Analysis of the Structural Properties and Scalability of Complex Networks

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Braulio Gabriel Dumba

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Prof. Zhi-Li Zhang

July, 2018

c Braulio Gabriel Dumba 2018
ALL RIGHTS RESERVED

Acknowledgements

There are many people that have earned my gratitude for their contribution to my time in graduate school.

First of all, I would like to thank my adviser, Professor Zhi-Li Zhang, for his continuous support, criticism, and guidance on conducting research and achieving personal long-term career goals.

I thank my lab mates for collaborating with me on a number of exciting research projects that greatly enriched my research experience. In particular, Golshan Golnari, Hesham Mekky, Cheng Jin, Saurabh Verma, Guobao Sun and Eman Ramadan.

I thank all my fellow graduate students working with Professors Zhi-Li Zhang, Tian He, and David Du for their valuable feedback on my research during our group meetings and for creating and maintaining a friendly and intellectually creative environment.

I thank Professors Georgios Giannakis, Abhishek Chandra, Jaideep Srivastava for their support, feedback and advice as my PhD thesis (or oral) committee members.

Finally, I am thankful to my family for supporting me over the years.

My research was supported by various sources of funds: Raytheon/NSF subcontract 9500012169/CNS-1346688, DoD ARO MURI Award W911NF-12-1-0385, DTRA grant HDTRA1-14-1-0040, HDTRA1- 09-1-0050, NSF grant CNS-1117536, CRI-1305237, CNS-1411636, CNS-1618339 and CNS-1617729.

Dedication

To my family

Abstract

Many real-world systems around us can be described as complex networks (e.g., electric grids, cyber-physical systems, chemical and energy systems). Hence, there has been a quickly growing interest in such networks, since they can help us to represent, analyze and evaluate many of the complex and dynamic systems that have become a critical resource in our daily and social life (i.e., the Internet and the World Wide Web, online social networks, road networks, etc). Complex networks have been studied in different contexts (i.e., communities extraction, path lengths, cluster coefficient and degree distributions, small-world networks, etc.) for a long time. However, our understanding of the possible organizing principles shaping the observed topology of complex networks is still in its infancy. In this dissertation, we advance the current knowledge in understanding the topology and formation of complex systems. More specifically, we explore the concept of "reciprocal network" and present new methods to "uncover" and "dissect" the core structure of complex networks with the goal of improving our understanding of such systems.

First, we present a comprehensive measurement-based characterization of the reciprocal network extracted from a directed complex network – using the online social network Google+ as a case study – and its evolution over time, with the goal to gain insights into the structural properties of a complex network. In a sense, the reciprocal network can be viewed as the stable skeleton network of a directed network that holds it together. Thus, it could reveal the possible organizing principles shaping the observed network topology of a directed complex network.

Second, we have advanced and developed an effective procedure to extract the core structure of complex networks. To achieve this, we propose two new metrics a node "dependence value" and a subgraph "nucleon-index". Then, using these metrics, we proposed a modified version of the traditional k-shell decomposition method by identifying the k_C-index where we should stop pruning the network in order to preserve its core structure and extract a meaningful "core" for complex networks.

Third, with the goal of dissecting the structure of the nucleus of a massive complex network, we propose a two-step procedure to hierarchically unfold the nucleus of complex

networks by building up and generalizing ideas from the existing clique percolation approaches. Our scheme builds (hyper)graphs that provide us with a "big picture" view of the core structure of a complex network and how it is formed. Our methodology is very scalable and can be applied to massive complex networks (hundreds million nodes and billion edges).

In summary, this thesis proposes new tools to understand the structural properties and formation of complex networks. Our developed schemes are capable of: i) helping to understand possible organizing principles shaping the observed network topology of a directed complex network; ii) extracting the core structure of complex networks; and iii) dissecting the structure of the dense nucleus of massive complex networks.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Many real-world systems in biology, neuroscience, physics, engineering and social science can be described as complex networks (e.g., electric grids, cyber-physical systems, and chemical and energy systems). There has been a quickly growing interest in such networks, since they can help us to represent, understand and evaluate many of the complex and dynamic systems around us. Today's internet and social networks are example of complex networks that have become a critical resource in our daily and social life: interacting with people, processing information, and diffusing social influence.

Complex networks have been studied in different contexts for a long time [2, 3, 4, 5, 6, 7, 8, 9]. Many of these studies focus on detecting the underlay structures of complex systems by finding subnetworks (e.g., communities, core-periphery) in order to understand the topology of complex networks; while others focus on highlighting statistical properties (e.g., path length, diameter, density and degree distributions) that characterize the structure and behavior of networked systems, and on creating models of networks that can help us to understand the properties of complex networks. However, our understanding of the possible organizing principles shaping the observed network topology of complex networks is still in its infancy. In the words of E. O. Wilson 18 , "The greatest challenge today, not just in cell biology and ecology but in all of science, is the accurate and complete description of complex systems" [10].

This thesis spans the areas of methodologies and algorithms to understand the topological organizing principles and formation of complex network systems. Why is network

structure/topology so important to characterize? Because structure always affects function. For instance, the topology of social networks affects the spread of information and disease, and the topology of the power grid affects the robustness and stability of power transmission [15]. To this end, this thesis addresses the following closely related problems. First, we investigate the topological structure of complex network by focusing on concept of "reciprocal network". Second, we propose a novel procedure to uncover the "nucleus" of complex network in order to understand their formation. Third, we design a algorithm to dissect the dense structure of the nucleus of massive complex networks to help us further understanding the nucleus of complex networks and develop tool/algorithms to take advantage of this structure.

1.1 Thesis Statement

The central thesis of this dissertation is as follows:

Complex networks have become a critical resource in our daily life for a long time. However, our deep understanding of the organizing principles shaping the observed topology of complex networks is still in its infancy.

This thesis explores new concepts and develops algorithms that could reveal possible mechanism of social, biological or different nature that systematically acts as organizing principles shaping the observed network topology of complex networks. We specically focus on three key points: reciprocal network, network core extraction and network core dissection.

