similarly, if we consider the Shrikhande graph (Figure 2.14a) and the 4x4 rooks graph (Figure 2.14b) – they are both strongly regular graphs and are not distinguishable by 1-dim WL (Arvind et al. [2020]) [8]. Nevertheless, both of them can be distinguished by $\{2,3,4\}$-gdd



(a) Shrikhande graph, source of the image: Wikipedia [2023]

(b) 4x4 graph, source of the image: Abreu et al. [2021]

Figure 2.14: Shrikhande and 4x4 rooks graphs are not distinguishable by 1-dim WL but are distinguishable by $\{2,3,4\}$-gdd

Remark. All this makes it tempting to assert that graphlets might not be the best for isomorphism testing – especially when we try to distinguish two graphs

[7] To correctly utilize it in our framework where we consider graphlets only on connected graphs, we add additional vertex (gray edges in the image) that ensure connectedness but do not make graphs distinguishable by 1-dim WL – this was tested by hand but we omit the test for brevity.

[8] As a matter of fact, they are not distinguishable by 3-dim WL (Arvind et al. [2020]), but we have not yet defined this concept.

whose topology might be different but for every type of graphlet tested that might indicate the difference, there is another slight difference that balances out the change in graphlet counting [9] – what they are good for, on the other hand, is distinguishing the local topology of each vertex which might be stating the obvious when we know the definition of graphlets, but now we can contract this capacity with 1-dim WL. Despite the temptation, we should be wary of jumping to such rushed conclusions as they do not stand on much more than a gut feeling and a few, although illustrative, examples.

Following up on the exploration of the relation between graphlets and WL, there are further connections to be drawn. Nonetheless, that requires us to move from 1-dim WL to k-dim WL.

An extension of 1-dim WL is k-dim WL, introduced by Babai and Mathon (1980). This algorithm considers k-tuples of vertices instead of just the degrees of vertices. In this algorithm, we first assign a color to each vertex based on its isomorphism class [10]. The refinement step is then carried out as follows: given a color of $(u_1, \ldots, u_k)$, the refinement by the vertex v is the multiset of k-tuples of colors previously assigned to $(v, u_2, \ldots, u_k)$, $(u_1, v, \ldots, u_k)$, $\ldots$, $(u_1, \ldots, u_{k-1}, v)$.

Remark. When using this definition of k-dim WL, 1-dim WL is not the same as k-dim WL for $k = 1$. In fact, 1-dim WL is equivalent to k-dim WL where $k = 2$. We continue using 1-dim WL since that is the convention.

Higher dimensional WL is substantially more powerful than lower dimensional WL. For every $k > 2$, there exists a pair of graphs that cannot be distinguished by k-dim WL but can be distinguished by k+1-dim WL and vice versa (Cai et al. [1992]). However, as shown by Cai, Immerman, and Furer, who translated the problem into the language of first-order logic with counting (1992), and by Evdokimov and Ponomarenko, who approached it from the perspective of cellular algebras (Evdokimov and Ponomarenko [1999])[11], we need at least k-dim WL, where $k \in \Omega(n)$, in order to distinguish non-isomorphic graphs on n vertices.

What is the connection between k-dim WL and $\{2, \ldots, k\}$-gdd? It turns out that these concepts are related, as shown by Chen et al. (2020). This relation is especially clear in the case of $\{2, \ldots, k\}$-gdd$^\circ$, a simpler version of graphlet degree distribution where we do not consider orbitals and focus only on the entire graphlet that each vertex touches (see remark 2.1). The proof is inspired by Theorem 3.7 in the paper by Chen et al. (2020, p. 22) where they focused on counting patterns of k or less vertices by k-WL in the context of Graph Neural Networks.

Claim 10. *Let G be a graph. Given labels of k-tuples of vertices of graph G, we can reconstruct the entire $\{k\}$-gdd$^\circ$ of the graph G.*

Proof. Suppose that we have all possible k-tuples with information about the graph to which they are isomorphic – this corresponds to the initialization of

[9] Exactly this can be illustrated on the graphs in Figure 2.12 – when considering for example $g_1(v)$, we get $4 + 2 + 2 = 6 = 3 + 3$, a difference in topology on the neighboring vertex with degree 5 is balanced out by difference in topology by remaining neighbors.

[10] There is a number of different ways of initializing k-dim WL. For more examples see Cai et al. (1992)

[11] Both of these theories and proofs are nontrivial and go beyond the scope of this thesis. For more information, refer to the original papers.

k-dim WL (although with a slight shift in emphasis). We can consider all those k-tuples that correspond to graphs on k vertices. Since we have information about which ones of these graphlets are isomorphic, we can find corresponding graphlets. Finally, for each vertex in the graph, we can identify the graphlets that the vertex touches by the corresponding k-tuple including the vertex. In this way, we can establish all graphlets on k vertices in which each vertex participates – if this was not the case and there was an unaccounted-for graphlet, we could construct a k-tuple of its vertices and show a contradiction with the fact that we suppose all possible k-tuples. □

Clearly, this simple link works only during the initialisatin of k-dim WL since after that we lose information about concrete k-tuples accounted for in the process due to the recoloring after every iteration. Further, the link is not applicable for $\{k\}$-gdd k-dim WL does not take into account orbits.

The Theorem 10 remains useful since we can use the counterexamples constructed by Cai, Immerman, and Furer (1992) and Evdokimov and Ponomarenko (Evdokimov and Ponomarenko [1999]) and claim we need at least $\{k\}$-gdd$^\circ$ where $k \in \Omega(n)$ to distinguish such two graphs – probably, we need much larger graphlets since the counterexamples apply for unlimited iterations of k-dim WL whereas our statement works only with the initialization.

2.3.2 Graphlet degree distribution to graphs

Having coarsely explored what graphlet degree distributions can we expect from certain graphs, it is time to turn the question around and ask what graphs can we expect from a concrete graphlet distribution. This question is dependent on how large graphlets we take into consideration – if we consider $\mathcal{G}_n^n$ mapping, we have complete information about the graph encoded in the $\{2, \dots n\}$-gdd, and the mapping is bijective and, thus, invertible. Whereas, when we consider $\mathcal{G}_2^n$, we can expect all graphs that fulfill the given degree distribution, i.e. have the same $\{2\}$-gdd – an example of which can be seen in the proof of claim 4. In this section, we want to explore the question of what graphs can be reconstructed from a given $\{2, \dots, \gamma\}$-gdd. We first explore the question from below, for values of $\gamma = 2, 3$, and suggest algorithms that for a given graphlet degree distribution produce a correct graph if it exists. Later, we explore the question from above, for $\gamma = n - 1$, and link the question of reconstruction from graphlet degree distribution with reconstruction conjecture.

Reconstruction from below

When we are attempting to construct a graph from a graphlet degree distribution where only small graphlets are considered, there are two main things that we must ensure – each vertex v in the resulting graph G[12] must have correct $\{2, \dots \gamma\}$-gdd$_G(v)$ and the produced graph must be feasible – vertices must be connectable so that the final graph had desired graphlet degree distribution. These conditions turn out to be exceedingly difficult to keep for larger γ.

[12]We can focus on a concrete vertex v thanks to the fact that the mapping being explored, $\mathcal{G}_\gamma^n$, is defined on labeled graphs into labeled graphlet degree distributions.

In the following section, we suggest two algorithms for graph reconstruction from graphlet degree distribution for $\gamma = 2, 3$, which produce a correct graph if reconstruction is possible and announce when it is not.

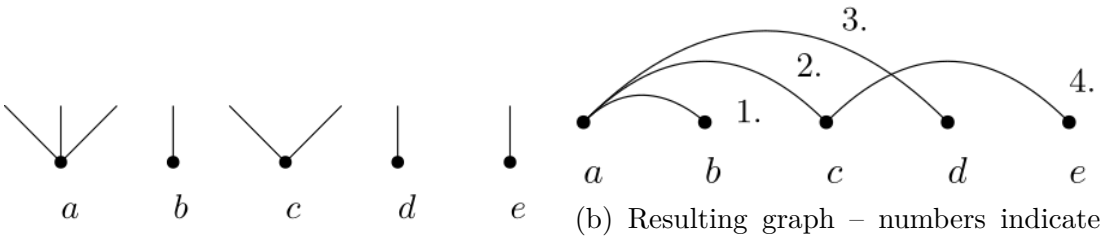
If we have a $\{2\}$-gdd and want to construct a graph with predetermined degree distribution, we can use the *configuration model* (Newman [2010]) which we call *2-configuration model* as it deals only with graphlets of size 2. The algorithm is presented as algorithm 2:

Algorithm 2 2-configuration model

Input: matrix $GDD \in \mathbb{N}^{n \times 1}$
Output: graph G with $\{2\}$-gdd$(G) = GDD$

start with a set of n vertices where n is the number of rows in GDD. For each vertex, create *stubs* for the degree it is supposed to have.
order all vertices with studs lexicographically in a line
while there is a vertex v with the smallest ordering number in the line with unconnected studs **do**
    for all unconnected stud s at vertex v **do**
        find another vertex u with unconnected stud s' s.t. $u \neq v$ and there is not yet an edge between the two vertices
        if such vertex exist **then**
            connect studs s and s'
        else
            algorithm failed
        end if
    end for
end while
return connected vertices

In the initial ordering of vertices with studs, the sequence of connecting and resulting graph can be seen in Figure 2.15



(a) Initial ordering of vertices with studs

(b) Resulting graph – numbers indicate the connecting process

Figure 2.15: Process of 2-configuration model on degree distribution (3-1-2-1-1)

Theorem 11. *Given a graphlet distribution* $\{2\}$*-gdd the 2-configuration model, if possible, outputs a graph with correct degree distribution, if it is not, it declares failure.*

Proof. Full proof can be found in Newman [2010]. Here, we present only a sketch of the proof. Clearly, in each iteration, the number total number of studs among all vertices is either reduced by 2, when two connectable studs are found in step

7., or the algorithm declares failure. From this, it follows that the number of studs must be even. Furthermore, if a graph is returned, it contains no self-loops or multi-edges – this is secured by the check-in step 5. Finally, since in each iteration, all unconnected studs are to the right in the line of the processed stud, we ensure that if a graph with desired degree distribution exists, it will be found. □

Drawing inspiration from from the 2-configuration model, we suggest a *3-configuration model*, which, if possible, constructs a graph from a given $\{2, 3\}$-gdd. Our approach, however, does not lead to as clean algorithm as does the algorithm for 2-configuration model since we must ensure many more conditions than just that there are not self-loops and multi-edges. As such, we have doubts of its applicability beyond theoretical discussion [13]. Hence, we refrain from providing an in-depth discussion of the algorithm but instead highlight the challenges that arise in the process of constructing a suitable graph and suggest possible ways to resolve them. Gradually, throughout the process, we assemble all the necessary components for the algorithm. In this manner, rather than achieving a provably well-behaving algorithm, we offer insight into the complexities of graph construction for a given graphlet degree distribution, which we consider more valuable.

The only graphlets on 3 vertices are a path on 3 vertices, G_1, and a triangle, G_2 see Figure 2.1 for a reminder of the convention. A straightforward approach, arising from the 2-configuration model, that comes to mind would be to take the studs determined by the degree of a given vertex, g_0, and categorize them into those studs that participate in a triangle and those that do not. The issue, nevertheless, is that we do not know in how many triangles studs of a given vertex participate – take for example the two graphs G and H illustrated in Figure 2.16. The vertex v has graphlet degree distribution $(n_0, n_1, n_2, g_3) = (4, \_, 4, 2)$ in both G and H. Note that we omit n_1 since it depends on the local structure and, thus, cannot be determined based on the 2-neighborhood of v alone – it is a problem to which we return shortly. For now we restrain our considerations to $n_0(v), n_2(v)$ and $n_3(v)$.

(a) graph G (b) graph H

Figure 2.16: Vertex v has graphlet degree distribution $(g_0, g_1, g_2, g_3) = (4, \_, 4, 2)$ in both G and H.

From this, we learn two things. First, information about participation in graphlets of size 3 gives us limited information as it does not allow us to distin-

[13] 2-configuration model is commonly used for testing community structrue of graphs, modeling of random networks or calculating modularity (see chapter 1 for better contextualization) thanks to its mathematically convenient properties. None of which we find in our 3-configuration model.

guish between these two cases in Figure 2.16. Second, when producing a graph, we have to allow for both of these cases.