1.2 Outline and Contributions

This dissertation studies the reciprocal network, core extraction and dissection of complex networks separately. The outline of this dissertation, along with the primary contributions of this dissertation are as follows:

Reciprocal Networks and their Evolution (Chapter 3). Many complex networks such as Twitter,Google+, Flickr and Youtube are directed in nature, and have been shown to exhibit a nontrivial amount of reciprocity. Reciprocity is defined as the ratio of the number of reciprocal edges to the total number of edges in the network, and has

been well studied in the literature. However, little attention is given to understand the connectivity or network form by the reciprocal edges themselves (reciprocal network), its structural properties, and how it evolves over time. In this chapter, we bridge this gap by presenting a comprehensive measurement-based characterization of the connectivity among reciprocal edges in a directed complex network – using the online social network (OSN) Google+ as case study – and their evolution over time, with the goal to gain insights into the structural properties of a complex network. In a sense, the reciprocal network can be viewed as the stable skeleton network of a directed network that holds it together. Thus, they could reveal the possible organizing principles shaping the observed network topology of a directed complex network. Moreover, understanding the dynamic structural properties of the reciprocal network provides us with additional information to characterize or compare directed networks that go beyond the classic reciprocity metric, a single static value currently used in many studies.

Uncovering the Nucleus of Complex Networks (Chapter 4). Many complex network studies have focused on identifying communities through clustering or partitioning a large complex network into smaller parts. While community structure is important in complex network analysis, relatively little attention has been paid to the problem of core structure analysis in many complex networks. Intuitively, one may expect that many complex networks possess some sort of a core which holds various parts of the network (or constituent communities) together. We believe that it is just as important to uncover and extract the core structure referred to as the nucleus in this paper of a complex network as to identify its community structure. In this chapter, we have advanced and developed an effective procedure to extract the core structure of complex networks. To achieve this, we introduce a new metric the node "dependence value" that measures the location importance of a node in a network. Then, we define a new measure called "nucleon-index" that captures the extend to which a subgraph is a densely intra-connected and topological central core. Then, using these metrics, we proposed a modified version of the traditional k-shell decomposition method by identifying the k_C-index where we should stop pruning the network in order to preserve its core structure and extract a meaningful "core" for complex networks.

Dissecting the Nucleus of Massive Complex Networks using (Hyper)Graphs (Chapter 5). In this chapter, with the goal of dissecting the structure of the nucleus

of a massive complex network (using Google+ reciprocal network as a case study), we propose a two-step procedure to hierarchically unfold the nucleus of complex networks by building up and generalizing ideas from the existing clique percolation approaches. Using maximal cliques as the basic atomic structures of the network nucleus, we build (hyper)graphs that provide us with a higher-level representation of the dense core graph of complex networks. Hence, our scheme provides a "big picture view of the core structure of a complex network and how it is formed. Our methodology is very scalable and can be applied to massive complex networks (hundreds million nodes and billion edges).

This thesis proposes new tool to understand the structural properties and formation of complex networks. Our developed schemes are capable of: i) helping to understand possible organizing principles shaping the observed network topology of a directed complex network; ii) extracting the core structure of complex networks; and iii) dissecting the structure of the dense nucleus of massive complex networks.

The remainder of this dissertation introduces background and motivation (Chapter 2); presents our comprehensive measurement-based characterization of the connectivity among reciprocal edges in a directed complex network (Chapter 3); presents our effective procedure to extract the core structure of complex networks (Chapter 4); presents our two-step procedure to hierarchically unfold the nucleus of massive complex networks (Chapter 5); discusses future directions and finally concludes (Chapter 6).

1.3 Bibliographic Notes

Part of the contents of Chapter 3 on studying reciprocal networks and their evolution is from a conference paper, titled "Analysis of a Reciprocal Network Using Google+: Structural Properties and Evolution", which appeared in the Proceedings of the 5th International Conference on Computational Social Networks (CSoNet'16), Ho Chi Minh City, Vietnam, August 2-4, 2016 [11]. Our developed effective procedure to extract the core structure of complex networks is presented in a conference paper titled "Uncovering the Nucleus of Social Networks", which appeared in the Proceedings of the 10th ACM Conference on Web Science (WebSci'18), May 27-30, 2018, Amsterdam, Netherlands [12]. This constitutes part of Chapter 4. Part of the contents of Chapter 5 are from two papers titled "Uncovering the Nucleus of a Massive Reciprocal Network",

which appeared on the World Wide Web Journal - Special issue on “Social Computing and Big Data Applications, (2018) [13] and another paper titled “Unfolding the Core Structure of the Reciprocal Graph of a Massive Online Social Network.”, which appeared on the Proceedings of the 10th Annual International Conference on Combinatorial Optimization and Applications (COCOA’16), Hong Kong, China, December 16-18, 2016” [14].

Chapter 2

Background and Motivation

Complex networks (i.e. networks with non-trivial topological features [1]) are a fundamental tool to represent and model the structure of many real-world complex systems. These include the Internet [33, 32], World Wide Web [42], mobile phone [59], collaboration [28] and citation [21] networks, but also systems of interest in biology, physics, neuroscience and statistics. Today, many complex systems have become a critical resource in our daily and social life. For example, the Internet is arguably the largest complex network ever created by mankind. The Internet is a computer network which consist of millions of switches/routers and communication links. It interconnects hundreds of millions of hosts or end systems throughout the world: PCs, PDAs, laptops, sensors, webcams, game consoles, picture frames, cellphones, automobiles, home electrical and security devices, etc. Similar to today's Internet, social networks (e.g., Facebook, Twitter, Google+) are another example of complex networks that have become a critical resource in our daily or social life – users represent vertices or nodes and edges capture specic relations (e.g., friendship and co-authorship). In fact, social networking became the most popular online activity worldwide [50].

There has been a quickly growing interest in the study of complex networks, since they can provides us with new insight into a vast array of complex and previously poorly understood phenomena in many of the complex and dynamic systems around us. Thus, toward understanding complex network researchers have identified a series of unifying principles and statistical properties common to most of the real network. For

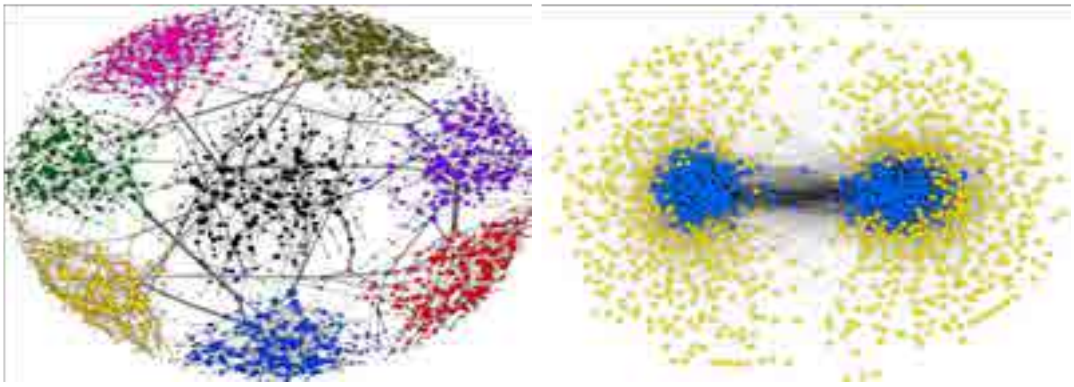
Figure 2.1: Examples of parasocial (one-way edge) and reciprocal (bi-directional edges) edges

example, node degree, network diameter, path length, cluster coefficient, density, reciprocity, community and core-periphery structures, etc. Furthermore, a lot of work has been devoted on building mathematical modeling of networks, including random graph models and their generalizations, exponential random graphs, p-models and Markov graphs, the small-world model and its variations, and models of growing graphs including preferential attachment models and their many variations [15]. However, the study of networks is by no means a complete science yet, our understanding of the possible organizing principles shaping the observed network topology of complex networks is still in its infancy.