To address this, we introduce the *triangleness* of a stud, the number of triangles in which it participates. Further we introduce *i-triangleness* of a vertex v, denoted as $\Delta_i(v)$ that indicated how many studs participate in i triangles. For example in G above, we have $\Delta_1(v) = 4$ and $\forall_{k\neq 1}\Delta_j(v) = 0$, since all studs participate in exactly one triangle, whereas in H, we have $\Delta_0(v) = 1$, $\Delta_1(v) = 2$, $\Delta_2(v) = 1$ and $\forall_{k\notin\{0,1,2\}}\Delta_j(v) = 0$. Since every triangle in which v participates consists of two studs and the total number of triangles must correspond to triangleness over all studs of v, we can see that:

$$g_3(v) = \sum_{i=0}^{(n-1)(n-2)} \frac{i(\Delta_i(v))}{2} \tag{2.8}$$

Here n is the total number of vertices that are supposed to be in the final graph, we need to account for up to $(n-1)(n-2)$ triangleness of a stud since that is in how many triangles a stud participates in the complete graph which gives us the largest graphlet count according to lemma 2.

In the case of graph G and H, we can see that this holds true since $(1*4)/2 = 2 = g_3(v) = 2 = (0*1)/2+(1*2)/2+(2*1)/2$. From this, we can further see that the number of possible stud trinaglenesses is tied to the number of decomposition of $g_3(v)$ by the formula 2.8.

Also, from the definition of $\Delta_i(v)$, we get the following

$$n_0(v) = \sum_{i=0}^{(n-1)(n-2)} \Delta_i(v) \tag{2.9}$$

Furthermore, if a stud participates in i triangles, there are exactly $n_0 - i - 1$ G_1s which v touches at g_2 (if we fix a stud, there is exactly $n_0(v) - i - 1$ studs that form a path together with v and the fixed stud). From this, we get:

$$n_2(v) = \sum_{i=0}^{(n-1)(n-2)} \Delta_i(v)(n_0(v) - i - 1) \tag{2.10}$$

From this, we obtain constraints on the possible values of $\Delta_i(v), \forall v, i$ which lead to feasible values of $n_0(v), n_2(v)$ and $n_3(v)$. Although these conditions are sufficient to ensure that any possible values of $\Delta_i(v), \forall v, i$ satisfying them lead to correct values of $n_0(v), n_2(v)$ and $n_3(v)$, they do not yet secure that such a graph is constructible. To make that happen, consider the procedure 3. If we have two studs, we can connect them by edge. Each edge has its "used-triangleness" which indicates in how many triangles it participates and that is bounded by the triangleness of the two studs that where connected for its formation.

The ilustration of the procedure is presented in Figure 2.17.

We do not claim that this procedure necessarily return all possible feasible graphs. Rather, we present it as a good starting point for further elaborations.

Finally, we turn to the question of g_1. We cannot ensure that we obtain the correct value of n_1 when considering individual vertices the value of n_1 depends on the degrees of neighboring vertices for a given vertex (minus the number of common triangles). Thus, n_1 is the key property that determines that the connectedness of the graph truly corresponds to the fixed $\{2, 3\}$-gdd. We did not

Algorithm 3 Test of Δs

Input: triangleness of every stud of each vertex (based on Δs)
Output: Graph G with correct n_0, n_2 and n_3

while there is a vertex with unconnected stud **do**
    select a vertex v with the stud s of largest triangleness (larger than 0) that is so far not connected
    find a vertex u, so far not connected with v, with an unconnected stud of the same triangleness as s and with at least one other stud whose triangleness is not used, call it s'
    find a third vertex w that 1) has at least one stud with triangleness equal to s' that is either connected to s' or is not connected at all 2) has at least one other stud s''' whose triangleness is not used and that has a counterpart in a stud of vertex v
    if such studs exist **then**
        make a triangle of these three vertices uvw, connect their studs if needed and increase the used-triangleness of all studs and edges that participate in it by 1
    else
        algorithm failed
    end if
end while
return connected vertices

manage to find a direct analytical approach that ensures the correct value of n_1 – for this reason, we suggest using either BFS with heuristics based on the results discussed in section 2.2.1, or we can formulate it as a linear program where we must ensure the following properties:

- Every two edges are connected at most once.
- Triangles are well connected: All i triangle studs are connected with i triangle struts.
- g_1 is well connected: For every vertex v, the sum of the degrees of all neighboring vertices minus the number of triangles in which vertex v participates with each neighbor must be equal to g_1, i.e., $\sum_{u \in N(v)} g_0(u) -$ triangleness of edge $uv = g_0(v)$.

This rather daunting sequence of steps leads us to an algorithm that yields a correctly connected graph from a given $\{2, 3\}$-gdd. We do not lay out the entire algorithm in full since do not perceive it to be of substantial practical significance. Rather we state it to illustrate the thinking and types of challenges that graph reconstruction from small graphlet degree distribution entails – we have to ensure that the resulting graph has correct graphlet degree distribution (see illustration of the challenge in Figure 2.16) and that the graph is feasible (see illustration of the challenge in algorithm 3).

As the exploration above suggests, approaching graph reconstruction based on a graphlet distribution from the perspective of the configuration model becomes increasingly difficult as we need to address the complications arising from

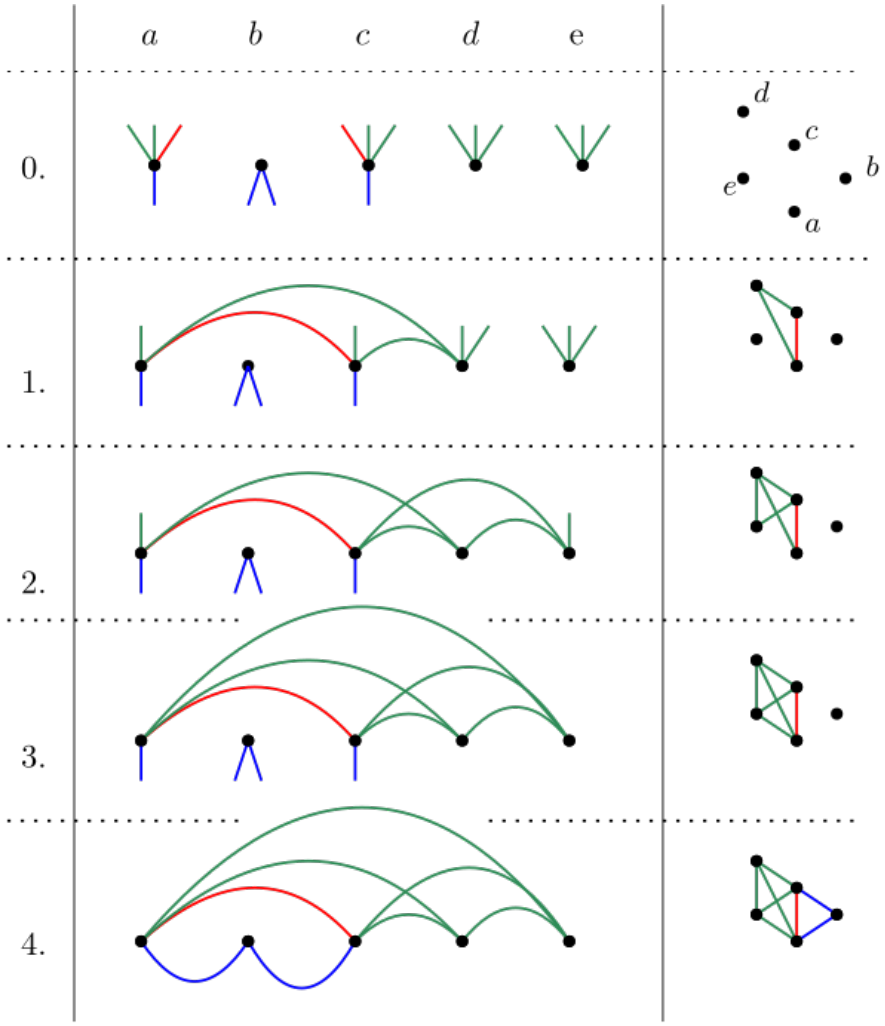


Figure 2.17: Steps of the algorithm. Iteration indicator is on the left, distribution of studs is in the center and gradually connected vertices forming graph are on the right. Starting with studs – color indicates triangleness (red:3; green:2; blue:1) – we gradually connect triangles that fulfill criteria.

overlapping graphlets and ensure correct connections between vertices across the entire graph. For this reason, we do not attempt to propose a 4-configuration model. Nevertheless, it should be noted that several other approaches might be worth exploring, such as constructing the largest rooted graphlets first and subsequently overlapping them or more extensively exploiting the properties of graphlet orbital interdependence discussed in Subsection 2.2.1.

Reconstruction from above

Turning our attention to reconstruction from above, we focus on $\{2, \ldots, n-1\}$-gdd, the graphlet degree distribution consisting of counts of all graphlets and their orbitals up to the size of $n-1$. Notably, there is a link between graphlet reconstruction and the reconstruction conjecture, as made explicit by Shervashidze et al. (2009). The reconstruction conjecture is a fundamental problem in graph theory that deals with the uniqueness of reconstructing a graph from its subgraph collection. In this subsection, we establish a connection between the

$\{2, \ldots, n-1\}$-gdd and the reconstruction conjecture, demonstrating under which conditions we are able to reconstruct a graph from its $\{2, \ldots, n-1\}$-gdd and discuss the implications this has for our framework for studying graphlets, the mapping $\mathcal{G}_\gamma^n$.

First, we restate graph reconstruction conjecture as proposed by Kelly (1957):

Theorem 12. *(reconstruction conjecture) Let G be a graph on n vertices. Let us denote G_v a subgraph of G formed by deleting vertex v from G. Let $D(G) = \{G_v | v \in V(G)\}$ be a multiset of all vertex-deleted subgraphs of G, called deck. Then any two graphs with $n > 2$, G, and H, such that their multisets $D(G)$ and $D(H)$ are equal, are isomorphic.*

This implies that given a deck, we are capable of reconstructing a uniquely determined graph (except for isomorphism). A positive answer to the conjecture has been given for a handful of graph classes – importantly for trees (Kelly [1957]); regular graphs (Harary [2006]); and disconnected graphs (Harary [2006]) – and, by computer-assisted enumeration of cases, for all graphs on $2 < n \leq 6$ vertices by Kelly (1957) which was later improved to graphs on $2 < n \leq 11$ vertices by McKay (1997). The general case, though, remains a conjecture.

When thinking about the reconstruction conjecture in the context of graphlets, we can notice that given a deck $D(G)$, we cannot securely determine appropriate $\{2, \ldots, n-1\}$-gdd since it requires the knowledge about orbitals for each vertex and its label in each G_v which would directly lead to the solution of the graph reconstruction conjecture.

Second, we can notice that

Theorem 13. *If we have a $\{2, \ldots, n-1\}$-gdd where every vertex participates in $n-1$ graphlet on $n-1$ vertices, it is possible the graph reconstruction deck of the graph.*

Proof. Since every vertex participates in $n-1$ graphlets on $n-1$ vertices, this means that the graph from which given $\{2, \ldots, n-1\}$-gdd originates is 2-vertex-connected. If this was not the case, there would exist a vertex x that, if deleted, would split the graphlet into two components and, thus, every vertex, except for x, could not have $n-1$ graphlets on $n-1$ vertices since deleting x does not lead to a graphlet on $n-1$ vertices. From this and theorem 6, it follows that $\{2, \ldots, n-1\}$-gdd is fully characterised by $\{n-1\}$-gdd. Further, if for every vertex we calculate all the graphlet on $n-1$ vertices that it touches, ignoring information about orbitals, sum this information over all vertices and divide it by $n-1$, we get all G_v graphs and thus full characterization of the deck. This follows from the fact that each graphlet is constituted by $n-1$ vertices and therefore when we sum the occurrence of each type of graphlet over all vertices, we get the total number for each graphlet $n-1$ times. □

Remark. Trying to determine a deck from a graphlet distribution without all values of $n-1$ graphlets is difficult since we encounter disconnected graphs where we have to determine individual components. We have discussed the same problem from different perspectives in remark 2.3.1.

It appears that it is diametrally easier to reconstruct a graph from a graphlet degree distribution than from a deck of vertex-deleted graphs. This is indeed the case as we can see in the following theorem:

Theorem 14. *Given a $\{2,\ldots,n-1\}$-gdd of a 2-vertex-connected graph where there is a vertex a that has a unique degree and a vertex b whose degree is different from all other degrees by at least two, we can uniquely reconstruct a graph with the given $\{2,\ldots,n-1\}$-gdd.*

Before we proceed with the proof, we show two small lemmas:

Lemma 15. *Given a $\{2,\ldots,n-1\}$-gdd and a graphlet G_i on $n-1$ vertices that v touches at g_j, we can uniquely determine the degree of vertex not included in G_i.*

Proof. First, we sum all $g_0(v)$ over all v in the graphlet distribution. This gives us double the number of edges in the whole graph since each edge is constituted by two vertices and, thus, contributes to the sum twice. Subsequently, we count the number of edges in the graphlet G_i and subtract this from the total number of edges in the whole graph. This gives us the number of edges not included in the graph which corresponds to the number of edges that the vertex not included in G_i touches, its degree. □

Lemma 16. *Given a $\{2,\ldots,n-1\}$-gdd of a 2-vertex-connected graph where there is a vertex a that has a unique degree, we can determine a graph vertex deleted graph G_a and establish how many edges are between a and some vertices in a certain automorphism orbit of G_a.*

Proof. From the 2-vertex-connectedness of the underlying graph, there are all $n-1$ graphlets on $n-1$ vertices. Therefore, we are assured that for each vertex v, there is a graphlet G_a where a vertex a is missing. Since a has a unique degree, this graphlet is unique by lemma 15. Furthemore, we can establish where each of the vertices $v \neq a$ are touching G_a from the $\{2,\ldots,n-1\}$-gdd and compare the degree of each vertex v in G_a with its $\{2\}$-gdd(v). If the two values differ, it means that the vertex is connected with the missing vertex a. Therefore, how many vertices of orbit g_i of G_a are connected with a by an edge? An illustration of the result can be seen in Figure 2.18.

□

Proof. (of theorem 14) The proof has two steps. First, from Lemma 16 we can identify a specific graphlet G_a where we know the position of all vertices except for a and its connection with the rest. We can consider the vertex b which has a unique degree even in G_a – thanks to the fact that it is different by at least two from all other vertex degrees, it maintains its uniqueness independently of whether it is connected to a. Thanks to this, we can uniquely identify b in G_a and for each vertex note whether there is an edge between it and b. We keep this information.

Second, thanks to the unique degree of b, we can get complete information for G_b through a process described in 16, and based on the information from the first step, we can determine which vertices are connected to b. This completes the reconstruction of G from $\{2,\ldots,n-1\}$-gdd. □

Remark. It should be noted that if we did not insist on the guarantee of a unique solution, we could stop at the result from lemma 16 and proceed by assigning possible connections between orbitals and vertex a and testing whether graphlets

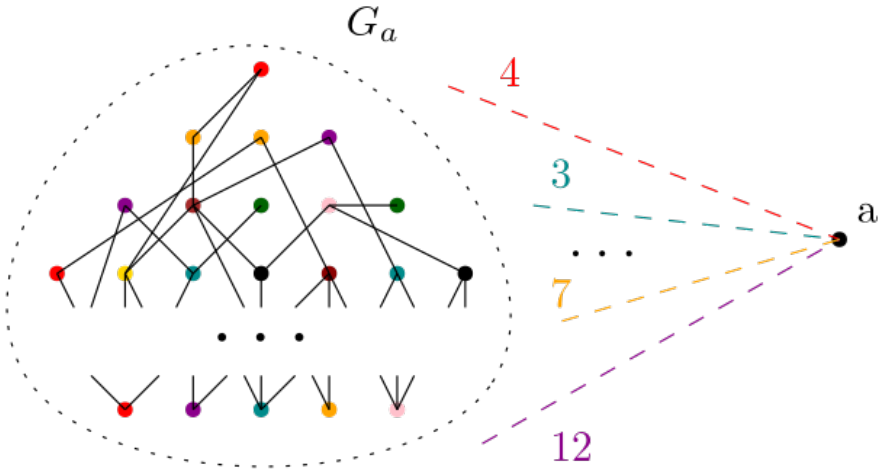


Figure 2.18: The remaining vertices can be placed into orbits of a-vertex deleted graphletG_a (colored vertices in the image) and we can find out how many edges are between vertices of certain orbit and vertex a (colored dashed lines with a number indicating number of edges).

of the size $n-1$ correspond until we found a feasible solution. This process would ensure that, if there is a solution, it will be found and after at most $\binom{n}{\lfloor \frac{n}{2} \rfloor}$ tries. [14]

If we turn back to our framework for thinking about the strength of graphlets, the mapping $\mathcal{G}^n_\gamma$. We can conclude that, if reconstruction conjecture holds, $\mathcal{G}^n_{n-1}$ is bijective for 2-vertex-connected graphs as was shown in theorem 13. Further, independently of reconstruction conjecture, we have shown that $\mathcal{G}^n_{n-1}$ gives us sufficient information if the class of graphs is 2-vertex-connected and has at least two sufficiently special vertices as was proved in theorem 14.

2.4 Summary and further work

To conclude, graphlets are an interesting tool for exploring graphs and their topology, offering a unique perspective different from commonly used approaches such as the Weisfeiler-Lehman isomorphism test or reconstruction from vertex-deleted graphs. They are also related enough to draw from the extensive literature in these explored areas. The main result of this chapter is to demonstrate the potential that graphlets can offer. However, the results and connections discussed above only scratch the surface of the possibilities. In this section, we outline possible directions for future research based on what has been shown.

1. In lemmas 2 and 3, we have provided constraints on the possible values of $\{p, \ldots, q\}$-gdd. Is it possible to establish a better range of $\{p, \ldots, q\}$-gdd for specific classes of graphs? Is it possible to establish the set of matrices for a given class of graphs so that the projection of graphs onto this set is surjective?

[14] This would occur when all vertices in G_a were touching the graphlet at the same orbit and there were $\frac{n}{2}$ edges to connect with a.

(a) What are the potential ranges of values in $\{p, \ldots, q\}$-gdd given a fixed number of vertices n and edges m in the graph, as well as fixed values of p and q? What if we consider only the sum of all values from the graphlet degree distribution?

2. How do graphlets behave on various classes of graphs? We have shown that graphlets are of limited use on tree-like graphs and particularly on regular graphs without small cycles, as stated in lemma 7 and theorem 8.

3. What is the distinguishing power of k-dim WL compared to $\{2, \ldots, \gamma\}$-gdd for certain $\gamma \geq 2$? In Figure 2.12 we showed some potentially complicated settings and in Figure 2.13 and 2.14 we presented some promising cases.

   (a) What is the smallest γ for which there are no two graphs with different labelings after 2 iterations of 1-dim WL, but with equal $\{2, \ldots, \gamma\}$-gdd? Solving this problem would strenghten the distinguishing power of graphlets by connecting it with the work of Babai and Kučera (1979) as discussed in remark 2.3.1.

   (b) What is the smallest γ such that, if the labeling of any two graphs after stabilization of 1-dim WL is different, implies that also their $\{2, \ldots, \gamma\}$-gdd are different?

   (c) Is there a γ that makes $\{2, \ldots, \gamma\}$-gdd capable of distinguishing non-isomorphic graphs more effectively than k-dim WL? Theorem 14 suggests this for $\gamma = n - 1$ for certain classes of graphs.

4. Theorems 10 and 14 suggest that $\{2, \ldots, k\}$-gdd$^\circ$ have different expressive power than $\{2, \ldots, k\}$-gdd. Is this truly the case? In which settings do these two definitions of graphlet degree distribution coincide, and when do they behave differently?

5. How large does γ need to be for $\mathcal{G}\gamma^n$ to be bijective, meaning that given $\{2, \ldots, \gamma\}$-gdd, we can uniquely reconstruct a graph? Theorem 14 provides a partial positive answer for $\gamma = n - 1$, while the graphs in Figure 2.11 provide a counterexample for $\gamma = \frac{n}{2} - 1$.

Furthermore, we can make the following observations, which may not be as rigorously grounded in analysis but can provide useful, albeit potentially misleading, intuition for working with graphlets. These observations can be particularly helpful when interpreting empirical results in the subsequent chapters.

- Graphlets are interrelated (as shown by in subsection 2.2.1 and 2.2.2). This suggests that complete graphlet degree distribution entails a large amount of redundant information.

- Graphlets perform poorly in characterizing graphs that locally behave tree-like. In such graphs, we observe only simple graphlets, paths, and small trees.

- Graphlets, nonetheless, prove to behave well in graphs with complex local structure (as suggested in figures 2.14 and 2.12) and they can outperform methods directed at identifying graph isomorphisms.

3. Graphlets in Complex Networks

Graphlets are capable of capturing the local topology of a network. For this reason, it is reasonable to explore how they can be used as a heuristic when comparing different network models. In this chapter, we focus on this inquiry. First, we summarise experimentally shown characteristics of graphlets in the context of network science[1] and, harnessing our theoretical results from chapter 2, make the case that graphlets are very well suited for network characterization. Second, we explore how graphlets can be used for network comparison – computing graphlet degree distribution yields a matrix that is not directly comparable with other matrices, thus, we carry out a discussion of possible approaches to network comparison. Finally, we use graphlets to compare the three classical models of networks (ER, GEO, and AB described in section 1.3.2) and suggest future directions for research.

Throughout this section, when we refer to graphlets, we consider graphlets on up to 5 vertices, $\{2, 3, 4, 5\}$-gdd, since 1) those are the largest graphlets that we can efficiently compute (Ribeiro et al. [2021]) and 2) it is the most commonly used size of graphlets considered and, thus, there is the most literature about them in which we can base our considerations.

3.1 Existing results

Graphlets, as described in chapter 2, were first introduced by Pržulj with the intention to capture the local topology of networks. In this section, we argue that graphlets are well-equipped for exactly this undertaking. The structure of this section is the following: we make three claims about the properties of graphlets and, subsequently, we support it with both literature and theoretical results. It should be noted that most of the empirical results in the literature on graphlets originate from the study of biological networks. This reason for that is simple – Pržulj is closely affiliated with bioinformatics and the concept has been studied mostly in reaction to her findings within the sphere of biological networks, although there is no basis for not harnessing its power in different subfields.

Graphlets are good at capturing local topology.

In our discussion of the expressive properties of graphlets in subsection 2.3.1, we suggested that graphlets, even if we consider only small graphlets, might be quite effective in distinguishing different graphs, especially when it comes to distinguishing local topology rather than pure isomorphism. This has been suggested in empirical research. Milenković and Pržulj (2008) compared nodes based on graphlets and demonstrated that clusters of nodes in protein-protein interaction networks, obtained with their graphlet-based distance measure, share common

[1] We did not include the results presented in this chapter in section 2.2 since, although related to graphlet, are mostly experimental and are, in the majority of cases, applicable only within network science (applied on networks) rather than to the properties of graphlets themselves.

protein properties. They showed how to use this approach to predict the functions of proteins and their memberships in protein complexes, subcellular compartments, and tissue expressions. The results were subsequently compared with laboratory experiments and over 80% percent of them proved correct (Hočevar and Demšar [2014]). Similar experiments were carried out by Milenković (2009). Furthermore, several methods for characterizing functions and local topology of vertices have been developed based on graphlets – such as GREAT (Crawford and Milenković [2015]), GR-align (Malod-Dognin and Pržulj [2014]), or L-GRAAL (Malod-Dognin and Pržulj [2015]) – all of which proved highly successful, commonly outperforming other approaches based on different attempts to describe local topology.

> Graphlets are especially good descriptors of graphs with higher edge density.