In this dissertation, we advanced the currently knowledge in understanding the topology and formation of complex network systems. More specifically, we explore the concept of *reciprocal network* and advance the knowledge on the methods to "uncover" and "dissecting" the *core structure* of complex networks with the goal of revealing the possible organizing principles shaping the observed network topology of complex networks. In the rest of this chapter, we summarize the background necessary for the following chapters and provide some motivation examples.

2.1 Reciprocity in Complex Networks

Many online social networks are fundamentally *directed*: they consist of both *reciprocal* edges, i.e., edges that have already been linked back, and *parasocial* edges, i.e., edges have not been or is not linked back [16] – see Fig. 2.1. Reciprocity is defined as the ratio of the number of reciprocal edges to the total number of edges in the network, and it is believed that it plays an important role in the structural properties, formation and evolution of online social networks. Hence, this metric has been widely studied



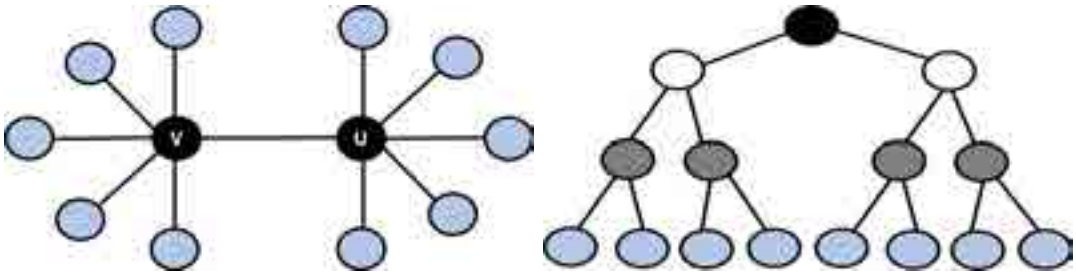
(a) Community (b) Core-periphery

Figure 2.2: Example networks with community structure and a core-periphery structure.

in the literature in various contexts, see, e.g., [16, 17, 18, 19, 20, 21]. For example, it has been used to compare and classify different directed networks, e.g., reciprocal or anti-reciprocal networks[17]. The authors in [16] investigate the factors that influence parasocial edges to become reciprocal ones. The problem of maximum achievable reciprocity in directed networks is formulated and studied in [18], with the goal to understand how bi-degree sequences (or resources or "social bandwidth") of users determines the reciprocity observed in real directed networks. The authors in [19] propose schemes to extract meaningful sub-communities from dense networks by considering the roles of users and their respective connections (reciprocal versus non-reciprocal ties). The authors in [20] examine the evolution of reciprocity and speculate that its evolution is affected by the hybrid nature of Google+, whereas the authors in [21] conduct a similar study and conclude that Google+ users reciprocated only a small fraction of their edges: this was often done by very low degree users with no or little activity. However, many studies have used reciprocity (a single-valued aggregate metric) to characterize massive *directed* OSNs, which we believe is inadequate (more in Chapter 3).

2.2 The Nucleus of Complex Networks

Many complex networks have both a community structure and a core-periphery structure [22, 23, 24]. Community is often considered to be a subset of vertices that are



(a) double star graph (b) binary tree

Figure 2.3: Example networks of double star and binary tree graphs. These structures cannot be decomposed using k-core decomposition.

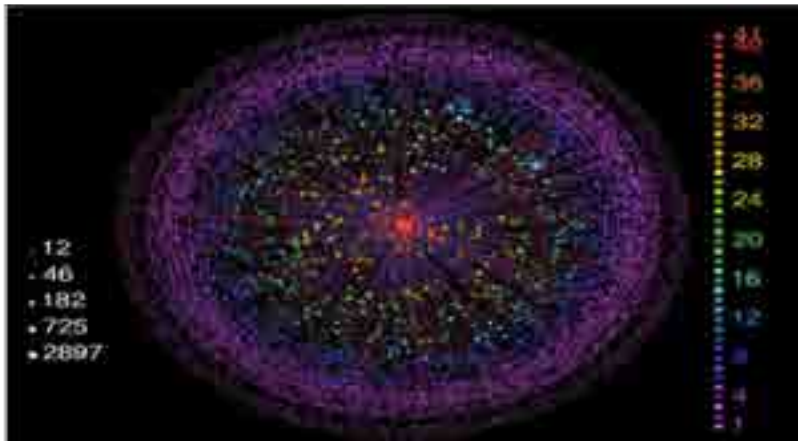


Figure 2.4: Visualization of the Internet at the AS level [1] using k-shell (constructed using the k-shell decomposition method).

densely connected internally but sparsely connected to the rest of the network [25, 26, 27, 28, 29] and it has received a lot of attention in the literature. This structure sometimes used to decompose large network into smaller components in order to control or manage dynamic system. However, not all "communities" are created equal in a network – some of them may overlap to form the core structure of a complex system: "a super community" (network core or nucleus). The core nodes in complex networks are fundamental for the structural properties of the network. Consequently, finding these nodes (or core-periphery structure) within a complex network is a powerful tool for understanding the functioning of a complex system, as well as for identifying a hierarchy of connections and understanding the organizing principles shaping the observed network topology (*top-down* or *bottom-up*[1] process). Figure 2.2 illustrates example networks with community and core-periphery structures.

One of the most popular quantitative methods to investigate core-periphery structure was proposed by Borgatti and Everett in 1999 [30]. Based on this study, several methods for identifying the core-periphery of a network have been proposed [31, 32, **?**]. These algorithms attempt to determine which nodes are part of a densely-connected core and which are part of a sparsely connected periphery by solving some complex optimization problem. In contrast, some studies simply define the network "core" as the maximal clique composed of the highest degree nodes in a network [33], while other studies focus instead on some notion of connectivity information (e.g. betweenness, closeness, etc.) to find the core and periphery of a network [31, 22, 32, 34, 35]. Consequently, most of these methods are computationally expensive and do not scalable to large networks.