The usage of graphlet-based approaches for network description and comparison has been criticized by Rito et al. (2010) based on the observation that values of graphlet degree distribution are unstable in regions of a network of low edge density. This observation can be linked with our observation (see Lemma 7) about the behavior of graphlet degree distribution in tree-line structures (which is the class of graphs that occurs most commonly in regions with low density since the presence of cycles would increase density) where we concluded that graphlets are not too strong in describing graphs with tree-like structure. Nevertheless, the work of Hayes et al. (2013) shows that although it is the case that graphlets are unstable and ill-fitted to describe regions of low density in real-world networks, the regions with low density tend to be unstable, hard to predict themselves and susceptible to minor errors in measurements (translation of data into networks, see section 1.3). Furthermore, no serious challenge to the capacity of graphlets to capture local topology in dense regions was put forward and, on the contrary, a number of studies reaffirmed graphlets as a good measure in this sense (Milenković and Pržulj [2008]; Hayes et al. [2013]).

> Graphlets implicitly include multiple heuristics.

Some researchers called for combining different heuristics in the attempt to characterize networks (see section 1.3.3) and when using machine learning techniques for model testing, created vectors that consisted of multiple simpler heuristics (Filkov et al. [2009]). Nevertheless, if we consider a graphlet vector $\{2,3,4,5\}\text{-gdd}_G(v)$, it turns out that a nontrivial number of local heuristics are included. For example, of those mentioned in section 1.3.1 [2], we can directly determine degree distribution (by g_2) and node clustering (by g_3), and we can approximate other such as local betweenness centrality can be estimated based on the number of times v touches central graphlet orbits (e.g. g_5, g_7, g_11, g_17, g_23...) rather than peripheral graphlet orbits (e.g. g_4, g_6, g_9, g_10, g_15, g_18...). The only heuristics of those discussed in section 1.2 that are virtually impossible to establish from graphlet degree distribution, unless a very strong version of

[2] Besides those mentioned in section 1.3.1, we can determine the following characteristics that were not mentioned since they are principally similar : rich club connectivity, average triangle coefficient, maximum triangle coefficient or average quadrangle coefficient.

graphlet reconstruction is proved, are global heuristics, however, those are often perceived as inferior in characterization to local heuristics (see 1.3.3).

It should be noted that Janssen et al. (2012) harnessed the strength of graphlets to caputre multiple characteristic in their effort to find the best network model using unsupervised machine learning methods.

3.2 Graphlets as a metric

Graphlets seem as a positive perspective in the efforts to characterize and compare networks. Nevertheless, if we want to use graphlets for the comparison of networks, we face the problem that graphlets are presented as a matrix of values (in the case of graphlets up to the size of 5, $\{2,3,4,5\}$-gdd $\in \mathbb{N}^{n \times 73}$). We have to establish how to transform $\{2,3,4,5\}$-gdd into a single comparable value that could tell us which networks are more similar. This section deals with this problem and offers an array of different approaches with justification for their use.

If we want to compare two graphlet degree distributions, $\{2,3,4,5\}$-gdd(G) and $\{2,3,4,5\}$-gdd(H), we either aggregate the data from a matrix in a way that is vertex permutation independent or somehow pair, align, vertices from graph G and H (if we used a permutation dependent approach without reasonable alignment, we could get completely different results based on which labeling was arbitrary chosen which is not desirable). We first discuss vertex-permutation independent approaches used in graphlet degree distribution comparison and then discuss how alignment can be carried out and the approaches that it enables.

There are many possible approaches to matrix comparison, we select a subset of those that were traditionally used in the comparison of networks using graphlets (graphlet degree distribution agreement, sum of differences) or those that we consider sensible (mutual information, distance correlation). We acknowledge that the selection is limited and a more thorough discussion is in place, nevertheless, we consider it sufficient to coarsely demonstrate what graphlets can tell us about various models.

3.2.1 Permutation independent approaches

The simplest possible approach to matrix comparison lies *sum of difference* (SD for short), summing over all the values in each of the graphlet degree distributions and subsequently computing the difference between the two sums. Although this approach is computationally efficient, simple, and permutation invariant, it erases most of the valuable information in the graphlet degree distribution – we are not able to distinguish if two networks are locally densely connected or sparsely connected if it leads to the total number of graphlets in the networks.

Another approach, called *Graphlet degree distribution agreement* (GDDA for short), initially proposed by Pržulj (2007) suggests the following, more elaborate approach – we appropriate the notation from the paper.

Let $d_G^j(k)$ be the sample distribution of the number of nodes in G touching the appropriate graphlet (for automorphism orbit j) k times. We scale $d_G^j(k)$ by k to decrease the contribution of larger degrees in a graphlet degree distribution and

call the new variable $S_G^j(k) = \frac{d_G^j(k)}{k}$ and subsequently normalize the distribution by the total which leads to the following:

$$N_G^j(k) = \frac{S_G^j(k)}{T_G^j} \tag{3.1}$$

where T_G^j is:

$$T_G^j = \sum_{k=1}^{\infty} \tag{3.2}$$

and when we want to compare two networks, we compute the following:

$$D^j(G,H) = (\sum_{k=1}^{\infty}[N_G^j(k) - N_H^j(k)]^2)^{1/2} \tag{3.3}$$

this gives us the degree distribution disagreement, the difference between the two networks, which we can inverse (subtracting it from 1) and sum over all possible graphlets to obtain graphlet degree distribution agreement

$$A_{\text{arith}}(G,H) = \frac{1}{73}\sum_{j=0}^{7} 2(1 - D^j(G,H)) \tag{3.4}$$

Note that the infinite sum when calculating T_G^j and $D^j(G,H)$ are in practice finite due to the bounded sample. We keep the infinite sum introduced by Pržulj (2007) although we could use our bound from lemma 2.

The graphlet degree distribution agreement is a feasible way to capture the representation of certain graphlets throughout the entire distribution. Nevertheless, although nuanced and sensibly normalized, the approach is unable to account for 1) specific regions with different topologies since it does not consider individual nodes, only the distribution of graphlet orbitals over the entire distribution and 2) a lot of information is again lost when summing over the entire distribution (although in a sensible way) as we are unable to account for local differences.

3.2.2 Permutation dependent approaches

Before we can start entertaining the idea of using permutation-dependent approaches for graphlet degree distribution comparison, we have to align the vertices of the two networks. We want to pair vertices that behave most alike – have similar topology around themselves and are connected into vertices that are also mutually paired. Doing this perfectly is extremely difficult and, for this reason, we resort to the usage of sensible approximate approaches.

Alignment of vertices

A simple approach would be to take degrees of vertices and pair them according to their degree and pair those with equal degrees randomly. This might be a computationally effective approach but might easily end up comparing vertices that are topologically completely different and, thus, completely obscure any results.

This simple procedure might be further improved – for example, we might take an iterative approach where we start pairing vertices into groups by degree and then continue splitting these groups by values of increasingly complicated graphlet orbitals (here we can utilize the ordering of graphlets). This approach might be a slight improvement in the approach above, least of all it would be deterministic, but it fails to harness the data about graphlets that we have and it is dependent on the convention that we use for ordering graphlets.

Based on this we suggest a graphlet alignment method that pairs vertices with the most similar graphlet degree vector ($\{2,3,4,5\}$-gdd$_G(v)$) – as it turned out during our investigation of literature a very similar approach was suggested by Milenković et al. (2010). We suggest using the Hungarian algorithm to establish a perfect matching between sets of vertices of each network where the cost function is given by a specified metric on 73-dimensional metric space in which we can embed graphlet degree vectors of individual vertices. This approach enables us to find the best possible matching for vertices based on the similarity of their local topologies. It is unable to establish that individual vertices are paired in such a manner that for each of two neighboring vertices, their pairs are also neighbors, but we consider this shortcoming as an acceptable limitation in return for the polynomial computational complexity that the Hungarian algorithm offers (Edmonds and Karp [1972]). Milenković et al. (2010) test this alignment method empirically and show that the "method detects topologically similar regions in large networks that are statistically significant" substantiating the functionality of the approach by empirical data.

Throughout the rest of this text and in analyzing data, we use the just-described approach to matrix comparison.

Permutation dependent approaches

If we have paired vertices, we can transform the matrix into a vector, flatten it, where the ordering of individual nodes depends on the pairing found – let us call them X_G and Y_H for the two networks G and H being compared. This, in turn, enables us to use many powerful statistical tools for measuring correspondence between the two vectors. We want to avoid tools that are capable of measuring only linear dependence such as the Pearson correlation coefficient since graphlets do not behave in that way and we want to evaluate similar behavior of similar nodes, not linearity of this behavior. For this reason, we propose to use *mutual information* and *distance correlation*.

First, we can compute *mutual information* (MI for short), bits of information obtained about one vector by observing the other. This tool is commonly used in statistics and machine learning to capture the statistical dependence between two random variables. The approach evaluates the difference between the joint distribution of the pair (X_G, Y_H) in comparison to the product of marginal distributions of X_G and Y_H. More technically, the mutual information (in this case for discrete distribution which is the case of graphlet counts) is calculated as follows:

$$I(X_G, Y_H) = \sum_{y \in \mathcal{Y}_\mathcal{H}} \sum_{x \in \mathcal{X}_\mathcal{G}} P_{(X_G, Y_H)}(x, y) log(\frac{P_{(X_G, Y_H)}(x, y)}{P_{X_G}(g) P_{Y_H}(y)}) \tag{3.5}$$

where $\mathcal{X}_\mathcal{G}$and $\mathcal{Y}_\mathcal{H}$ are the underlying spaces on which elements of X_G and Y_H

are defined. For a more detailed discussion of the properties of mutual information measure and information theory from which it originates, see Bishop (2006).

Second, *distance correlation* (dCor for short) is a statistical measure, introduced by Székely et al. in 2007, that captures dependence between two variables by evaluating the similarity of their pairwise distances. Unlike traditional correlation measures that focus on linear relationships, distance correlation is a nonparametric approach that can detect and quantify nonlinear associations between variables. It accomplishes this by considering all possible pairwise distances and evaluating the relationships among these distances in the input space. By considering all pairwise distances, distance correlation offers a comprehensive measure of association that is not influenced by the scales of the variables. More technically, the *distance correlation* (notation is appropriated from Székely and Rizzo [2009]) is computed as follows:

We compute all pairwise distances.

$$\begin{aligned}(a_{j,}) &= ||X_j - X_k||, \text{ for } j,k = a,\ldots,n \\ (b_{j,}) &= ||Y_j - Y_k||, \text{ for } j,k = a,\ldots,n\end{aligned} \tag{3.6}$$

Then we compute the doubly centered distance

$$\begin{aligned}A_{j,k} &= a_{j,k} - \overline{a_{j,\cdot}} - \overline{a_{\cdot,k}} + \overline{a_{\cdot,\cdot}} \\ B_{j,k} &= b_{j,k} - \overline{b_{j,\cdot}} - \overline{b_{\cdot,k}} + \overline{b_{\cdot,\cdot}}\end{aligned} \tag{3.7}$$

where $\overline{a_{j,\cdot}}$ is the average of j$^{\text{th}}$ row, $\overline{a_{\cdot,k}}$ is the average of k$^{\text{th}}$ column and $\overline{a_{\cdot,\cdot}}$ the average over the whole distribution. Similarly for $\overline{b_{j,\cdot}}$, $\overline{b_{\cdot,k}}$ and $\overline{b_{\cdot,\cdot}}$.
We can use this to calculate the *distance covariance*

$$dCov_n^2(X,Y) = \frac{1}{2}\sum_{j=1} n \sum_{k=1} nA_{j,k}B_{j,k} \tag{3.8}$$

From which, we can compute the distance correlation, denoted as $dCor^2(X,Y)$:

$$dCor^2(X,Y) = \frac{dCov_n^2(X,Y)}{\sqrt{dVar^2(X)dVar^2(Y)}} \tag{3.9}$$

where $dVar^2(X) = dCov_n^2(X,X)$ alike classical variance.

Discussion of the benefits and downsides of this approach is beyond the scope of this text, we refer to the text by Hastie et al. (2015) for a more thorough exposition. For the most part, we use it as a tool to provide us with some quantifiable information about the graphlet distribution.

Finally, it should be pointed out that *visualization* is also an important approach to graphlet matrix analysis as it can give us a sense, although not quantifiable in clear numbers, of what is going on in the distribution. This approach is also highly dependent on alignment since visualization based on that might look very different – consider for example ordering of degrees of vertices in a network which was the basis for the development of the AB model, if we ordered the degree distribution differently, we might not be able to see the underlying pattern. Visualization is an important tool during analysis but we should take the conclusions drawn from it with caution.

3.3 Application on real-world data

Having discussed how graphlets can be used for the comparison of different models, we carry out experiments on real-world data and study how well classical models capture the behavior of graphlets. Concretely, in this section, we first discuss what data we used and how they were created, second, we summarize how we process the data and come up with the results that we have and, finally, we discuss the results.

Similar experiments were carried out by Pržulj (2007) and Milenković (2010). Their analysis was carried out on biological networks, more precisely protein-protein interaction networks, and the main approach for comparing networks was based on graphlet degree distribution agreement measure – their study concluded that GEO models are the best suited for PPI modeling. In this section, we experiment with a wider array of networks and employ a series of comparison methods (described in the previous section). These approaches and results are (except for GDDA), to our admittedly limited knowledge, novel.

3.3.1 Data used

We considered 31 networks saved in Network Repository (Rossi and Ahmed [2015]). We selected four different types of networks : relation between genes (9 networks), interaction between neurons in the brain (8 networks), networks of ecosystems (7 networks), and infrastructure networks (7 networks). The criterion for selecting a network was 1) size (our computational capacity was limited), 2) completeness (we wanted to avoid networks with incomplete data), and 3) reasonable density (some networks were too dense or too sparse which is not representative and introduces a large amount of noise in the graphlet counting).

3.3.2 Data processing

Given a network, we process the data in the following fashion:

1. first computed degree distribution for the original network (using Orca developed by Hocevar and Demsar (2014));
2. generated ER, GEO, and AB models (described in 1.3.2) that have an equal number of vertices and a similar number of edges as the original network (since all of these models are probabilistic, it is difficult to ensure that a model has the same number of edges, we accepted 2.5% deviation in the number of edges compared to the original network);
3. paired vertices according to the process described in section 3.2.2
4. compute the metrics for network comparison described in section 3.2
    - sum of differences
    - graphlet degree distribution agreements
    - mutual information
    - distance correlation
5. visualize the data into a 3-dimensional histogram

3.3.3 Results and discussion of experiments

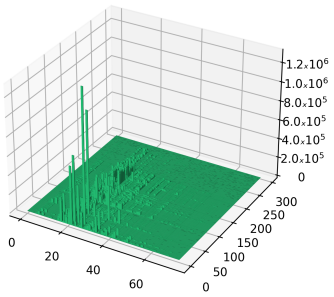
To present the results, we have decided to select 4 representative networks from the 31 considered, one from each type of network. Individual networks differ in many parameters (especially prominent differences between different types of networks – brain vs. infrastructure), but the patterns described below were observed in over 85% of the networks considered. Since we do not dwell on details of individual networks and follow general patterns, we consider these observations representative.

We first present representative networks, visualize the results (larger images are in appendix A.1) and present results of comparison methods, subsequently, we analyze and discuss the results.

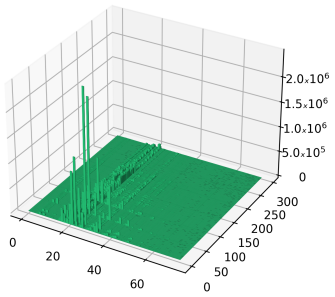
Note that all the graphlet degree distributions presented below are vertex-aligned using the mechanism described in section 3.2.2 against the original network.

Results

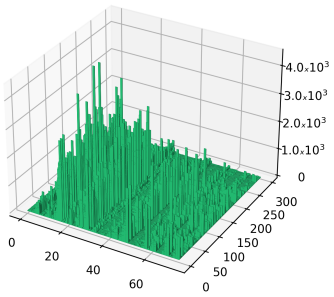
The following data are based on the neural network of *C. elegans* (Chen et al. [2006]) on 297 vertices and 1720 edges.



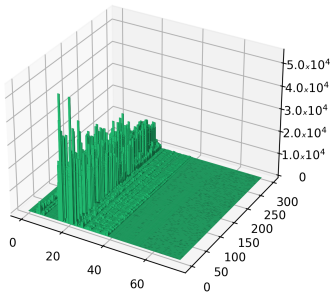
(a) Original network



(b) Albert-Barabási model



(c) Geometric model



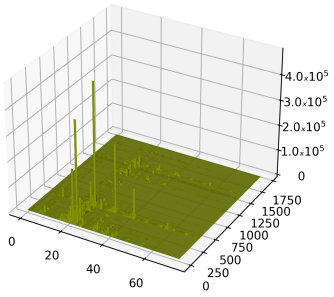
(d) Erdős-Rényi model

Figure 3.1: Graphlet degree distribution of the neural network of *C. elegans* and models of it (AB, GEO, ER)

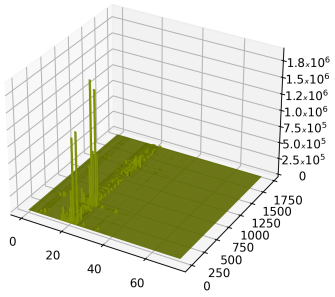
Table 3.1: Results of different comparison methods – all values are calculated compared to the original network, lighter color indicates better standing

Name	**Original network**	**AB model**	**GEO model**	**ER model**
SD	0	40830	76033	77783
dCor	1.0	0.942	0.236	0.360
MI	6.40	4.49	3.19	3.34
GDDA	0.0	0.329	0.255	0.337

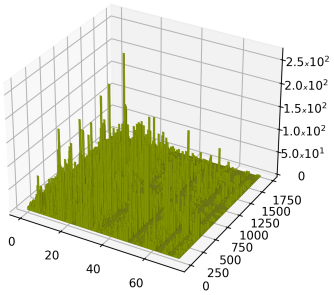
The following is based on a network of interaction between genes in an unnamed plant (Rossi and Ahmed [2015]) on 1717 vertices and 3098 edges:



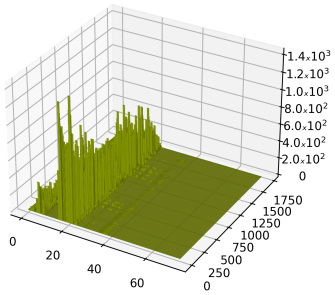
(a) Original network



(b) Albert-Barabási model



(c) Geometric model



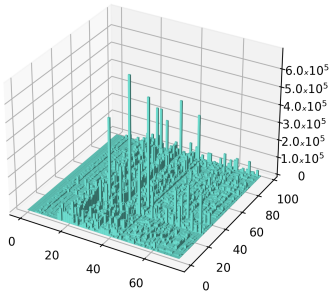
(d) Erdős-Rényi model

Figure 3.2: Graphlet degree distribution of the interaction between genes in an unnamed plant and models of it (AB, GEO, ER)

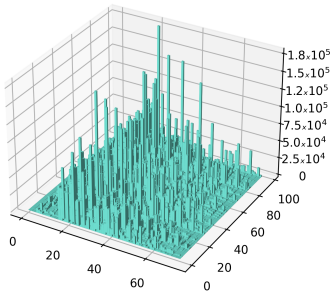
Table 3.2: Results of different comparison methods – all values are calculated compared to the original network, lighter color indicated better standing

Name	**Original network**	**AB model**	**GEO model**	**ER model**
SD	0	170293	151820	166468
dCor	1.0	0.636	0.067	0.102
MI	2.28	0.868	0.398	0.459
GDDA	0.0	0.121	0.109	0.07

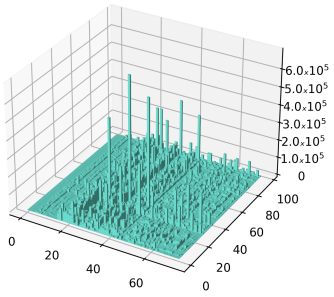
The following is based on a network of food web in tropical ecosystems of Florida (Ulanowicz and DeAngelis [1998]) on 87 vertices and 1446 edges:



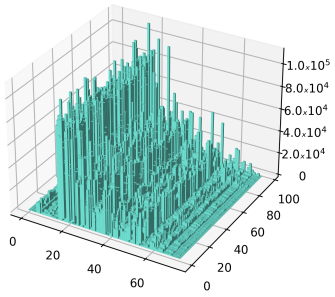
(a) Original network



(b) Albert-Barabási model



(c) Geometric model



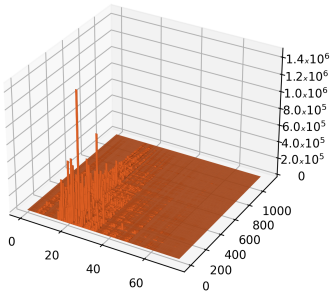
(d) Erdős-Rényi model

Figure 3.3: Graphlet degree distribution of the food web in tropical ecosystems of Florida and models of it (AB, GEO, ER)

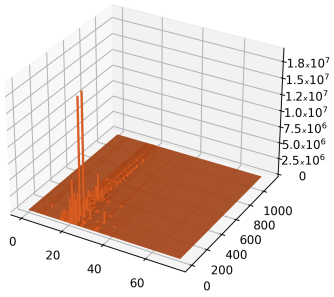
Table 3.3: Results of different comparison methods – all values are calculated compared to the original network, lighter color indicated better standing

Name	**Original network**	**AB model**	**GEO model**	**ER model**
SD	0	8775	18846	11782
dCor	1.0	0.629	0.417	0.403
MI	8.46	8.03	7.54	8.164
GDDA	0.0	0.423	0.426	0.41

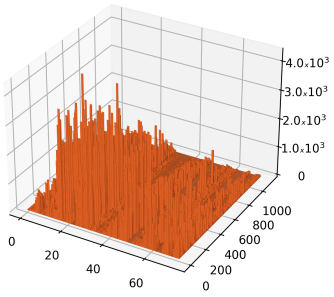
The following is based on data about email communication on an unnamed US university (Rossi and Ahmed [2015]) on 1133 vertices and 5451 edges:



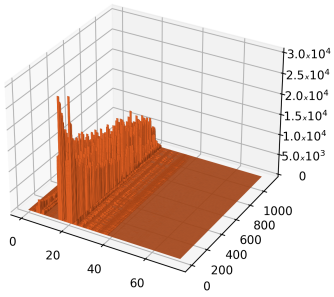
(a) Original network



(b) Albert-Barabási model



(c) Geometric model



(d) Erdős-Rényi model

Figure 3.4: Graphlet degree distribution of the email communication network and models of it (AB, GEO, ER)

Table 3.4: Results of different comparison methods – all values are calculated compared to the original network, lighter color indicated better standing

Name	**Original network**	**AB model**	**GEO model**	**ER model**
SD	0	161399	235012	272087
dCor	1.0	0.648	0.391	0.467
MI	5.77	3.168	2.231	2.08
GDDA	0.0	0.21	0.244	0.165

Observation and analysis

The results above are only representatively chosen samples. The following observations are made on all the data produced:

- The best performing model seems to be the Albert-Barabási model – both from the perspective of visual analysis, graphlet degree distribution has a similar shape and structure as original networks, and from the perspective of comparison methods based on graphlets. Metrics for comparison suggest that this holds even in denser networks where visual analysis becomes increasingly difficult (see figure 3.3) Nevertheless, the correspondence between the AB model and the original network is not without flaws – upon closer inspection, we can see that there are often a few nodes that appear to participate in complex graphlets (this can be seen well in figures 3.1, 3.2 or 3.4). This observation further corresponds to the result of Haddadi et al. (2009) who claim that the AB model fails to capture small well-connected regions in sparser real networks due to its excessive focus on scale-free property.

- Erdős-Rényi models produce networks with too little complex local topology (orbitals from the most tree-like graphlets from each graphlet size (g_4-$g_1$1 and $g_1$5-$g_2$3)) – this stops to apply as much in dense networks (such as the one represented on figure 3.3) where less complicated graphlets are still overrepresented but the difference appears smaller. This can be linked with the properties of the ER model (see section 1.3.2) – there is little clustering in sparse networks but the average paths are relatively short, which leads to a large number of relatively simple graphlets which is exactly what we observe.

- The tool for comparison of two networks, graphlet degree distribution agreement, is the only metric through which the Geometric model appears best-suited for modeling real-world networks. We theorize that this might be caused by the fact that more frequent graphlets are normalized during GDDA calculation which prioritized either very well-fitting models or models that have an overall lower and homogeneous distribution of graphlets which is exactly the case of the Geometric model.