The authors in [36] used the notion of α-β community to extract the "core" of a graph. An α-β community is a connected subgraph C with each vertex in C connected to at least β vertices of C and each vertex outside of C connected to at most α vertices of C ($\alpha < \beta$). They extract the network core structure by taking the intersection of α-β communities of different size k. A core thus corresponds to one or multiple dense regions of the graph. As a result, the proposed heuristics in [36] may return multiple dense regions ("cores") for a given network. In addition, this algorithm does

[1]top-down: existing networks expand by adding "new branches"; bottom-up: existing networks are interconnected (by adding new links or building a new interconnection network).

not guarantee to terminate within a reasonable amount of running time.

The authors in [**?**] propose the k-core decomposition to discover interesting structural properties of networks. A k-core of G is a subgraph G^* obtained by recursively removing all the vertices of degree less than k, until all the vertices in the remaining graph have degree at least k. This method is very scalable and it has a time complexity similar to the k-shell decomposition for general graphs: $(O(V + E))$. However this method is unable to uncover the structural properties for certain type of graphs or substructures. For example, a double star-like graph S formed by two connected vertices v and u with high degrees that connect many vertices with degree one cannot be decomposed beyond 1-shell (or 1-core), containing all the vertices in graph S, no matter how high are the degree of the vertices v and u. Similarly, a binary tree graph T cannot be decomposed beyond the first shell, independently of the depth of the tree T – see Fig. 2.3 for an illustration.

In [1] the authors proposed the "k-shell decomposition method", one of the most popular and scalable method to investigate and visualize the core-periphery structure in complex networks. Different from k-core decomposition, in this method at each step k, we prune vertices of degree k or less. This method has been successfully used as a visualization tool for studying and uncovering the core structure of networks such as the Internet AS graph [1](see Fig. 2.4). However, this method is unable to uncover the core structure of some complex networks (more on Chapter 4).

2.3 Implications of Finding the Nucleus of a Network

In this section, we discuss the implications of uncovering the nucleus of complex networks. While the implications are likely applicable to many different applications, we concentrate on their effect on network formation, design, robustness and control:

Network Formation: A network core gives a well-defined starting point and a way to explore the network topology systematically. For example, a network can be reconstructed layer by layer from the core to its periphery. Then, topological features of the nodes and structural properties of the network can be measured at each layer. Furthermore, using the core, we can build macroscopic models of the network that can help us predict the topological growth of the network and provide good upper bounds of the

distance between the nodes – see the jellyfish model of the Internet in [33]. Therefore, unveiling the core structure of networks can help us uncover and understand possible organizing principles shaping the observed network topological structure and network formation.

Network Design: Observing the evolution patterns of the core structure of social networks can give insights for the design of future social networks by other social networking service providers who would like to enter the market. Furthermore, it can also help applications for social networks to be designed to take advantage of the network core properties.

Network Robustness: Robustness is often defined as the ability of a network to continue to function when it is subject to failures. Uncovering the core structure of networks is fundamental in the development of techniques for analyzing the vulnerability or robustness of networks. For example, in Google+ the tight core coupled with high link reciprocity implies that users in the core appear on large number of the shortest paths in the network. Thus, if malicious users are able to penetrate the core, they can destroy or remove the hubs of information flow (core nodes) in the network. Hence, disrupting the functionally of the network. Then, by strengthening the defenses in the core subgraph, we can increase the robustness of the social networks.

Control Dynamics Systems: Dynamic network decomposition has been studied in different contexts for a long time [37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]. The key of many decomposition methods is to detect the underlaying structures of the network by finding subnetworks with significantly more links between the nodes inside than across them. To find such subnetworks, the well-known concept of "community" structure has been used to systematically decompose the network into subnetworks. Then, these subnetworks are used to implement distributed control schemes by assigning controllers to these structures, with the corresponding controllers coordinated through some level of information sharing. Thus, the stability and manageability of the entire system can be guaranteed by stabilizing these subnetworks. This technique (referred to as "distributed control") has been widely used to control large networked systems (e.g., electric grids [50], cyber-physical systems [51], and chemical and energy systems [52]). Dynamics networks have a core-periphery structure. Thus, the identification of such structure is crucial for improving the control of these complex systems.

Chapter 3

Reciprocal Networks and their Evolution

3.1 Introduction

It has been shown that major online social networks (OSN) that are directed in nature, such as Twitter, Google+, Flickr and Youtube, all exhibit a nontrivial amount of reciprocity: for example, the global reciprocity of Flickr [53], Youtube [53], Twitter [54] and Google+[55] have been empirically measured to be 0.62, 0.79, 0.22 and 0.32, respectively. Reciprocity has been widely studied in the literature in various contexts, see, e.g., [16, 17, 18, 19, 20, 21]. Reciprocal edges represent the most stable type of connections or relations in directed network – they reflect strong ties between nodes or users [56, 57, 58], such as (mutual) friendships in an online social network or "following" each other in a social media network like Twitter. Connectivity among reciprocal edges can thus potentially reveal more information about users in such networks. For example, a clique formed by reciprocal edges suggest users involved are mutual friends or share common interests. More generally, it is believed that nontrivial patterns in the *reciprocal network* – the bidirectional subgraph (see Figure 3.1) of a directed graph could reveal possible mechanism of social, biological or different nature that systematically acts as organizing principles shaping the observed network topology [17]. Moreover, understanding the dynamic structural properties of the reciprocal network can provide

us with additional information to characterize or compare directed networks that go beyond the classic reciprocity metric, a single static value currently used in many studies. However, little attention has been paid in the literature to understand the connectivity between reciprocal edges – the reciprocal network – and how it evolves over time.

In this paper we perform a comprehensive measurement-based characterization of the connectivity and evolution of reciprocal edges in Google+ (thereafter referred to as *G+* in short), in order to shed some light on the structural properties of G+'s reciprocal network. We are particularly interested in understanding how the reciprocal network of G+ evolves over time as new users (nodes) join the social network, and how reciprocal edges are created, e.g., whether they are formed mostly among extant nodes already in the system or by new nodes joining the network. For this, we employ a unique massive dataset collected in a previous study [21]. We start by providing a brief overview of G+ and a description of our dataset in Section 3.2. We then present our methodology to extract the reciprocal network of G+ using Breadth-First-Search (BFS), together with some notations in Section 3.3. In Section 3.4.1, we discuss a few key aggregate properties of the reciprocal network including the growth of the numbers of nodes and edges over time, the in-degree, out-degree, and reciprocal or mutual degree distributions. We then analyze the evolution of the reciprocal network in terms of its density, and categorize the nodes joining the reciprocal network based on the (observed) time they joined the network in Sections 3.4.2, and study the types of connections they make (reciprocal edges) in Section 3.4.3. Finally we discuss the implications of our findings and we conclude the paper in Section 3.5. We summarize the major findings of our study as follows:

- We find that the density of G+ – which reflects the overall *degree of social connections* among G+ users – decreases as the network evolves from its second to third year of existence. This finding differs from the observations reported in [20], where it found that G+ social density fluctuates in an increase-decrease fashion in three phases, but it reaches a steady increase in the last phase during its first year of existence.