- Geometric models produce networks with very complex local topology (orbitals from highly connected graphlets are highly represented even in relatively sparse networks see figures 3.1, 3.2 or 3.4). This can be linked to their tendency to create clusters that participate in more complex graphlets (see section 1.3.2). This result disputes the results of Pržulj (2007) and Milenković (2010) who proclaimed Geometric network as the best model for real-world networks under the graphlet degree distribution agreement measure – although GEO model appears as a good fit from the perspective of GDDA, other measures do not support this view (see previous observation).

- In general, we can observe that: 1) when networks are sparse, we can observe the presence of slightly more complicated graphlets than pure paths and trees with a few outliers that participate in more complex graphlets; 2)

when the network gets denser, we observe increase in local complexity but rather than observing a small number of dense clusters, the complexity of local topology increases overall.

3.4 Summary and further work

In this section, we have shown that there is a number of possible usages of graphlets in network description and comparison. Further, we utilized these and applied them to real-world networks. We suggest, based on the application of graphlet-based network comparison methods, that the AB model is best suited for real-world network modeling, successfully mimicking the graphlet degree distribution.

Furthermore, besides the results shown above, we have left behind many loose ends that invite promising future research. Concretely:

- All the approaches for aggregating information from graphlet degree distribution described in section 3.2 are formed on an ad hoc basis – there is a need for a rigorous framework assessing different approaches carefully tested on real-world data as well as synthetic networks. Different graphlet-based approaches should be thoroughly tested to establish which captures which property of networks. Ideally, this framework should be rooted in well-studied theoretical properties of graphlets.

- Graphlet-based analysis should be carried out on a larger number of large networks and compared to more models used for modeling networks. In this way, the presented results could undergo thorough scientific scrutiny.

- Based on our analysis of chapter 2 and the experimental results discussed in section 3.1, graphlets appear to be a potentially very good heuristic for network comparison based on its structure. In this light, graphlet might be a good basis for a new model for real-world networks. Nevertheless, much research into the theoretical and experimental properties of graphlets is needed before this step is attempted.

Conclusion

In this thesis, we defined and discussed graphlets in the context of network science. The discussion was carried out in three separate chapters.

In the first chapter, we discussed the context of network science. We presented a brief overview of the historical development of the field and offered insights into the prevailing methodologies. Furthermore, we distinguished two key, albeit overlapping, aspects within the field: network description based on characterization methods and network modeling and comparison. We presented commonly used tools and models in the field.

In the second chapter, we delved into the topic of graphlets themselves. We provided a comprehensive definition of graphlets and surveyed the existing literature related to their theoretical properties. Subsequently, we developed a framework for studying the relationship between graphs and graphlets. Within this framework, we explored several theoretical properties of graphlets, including the relation between large graphlets and smaller ones, their behavior on simple classes of graphs, and graphs that lead to the same graphlet degree distribution. We also proposed a way of extending the *configuration model* to the *3-configuration model* that enables the generation of graphs with a prescribed graphlet degree distribution for graphlets of size up to 3. Additionally, we established preliminary connections between graphlets and the *Weisfeiler-Lehman isomorphism test* and the *reconstruction conjecture*, demonstrating the reconstructability of a graph from its graphlet degree distribution under certain conditions (see theorem 14 for more details). Finally, we proposed avenues for further research that graphlets offer.

In the third and final chapter, we bridged the previous two chapters and explored how graphlets can be employed in network science for network characterization and comparison. Building upon existing experimental literature on graphlets, we argued for their efficacy as a tool for network characterization. Subsequently, we discussed sensible approaches for utilizing graphlets in network comparison, including established methods and alternative strategies. Furthermore, we applied graphlet-based comparison methods to a series of real-world networks, demonstrating that, contrary to existing research, the Albert-Barabási model appeared to be the best fit for real-world networks according to our graphlet-based metrics. Finally, we proposed possible extensions of our empirical findings and encouraged further exploration of graphlet-based comparison methods.

Overall, graphlets show great promise for future research, both theoretically and empirically, with the potential to yield far-reaching consequences in the field of network science and establish intriguing connections to graph theory. This thesis only scratches the surface of the possibilities that graphlets offer, leaving ample room for further exploration and application of this powerful tool.

Bibliography

Marién Abreu, John Baptist Gauci, and Jean Paul Zerafa. Saved by the rook: a case of matchings and hamiltonian cycles. *arXiv preprint arXiv:2104.01578*, 2021.

Réka Albert. Scale-free networks in cell biology. *Journal of cell science*, 118(21): 4947–4957, 2005.

Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

Yael Artzy-Randrup, Sarel J Fleishman, Nir Ben-Tal, and Lewi Stone. Comment on "Network motifs: simple building blocks of complex networks" and "Superfamilies of evolved and designed networks". *Science*, 305(5687):1107–1107, 2004.

Vikraman Arvind, Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky. On weisfeiler-leman invariance: Subgraph counts and related graph properties. *Journal of Computer and System Sciences*, 113:42–59, 2020.

Furqan Aziz, Afan Ullah, and Faiza Shah. Feature selection and learning for graphlet kernel. *Pattern Recognition Letters*, 136:63–70, 2020.

László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697, 2016.

László Babai and Ludik Kucera. Canonical labelling of graphs in linear average time. In *20th annual symposium on foundations of computer science (sfcs 1979)*, pages 39–46. IEEE, 1979.

László Babai and Rudi Mathon. Talk at the south-east conference on combinatorics and graph theory. 1980.

László Babai, Paul Erdos, and Stanley M Selkow. Random graph isomorphism. *SIaM Journal on computing*, 9(3):628–635, 1980.

Albert-László Barabási. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987): 20120375, 2013.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Dionysios Barmpoutis and Richard M Murray. Networks with the smallest average distance and the largest average clustering. *arXiv preprint arXiv:1007.4031*, 2010.

Alain Barrat and Martin Weigt. On the properties of small-world network models. *The European Physical Journal B-Condensed Matter and Complex Systems*, 13: 547–560, 2000.

Punam Bedi and Chhavi Sharma. Community detection in social networks. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 6(3):115–135, 2016.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308, 2006.

Béla Bollobás and Béla Bollobás. *Random graphs*. Springer, 1998.

Katy Börner, Soma Sanyal, Alessandro Vespignani, et al. Network science. *Annu. rev. inf. sci. technol.*, 41(1):537–607, 2007.

Geoffrey C Bowker and Susan Leigh Star. *Sorting things out: Classification and its consequences*. MIT press, 2000.

Dan Braha. Complex design networks: Structure and dynamics. *arXiv preprint arXiv:1801.02272*, 2018.

Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177, 2001.

Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE transactions on knowledge and data engineering*, 20(2):172–188, 2007.

Anna D Broido and Aaron Clauset. Scale-free networks are rare. *Nature communications*, 10(1):1017, 2019.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Tian Bu and Don Towsley. On distinguishing between internet power law topology generators. In *Proceedings. twenty-first annual joint conference of the ieee computer and communications societies*, volume 2, pages 638–647. IEEE, 2002.

Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

Beth L Chen, David H Hall, and Dmitri B Chklovskii. Wiring optimization can relate neuronal structure and function. *Proceedings of the National Academy of Sciences*, 103(12):4723–4728, 2006.

Xin W Chen. Network science models. In *Network Science Models for Data Analytics Automation: Theories and Applications*, pages 1–16. Springer, 2022.

Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? *Advances in neural information processing systems*, 33:10383–10395, 2020.

Joseph Crawford and Tijana Milenković. Great: graphlet edge-based network alignment. In *2015 IEEE International conference on bioinformatics and biomedicine (BIBM)*, pages 220–227. IEEE, 2015.

Sergei N. Dorogovtsev and José F.F. Mendes. *Evolution of networks: From biological nets to the Internet and WWW*. Oxford university press, 2003.

Jack Edmonds and Richard M Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2): 248–264, 1972.

Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60, 1960.

Rafael Espejo, Guillermo Mestre, Fernando Postigo, Sara Lumbreras, Andres Ramos, Tao Huang, and Ettore Bompard. Exploiting graphlet decomposition to explain the structure of complex networks: the GHuST framework. *Scientific reports*, 10(1):12884, 2020.

Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, pages 128–140, 1741.

Sergei Evdokimov and Ilia Ponomarenko. On highly closed cellular algebras and highly closed isomorphisms. *the electronic journal of combinatorics*, pages R18–R18, 1999.

Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. *ACM SIGCOMM computer communication review*, 29(4):251–262, 1999.

Stephen E Fienberg. A brief history of statistical models for network analysis and open challenges. *Journal of Computational and Graphical Statistics*, 21 (4):825–839, 2012.

Vladimir Filkov, Zachary M Saul, Soumen Roy, Raissa M D'Souza, and Premkumar T Devanbu. Modeling and verifying a broad array of network properties. *Europhysics Letters*, 86(2):28003, 2009.

Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5): 75–174, 2010.

Edgar N Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4): 1141–1144, 1959.

Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12): 7821–7826, 2002.

Michael E Gorman. Scientific and technological thinking. *Review of General Psychology*, 10(2):113–129, 2006.

Hongyu Guo, Khalique Newaz, Scott Emrich, Tijana Milenkovic, and Jun Li. Weighted graphlets and deep neural networks for protein structure classification. *arXiv preprint arXiv:1910.02594*, 2019.

Hamed Haddadi, Miguel Rio, Gianluca Iannaccone, Andrew Moore, and Richard Mortier. Network topologies: inference, modeling, and generation. *IEEE Communications Surveys & Tutorials*, 10(2):48–69, 2008.

Hamed Haddadi, Damien Fay, Almerima Jamakovic, Olaf Maennel, Andrew W Moore, Richard Mortier, and Steve Uhlig. On the importance of local connectivity for internet topology models. In *2009 21st International Teletraffic Congress*, pages 1–8. IEEE, 2009.

Frank Harary. A survey of the reconstruction conjecture. In *Graphs and Combinatorics: Proceedings of the Capital Conference on Graph Theory and Combinatorics at the George Washington University June 18–22, 1973*, pages 18–28. Springer, 2006.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

Wayne Hayes, Kai Sun, and Nataša Pržulj. Graphlet-based measures are suitable for biological network comparison. *Bioinformatics*, 29(4):483–491, 2013.

Pietro Hiram Guzzi, Francesco Petrizzelli, and Tommaso Mazza. Disease spreading modeling and analysis: A survey. *Briefings in Bioinformatics*, 23(4): bbac230, 2022.

Tomaž Hočevar and Janez Demšar. A combinatorial approach to graphlet counting. *Bioinformatics*, 30(4):559–565, 2014.

Lele Hu, Tao Huang, Xiaohe Shi, Wen-Cong Lu, Yu-Dong Cai, and Kuo-Chen Chou. Predicting functions of proteins in mouse based on weighted protein-protein interaction network and protein hybrid properties. *PloS one*, 6(1): e14556, 2011.

Yuriy Hulovatyy, Huili Chen, and Tijana Milenković. Exploring the structure and function of temporal networks with dynamic graphlets. *Bioinformatics*, 31 (12):i171–i180, 2015.

Jeannette Janssen, Matt Hurshman, and Nauzer Kalyaniwalla. Model selection for social networks using graphlets. *Internet Mathematics*, 8(4):338–363, 2012.

Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.

Paul J Kelly. A congruence theorem for trees. *Pacific J. Math.*, pages 961–968, 1957.

Johannes Kobler, Uwe Schöning, and Jacobo Torán. *The graph isomorphism problem: its structural complexity*. Springer Science & Business Media, 2012.

Jérôme Kunegis, Marcel Blattner, and Christine Moser. Preferential attachment in online networks: Measurement and explanations. In *Proceedings of the 5th annual ACM web science conference*, pages 205–214, 2013.

Noël Malod-Dognin and Nataša Pržulj. Gr-align: fast and flexible alignment of protein 3d structures using graphlet degree similarity. *Bioinformatics*, 30(9): 1259–1265, 2014.

Noël Malod-Dognin and Nataša Pržulj. L-graal: Lagrangian graphlet-based network aligner. *Bioinformatics*, 31(13):2182–2189, 2015.

Brendan D McKay. Small graphs are reconstructible. *Australasian Journal of Combinatorics*, 15:123–126, 1997.

Vesna Memišević, Tijana Milenković, and Nataša Pržulj. An integrative approach to modeling biological networks. *Journal of Integrative Bioinformatics*, 7(3): 65–86, 2010.

Tijana Milenković and Nataša Pržulj. Uncovering biological network function via graphlet degree signatures. *Cancer informatics*, 6:CIN–S680, 2008.

Tijana Milenković, Ioannis Filippis, Michael Lappe, and Nataša Pržulj. Optimized null model for protein structure networks. *PLoS One*, 4(6):e5967, 2009.

Tijana Milenković, Weng Leong Ng, Wayne Hayes, and Nataša Pržulj. Optimal network alignment with graphlet degree vectors. *Cancer informatics*, 9:CIN–S4744, 2010.

Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

Mark Newman. *Networks: An Introduction*. Oxford University Press, 03 2010. ISBN 9780199206650.

Mark EJ Newman. Assortative mixing in networks. *Physical review letters*, 89 (20):208701, 2002.

Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

J-P Onnela, Jari Saramäki, Jorkki Hyvönen, György Szabó, David Lazer, Kimmo Kaski, János Kertész, and A-L Barabási. Structure and tie strengths in mobile communication networks. *Proceedings of the national academy of sciences*, 104 (18):7332–7336, 2007.

Mathew Penrose. *Random geometric graphs*, volume 5. OUP Oxford, 2003.

Mason Alexander Porter, Jukka-Pekka Onnela, Peter J Mucha, et al. Communities in networks. 2009.

Derek J De Solla Price. Networks of scientific papers: The pattern of bibliographic references indicates the nature of the scientific research front. *Science*, 149 (3683):510–515, 1965.

Nataša Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.

Natasa Pržulj, Derek G Corneil, and Igor Jurisica. Modeling interactome: scale-free or geometric? *Bioinformatics*, 20(18):3508–3515, 2004a.

Natasa Pržulj, Derek G Corneil, and Igor Jurisica. Modeling interactome: scale-free or geometric? *Bioinformatics*, 20(18):3510, 2004b.

Selim Reza, Marta Campos Ferreira, JJM Machado, and João Manuel RS Tavares. Road networks structure analysis: A preliminary network science-based approach. *Annals of mathematics and artificial intelligence*, pages 1–20, 2022.

Pedro Ribeiro, Pedro Paredes, Miguel EP Silva, David Aparicio, and Fernando Silva. A survey on subgraph counting: concepts, algorithms, and applications to network motifs and graphlets. *ACM Computing Surveys (CSUR)*, 54(2): 1–36, 2021.

Tiago Rito, Zi Wang, Charlotte M Deane, and Gesine Reinert. How threshold behaviour affects the use of subgraphs for network comparison. *Bioinformatics*, 26(18):i611–i617, 2010.

George Ritzer. *The McDonaldization of Society : an Investigation into the Changing Character of Contemporary Social Life*. Pine Forge Press, 1996.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL `https://networkrepository.com`.

Anida Sarajlić, Noël Malod-Dognin, Ömer Nebil Yaveroğlu, and Nataša Pržulj. Graphlet-based characterization of directed networks. *Scientific reports*, 6(1): 1–14, 2016.

Comandur Seshadhri, Tamara G Kolda, and Ali Pinar. Community structure and scale-free collections of erdős-rényi graphs. *Physical Review E*, 85(5):056109, 2012.

Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *Artificial intelligence and statistics*, pages 488–495. PMLR, 2009.

Ryan W Solava, Ryan P Michaels, and Tijana Milenković. Graphlet-based edge clustering reveals pathogen-interacting proteins. *Bioinformatics*, 28(18):i480–i486, 2012.

Ulrich Stelzl, Uwe Worm, Maciej Lalowski, Christian Haenig, Felix H Brembeck, Heike Goehler, Martin Stroedicke, Martina Zenkner, Anke Schoenherr, Susanne Koeppen, et al. A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, 122(6):957–968, 2005.

Michael PH Stumpf, Carsten Wiuf, and Robert M May. Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proceedings of the National Academy of Sciences*, 102(12):4221–4224, 2005.

Gábor J Székely and Maria L Rizzo. Brownian distance covariance. *The annals of applied statistics*, pages 1236–1265, 2009.

Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. 2007.

Reiko Tanaka, Tau-Mu Yi, and John Doyle. Some protein interaction data do not exhibit power law statistics. *FEBS letters*, 579(23):5140–5144, 2005.

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.

Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. In *Social networks*, pages 179–197. Elsevier, 1977.

Robert E Ulanowicz and Donald L DeAngelis. Network analysis of trophic dynamics in south florida ecosystems. *FY97: The Florida Bay Ecosystem*, pages 20688–20038, 1998.

Ivan Voitalov, Pim Van Der Hoorn, Remco Van Der Hofstad, and Dmitri Krioukov. Scale-free networks well done. *Physical Review Research*, 1(3):033034, 2019.

Stanley Wasserman and Katherine Faust. Social network analysis in the social and behavioral sciences. *Social network analysis: Methods and applications*, 1994:1–27, 1994.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

Bernard M Waxman. Routing of multipoint connections. *IEEE journal on selected areas in communications*, 6(9):1617–1622, 1988.

Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. *Nauchno-Technicheskaya Informatsia, Series*, 2(9):12–16, 1968.

Wikipedia. Shrikhande graph — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Shrikhande%20graph&oldid=1091006448`, 2023. [Online; accessed 11-July-2023].

Ömer Nebil Yaveroğlu, Noël Malod-Dognin, Darren Davis, Zoran Levnajic, Vuk Janjic, Rasa Karapandza, Aleksandar Stojmirovic, and Nataša Pržulj. Revealing the hidden language of complex networks. *Scientific reports*, 4(1):4547, 2014.

Deukryeol Yoon, Dongjin Lee, Minyoung Choe, and Kijung Shin. Graphlets over time: A new lens for temporal network analysis. *arXiv preprint arXiv:2301.00310*, 2023.

Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

Luming Zhang, Mingli Song, Zicheng Liu, Xiao Liu, Jiajun Bu, and Chun Chen. Probabilistic graphlet cut: Exploiting spatial structure cue for weakly supervised image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1908–1915, 2013.

Shi Zhou and Raúl J Mondragón. Accurately modeling the internet topology. *Physical review E*, 70(6):066108, 2004.

List of Figures

List of Tables

A. Attachments

A.1 Graphlet degree distribution of networks

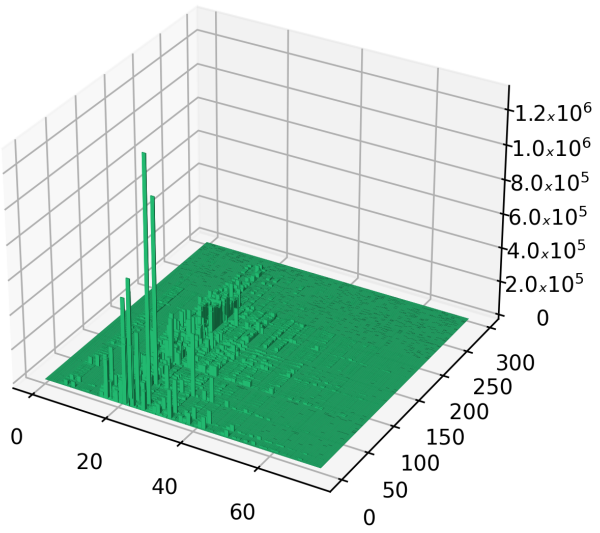


Figure A.1: Graphlet degree distribution of the neural network of *C. elegans* - Original network

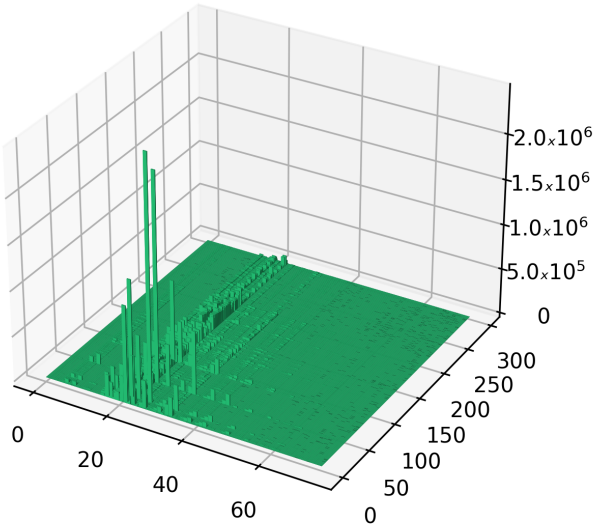


Figure A.2: Graphlet degree distribution of the neural network of *C. elegans* - Albert-Barabási model

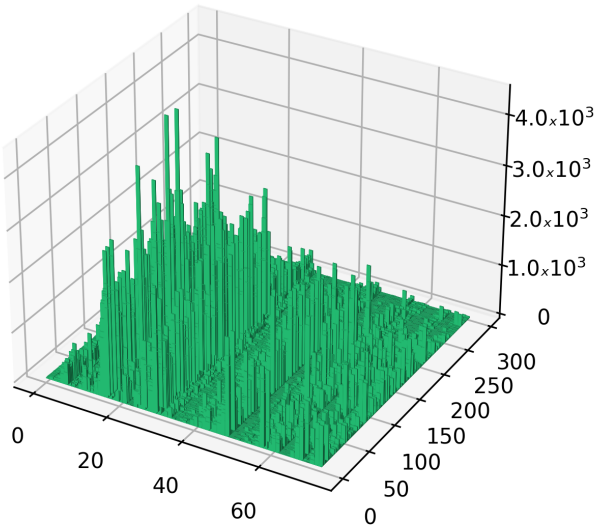


Figure A.3: Graphlet degree distribution of the neural network of *C. elegans* - Geometric model

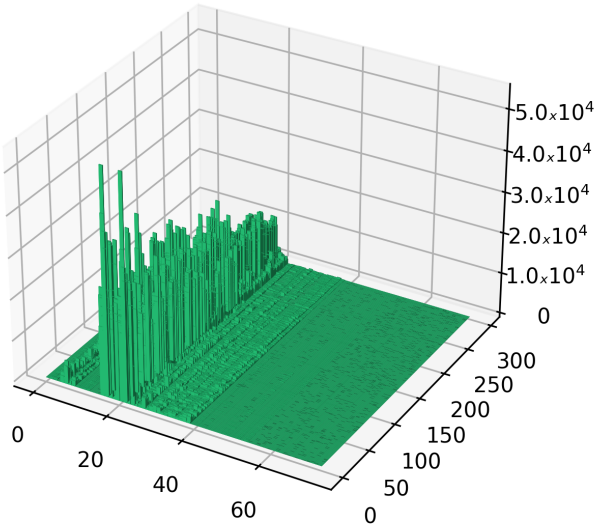


Figure A.4: Graphlet degree distribution of the neural network of *C. elegans* - Erdős-Rényi model

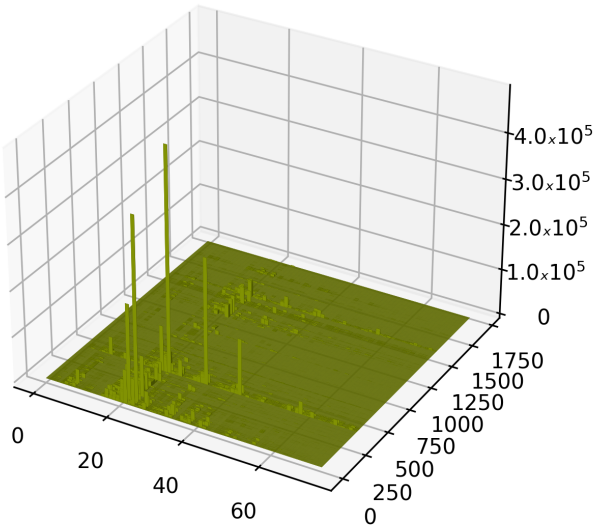


Figure A.5: Graphlet degree distribution of the interaction between genes in an unnamed plant - Original network