- Furthermore, we observe that both the density and reciprocity metrics of G+'s reciprocal network also decrease over time. Our analysis reveals that these are

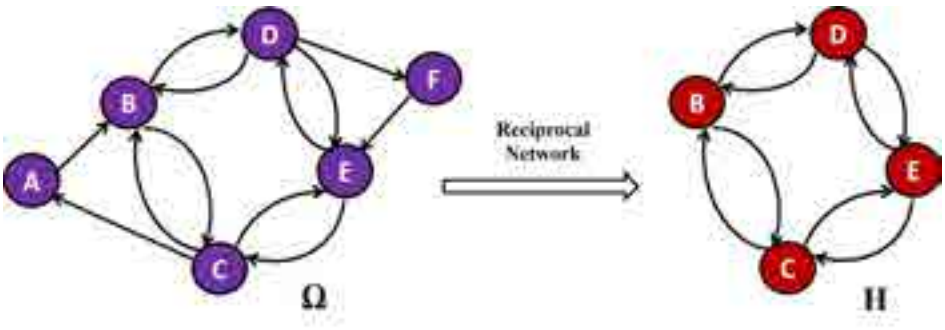


Figure 3.1: Illustration of the reciprocal network (H^{i+1}) of a directed graph (Ω^{i+1}). Specifically, (B, C), (C, B), (B, D), (D, B), (D, E), (E, D), (C, E), (E, C) are reciprocal edges; (A, B), (C, A), (D, F), (F, E) are parasocial edges. The reciprocity of Ω^{i+1} is $8/12 = 0.67$

due to the fact that the new users joining G+ later tend to be less "social" as they make fewer connections in general. In particular, i) the number of users creating at least one reciprocal edge is decreasing as the network evolves; ii) the new users joining the reciprocal network are creating fewer edges than the users in the previous generation.

- We show that if a user does not create a reciprocal edge when he/she joins G+, there is a lower chance that he/she will create one later. In addition, users who already have reciprocal connections with some users tend to create more reciprocal connections with additional users.

To the best of our knowledge, our study is the first study on the properties and evolution of a "reciprocal network" extracted from a *directed* social graph.

3.2 Google+ Overview and Dataset

In this section, we briefly describe key features of the Google+ service and a summary of our dataset.

Platform Description:June 2011 Google launched its own social networking service called Google+ (G+). The platform was announced as a new generation of social network. Previous works in the literature [20, 21] claim that G+ cannot be classified as

particularly asymmetric (Twitter-like), but it is also not as symmetric (Facebook-like) because G+ features have some similarity to both Facebook and Twitter. Therefore, they labelled G+ as a hybrid online social network[20]. Similar to Twitter (and different from Facebook) the relationships in G+ are unidirectional. In graph-theoretical terms, if user[1] x follows user y this relationship can be represented as a directed social edge (x, y); if user y also has a directed social edge (y,x), the relationship x, y is called symmetric[**?**]. Similar to Facebook, each user has a stream, where any activity performed by the user appears (like the Facebook wall). For more informations about the features of G+ the reader is referred to [59, 60].

Dataset: we obtained our dataset from an earlier study on G+ [21], so no proprietary right can be claimed. The dataset is a collection of 12 directed graphs of the social links of the users[2] in $G+$, collected from August, 2012 to June, 2013. We used BFS to extract the Largest Weakly Connected Component(LWCC) from all of our snapshots of G+. We label these set of LWCCs as subgraphs Ω^i (for $i = 1, ..., 12$). Since LWCC users form the most important component of G+ network[21], we extract the reciprocal network of G+ from the Ω^i subgraphs (see Sect. 3.3). However, for consistency in our analysis, we removed from the subgraphs $\Omega^{i=1,...11}$ those nodes that do not appear in our last snapshot at Ω^{12}. Table 1 summarizes the main characteristics of the extracted Ω^i.

3.3 Methodology & Basic Notations

In this section, we describe our methodology to extract the reciprocal network of G+. To derive the reciprocal network of G+, we proceed as follows: we extract the subgraphs composed of nodes with at least one reciprocal edge for each of the snapshots of Ω^i. We label these new subgraphs G^i (for $i = 1, 2, ..., 12$). By comparing the set of nodes and edges in each of the sugbraphs G^i, we observe that a very small percentage of nodes depart G^i as it evolves (*unfollowing behaviour*[**?**]). Therefore, for consistency in our analysis, we removed from the subgraphs $G^{i=1,...11}$ those nodes that don't appear in our last snapshot at G^{12}. We label these new set of subgraphs L^i (for $i = 1, 2, ..., 12$).

[1] In this paper we use the terms "user" and "node" interchangeable

[2] G+ assigns each user a 21-digit integer ID, where the highest order digit is always 1 (e.g., 100000000006155622736)

Table 3.1: Main characteristics of G+ dataset

ID	# nodes	# edges	Start-Date	Duration
Ω_1	66,237,724	1,291,890,737	24-Aug-12	17
Ω_2	69,454,116	1,345,797,560	10-Sept-12	11
Ω_3	71,308,308	1,376,350,508	21-Sept-12	13
Ω_4	73,146,149	1,406,353,479	04-Oct-12	15
Ω_5	76,438,791	1,442,504,499	19-Oct-12	14
Ω_6	84,789,166	1,633,199,823	02-Nov-12	35
Ω_7	90,004,753	1,716,223,015	07-Dec-12	40
Ω_8	101,931,411	1,893,641,818	16-Jan-13	40
Ω_9	114,216,757	2,078,888,623	25-Feb-13	35
Ω_{10}	125,773,639	2,253,413,103	01-Apr-13	25
Ω_{11}	132,983,313	2,356,107,044	26-Apr-13	55
Ω_{12}	145,478,563	2,548,275,802	20-Jun-13	N/A

However, L^i is not a connected subgraph. Hence, we use BFS to extract the Largest Weakly Connected Component (LWCC) for each of the snapshots of $L^{i=1,..12}$. We label these extracted LWCCs as subgraphs H^i (for $i = 1, 2, ..., 12$).