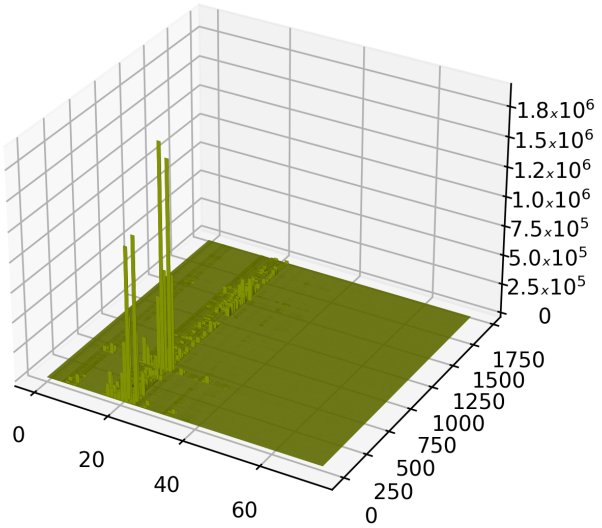


Figure A.6: Graphlet degree distribution of the interaction between genes in an unnamed plant - Albert-Barabási model

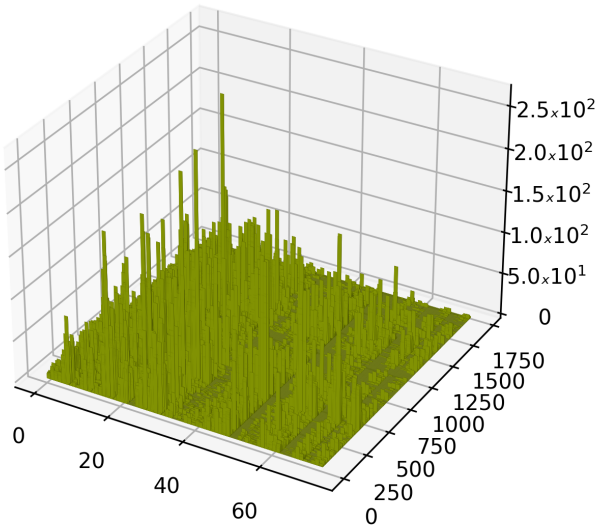


Figure A.7: Graphlet degree distribution of the interaction between genes in an unnamed plant - Geometric model

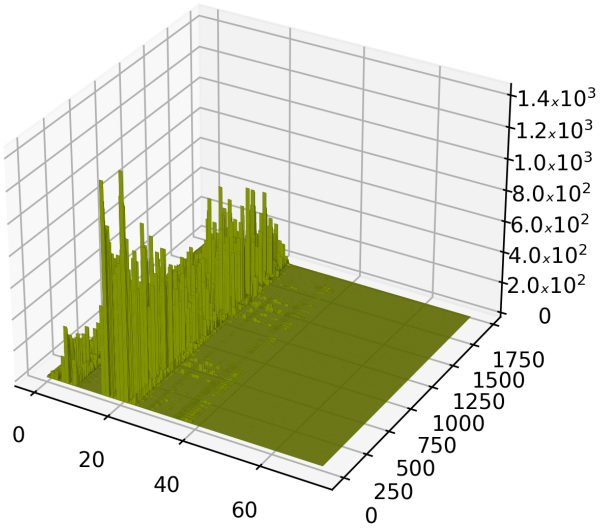


Figure A.8: Graphlet degree distribution of the interaction between genes in an unnamed plant - Erdős-Rényi model

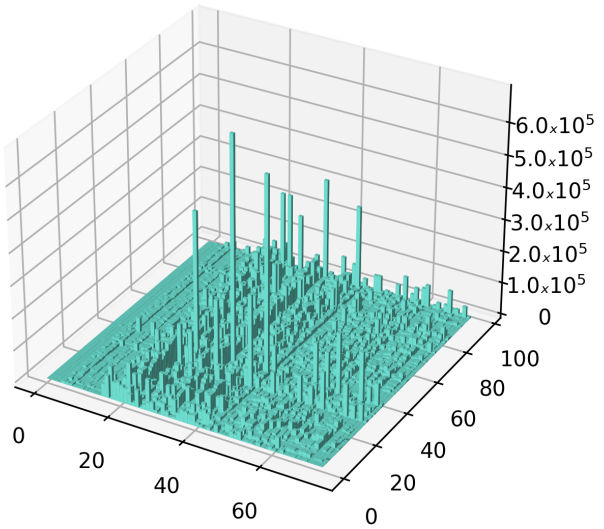


Figure A.9: Graphlet degree distribution of the food web in tropical ecosystems of Florida - Original network

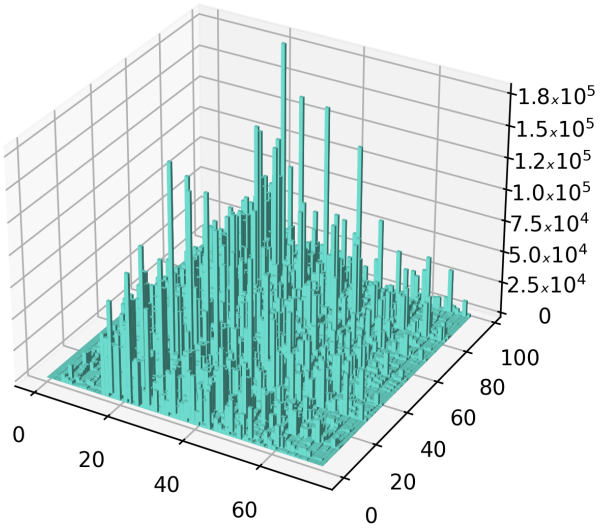


Figure A.10: Graphlet degree distribution of the food web in tropical ecosystems of Florida - Albert-Barabási model

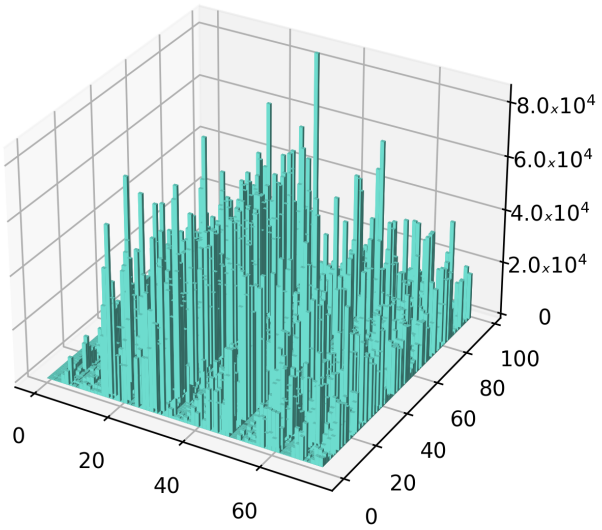


Figure A.11: Graphlet degree distribution of the food web in tropical ecosystems of Florida - Geometric model

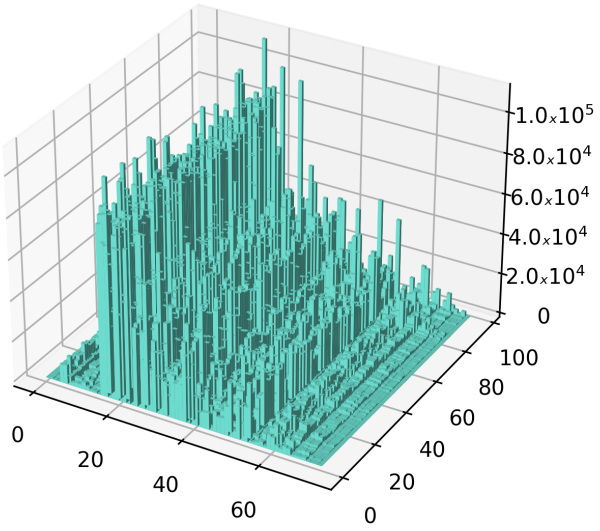


Figure A.12: Graphlet degree distribution of the food web in tropical ecosystems of Florida - Erdős-Rényi model

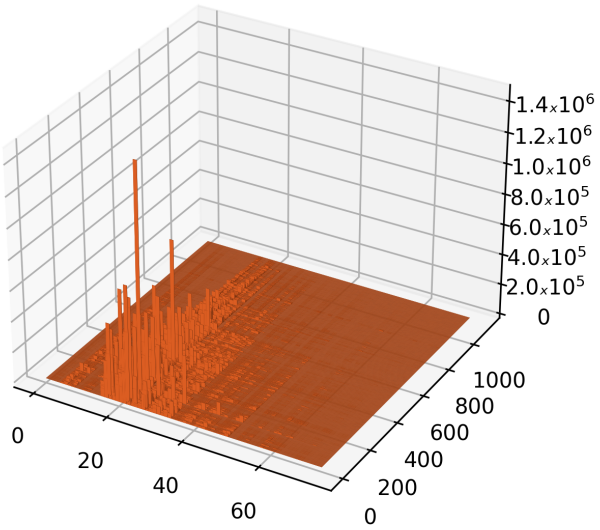


Figure A.13: Graphlet degree distribution of the email communication network - Original network

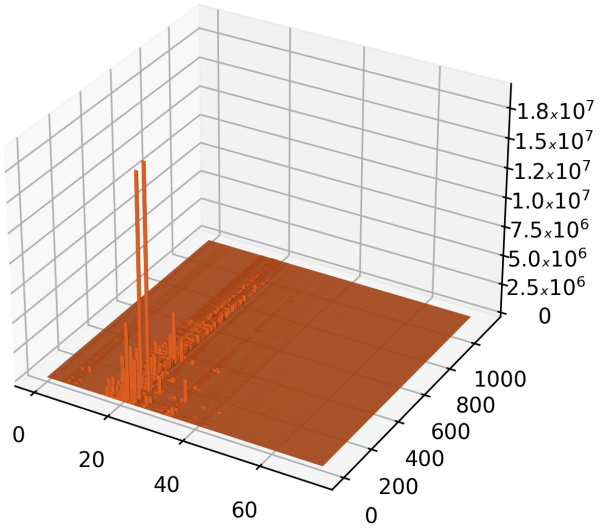


Figure A.14: Graphlet degree distribution of the email communication network - Albert-Barabási model

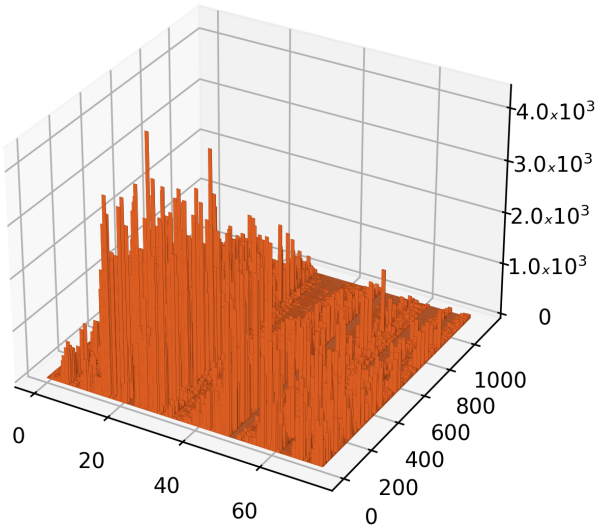


Figure A.15: Graphlet degree distribution of the email communication network - Geometric model

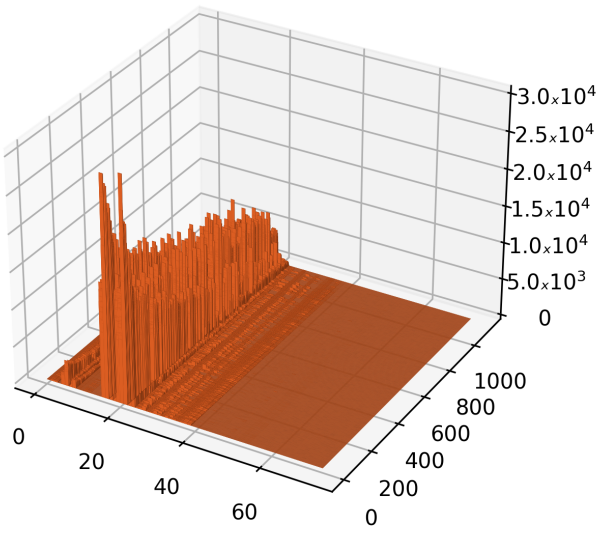


Figure A.16: Graphlet degree distribution of the email communication network - Erdős-Rényi model