In this paper, we consider subgraph H^i as the "reciprocal network" of G+ [3]. In the next sections, we will focus our analysis on the structural properties and evolution of H^i. To achieve this, we extract subgraphs H^i_j composed of the set of users that join the network at snapshot i and j represents this subgraph at specific snapshots ($j => i$).

Let ΔH^{i+1} denote the subgraph composed with the set of nodes that join subgraph H^i_j at snapshot $j = i + 1$. Then, we define the following relationship (see Table 3.2 and Fig. 3.2):

$$H^{i+1} = H^i \cup \Delta H^{i+1} \tag{3.1}$$

In the following sections, we use subgraphs ΔH^{i+1}, H^i_j and (3.1) to analyse the reciprocal network of G+. For clarity of notation, we sometimes drop the superscript i and subscript j from the above notations, unless we are referring to specific snapshots or subgraphs.

[3]It contains more than 90% of the nodes with at least one reciprocal edge in G+. Hence, our analysis of the dataset is eventually approximate.

Table 3.2: Summary of Notations

Ω^i	snapshots of the LWCC of G+
G^i	snapshots of the subgraphs composed with nodes with at least one mutual edge derivated from Ω^i
L^i	snapshots of the subgraphs derived from G^i by removing all the nodes that depart from G^i
H^i	snapshots of the LWCC of L^i
ΔH_j^{i+1}	subgraph composed with the set of nodes that join subgraph H^i at snapshot $j = i + 1$
i	subgraph index for i=1,...,12
j	snapshot index for j=1,...,12

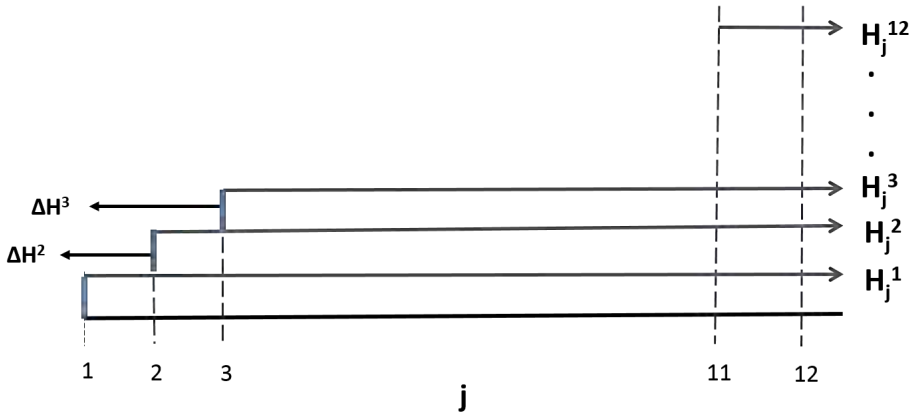


Figure 3.2: Notations Graph: illustration of the relationship between subgraphs ΔH^{i+1}, H_j^i and parameters i and j

3.4 Reciprocal Network Characteristics & Its Evolution

In this section, we present a comprehensive characterization of the connectivity and evolution of the reciprocal edges in G+, in order to shed an insightful light on the structural properties of the reciprocal network of G+. To achieve this, we proceed as follows: a) we provide a brief overview of the structural properties of the reciprocal network; b) we analyse the evolution of the density of the reciprocal network and c) we categorize the nodes joining the reciprocal network and their edges respectively.

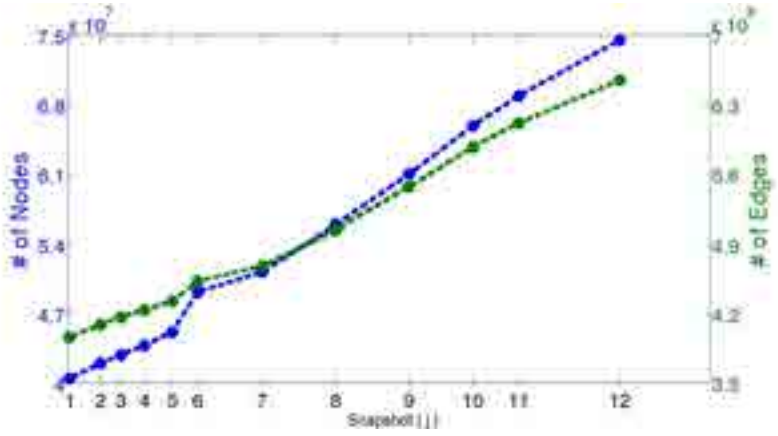


Figure 3.3: Growth in the number of nodes and edges in H

3.4.1 Overview of the Reciprocal Network

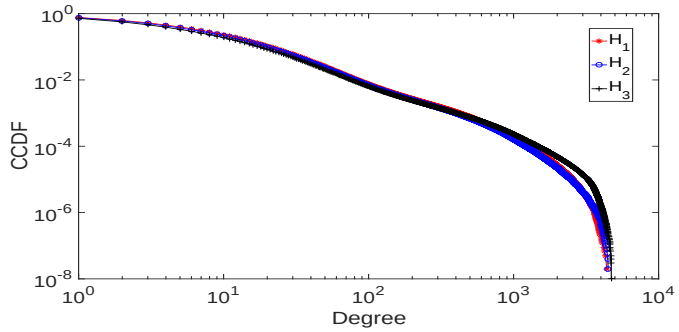
We start by providing a brief overview of some global structural properties of the reciprocal network of G+, more precisely, the growth of its number of nodes and edges, as well as, its degree distributions:

Nodes and Edges: Figure 3.3 plots the number of nodes (left axis) and edges (right axis) across time. We observe that the number of nodes and edges increase (almost) linearly as H^i evolves. The only exception is between H^i snapshots 5-6 ($19.Oct.12 - 02.Nov.12$), where we observe a significant increase in the number of nodes and edges. The time of this event correlates with the addition of a new G+ feature, on $31.Oct.12$, that allows users to share contents created and stored in Google Drive[61] directly into the G+ stream, as reported in[61]: "share the stuff you create and store in Google Drive, and people will be able to flip through presentations, open PDFs, play videos and more, directly in the G+ stream". Our dataset shows the impact of this event in G+: *it attracts more users to join G+ and many of these users might have already been using Google Drive in the past.*

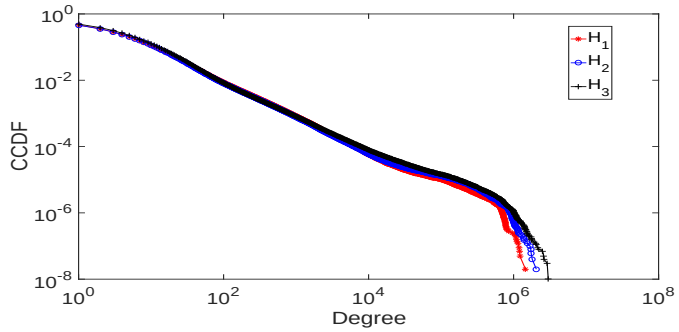
In-degree, Out-degree and Mutual Degree Distributions: Figure 3.4 shows the CCDF for mutual degree, in-degree and out-degree for nodes in subgraphs H^i. We can see that these curves have approximately the shape of a Power Law distribution. The CCDF of a Power Law distribution is given by $Cx^{-\alpha}$ and $x, \alpha, C > 0$. By using the tool in [**?**, **?**], we estimated the exponent α that best models our distributions. We obtained

$\alpha = 2.72$ for mutual degree, $\alpha = 2.41$ for out-degree and $\alpha = 2.03$ for in-degree. We observe that the mutual degree and out-degree distributions have similar x-axis range and the out-degree curve drops sharply around 5000. We conjecture this is because G+ maintains a policy that allows only some special users to add more than 5000 friends to their circles [55].

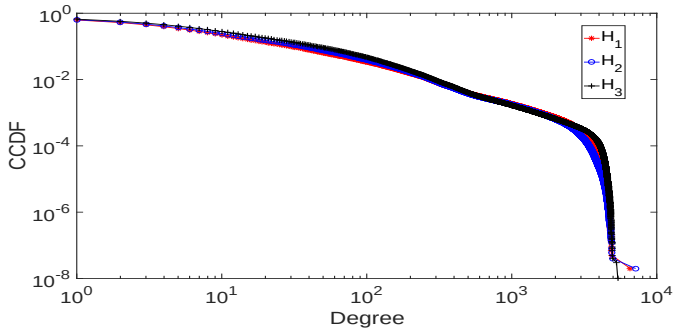
The observed power law trend in the distributions implies that a small fraction of users have disproportionately large number of connections, while most users have a small number of connections - *this is characteristics for many social networks*. We also observe that the shape of the distributions have initially evolved as the number of users with larger degree appeared.



(a) Mutual degree distribuition

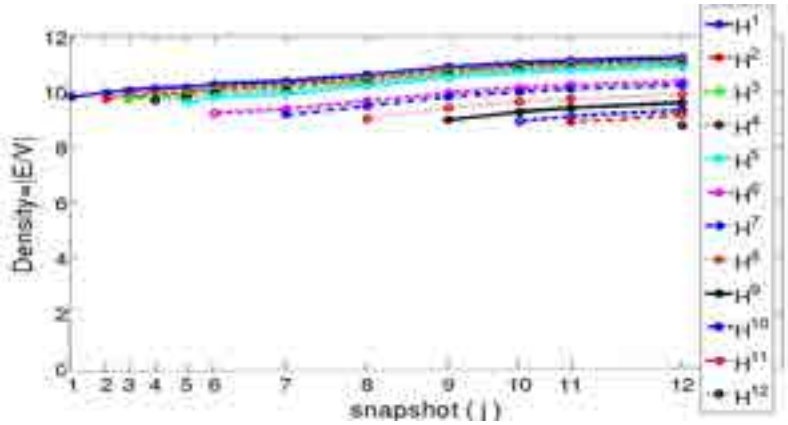


(b) In-degree distribution

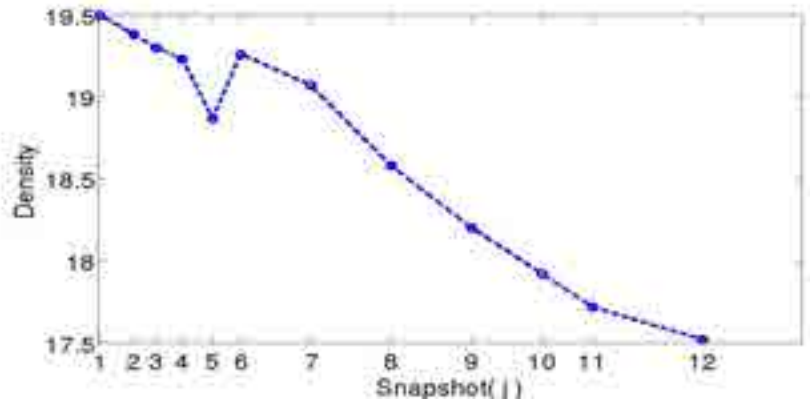


(c) Out-degree distribution

Figure 3.4: Log-log plot of a) mutual degree, b) in-degree and c) out-degree complementary cumulative distribution functions (CCDF) for several snapshots of the reciprocal network of Google+ (subgraphs H_i, i=1,2 and 3). All distributions show properties consistent with power-law networks.



(a) Density evolution - H^i



(b) Density evolution - Ω

Figure 3.5: Evolution of the Density for graphs Ω and H

3.4.2 Density Evolution & Nodes Categories

In this section, we analyze the evolution of the reciprocal network in terms of its density, and categorize the nodes joining the reciprocal network based on the (observed) time they joined the network. Next, we present our analysis:

Density: Figure 3.5(a) shows the evolution of the density of subgraph H^i, measured as the ratio of links-to-nodes[4]. We observe that as subgraph $H^{i=1,..12}$ evolves its density decreases. However, if we fix the number of nodes for each of the snapshots of H^i and analyse their evolution, we observe that the density is increasing (see Fig. 3.5(a)).

[4]We follow the terminology in [?] in order to compare with previous results

From these results, we conclude that the new users (ΔH^{i+1}) joining subgraphs H^i are responsible for the observed decrease in the density. Because these users initially create few connection when they join H^i (*cold start phenomenon*). However, the longer these users stay in the network, they discover more of their friends and consequently they increase their number of connections (edges). From the slopes of the graphs in Fig. 3.5(a), we observe that the new users are creating fewer links than the new users in the previous generation. Here, we define "previous generation" as the set of new users in the anterior snapshot, for example: the previous generation for new users in ΔH^3 are the users in ΔH^2.

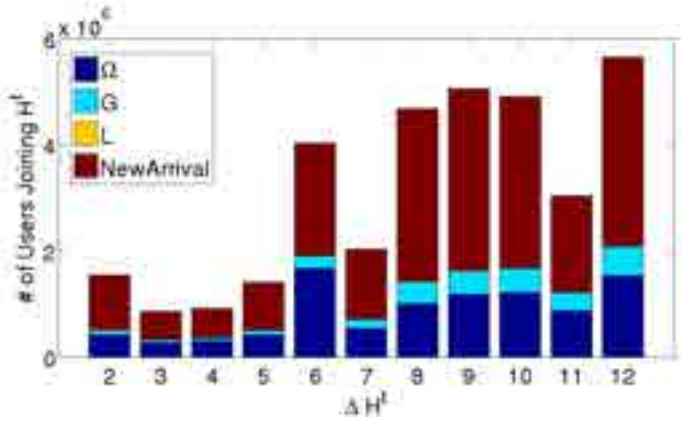
We also observe that the percentage of total users with at least one reciprocal edges in G+ decreases from 66.7% to 54.1% as the network evolves. Consequently, in our analysis, we also observe that the global reciprocity of G+ decreases (almost) linearly from 33.9% to 25.9%. From these results, we extract some important points: *a) the number of users creating at least one reciprocal edge is decreasing as the network evolves and b) the new users joining the reciprocal network are creating fewer edges than the users in the previous generation. Thus, the new users in G+ are becoming less social.*

Previous studies on social networks show that the social density for Facebook[**?**] and affiliation networks[**?**] increases over time. However, it fluctuates on Flickr[**?**] and is almost constant on email networks[**?**]. Differently, our dataset shows that the social density of G+ and of its reciprocal network (Fig. 3.5(a) and Fig. 3.5(b)) decrease as the network evolves. This is an interesting observation because it contradicts the *densification power law*, which states that real networks tend to densify as they grow[**?**].

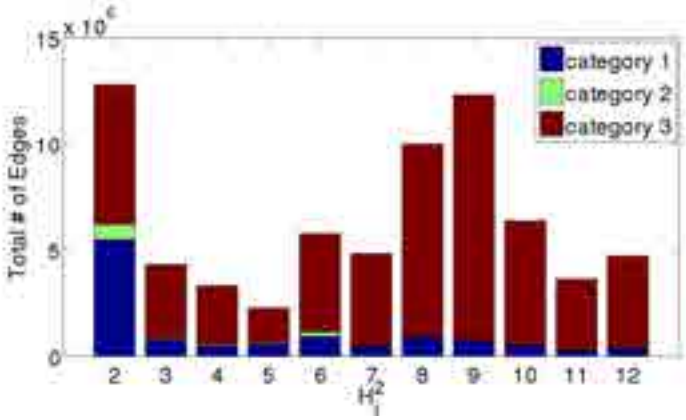
The authors in [20] analysed the evolution of the social density of G+ using a dataset collected in the first year of its existence ($06.Jun.11 - 11.Oct.11$). They reported that G+ social density fluctuates in an increase-decrease fashion in three phases, but it reaches a steady increase in the last phase[20]. *Differently, our results shows that the social density of G+ is decreasing as the network evolves from its second to third year of existence* – the only exception is between snapshots 5 to 6, due to the events discussed in Sect. 3.4.1.

Node Categories: we classify the nodes joining H into the following categories (for clarity of notations we drop the superscript i and subscript j):

- Ω: node "x" exists in subgraph Ω at snapshot $j-1$ and joins H at snapshot j



(a) Total number of nodes joining H^i per category



(b) Total number of new edges per category created in H_j^2 for each of j snapshots

Figure 3.6: Nodes and edges categories for subgraph H

- G: node "x" exists in subgraph G at snapshot $j-1$ and joins H at snapshot j
- L: node "x" exists in subgraph L at snapshot $j-1$ and joins H at snapshot j
- $NewArrival$: node "x" does not exist in the system at snapshot $j-1$ and joins both Ω and H at snapshot j

Figure 3.6(a) shows the distribution of the nodes joining H by categories. We observe that on average 63% of the nodes joining the subgraph H are new users in the system, 29% comes from the subgraph Ω and the remaining percentage comes from

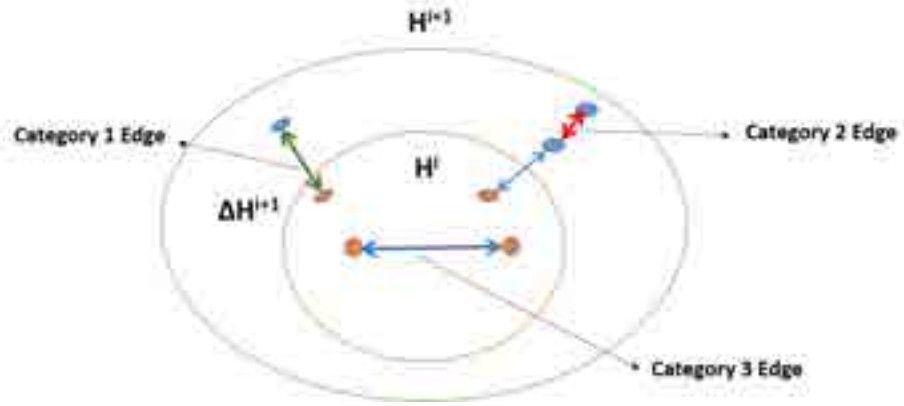


Figure 3.7: Categories of the edges in subgraph H^i (for i=1,...,12)

either subgraphs G or L. From these results we infer the following: *a) the majority of users that are joining the reciprocal network of G+ are new users in the system; b) if a user doesn't create a reciprocal edge when he/she joins G+, it is very unlikely that he/she will ever reciprocate a link in the network.*

3.4.3 Edge Categories & Its Evolution

In order to understand the connectivity between the nodes in the reciprocal network, we analyse the evolution of the reciprocal edges in H^i. To achieve this, we restrict our analysis[5] to the subgraphs H^1 and H^2. Firstly, we present our edges categories. Secondly, we analyse the evolution of the degree distribution for each edge category:
Edges Categories: we classify the edges created by nodes joining H^i into the following three categories (see Fig. 3.7 for an illustration):

- Category 1: $e(u, v)$ such that $u \in \Delta H^{i+1}$ and $v \in H^i$
- Category 2: $e(u, v)$ such that $u \in \Delta H^{i+1}$ and $v \in \Delta H^{i+1}$ and $\exists v^* \in H^i : e^*(u, v^*)$
- Category 3: $e(u, v)$ such that $u \in H^i$ and $v \in H^i$

Figure 3.6(b) shows the distribution of the edges based on the defined categories. We observe that most of the new edges seen across all snapshots of H_j^2 are due to category 3

[5]Similar results are obtained using the other subgraphs ($H^{i=3,\ldots,12}$)

edges. Furthermore, by looking at the last snapshot of H^i (for $i = 12$), we observe that 69% of the edges in H_{12}^{12} are between nodes in H^1 only. This result shows that although the density decreases as subgraph H^i evolves, the connectivity of a subset of its nodes is increasing (*densification*) and their connectivity accounts for a huge percentage of the total edges in the system.

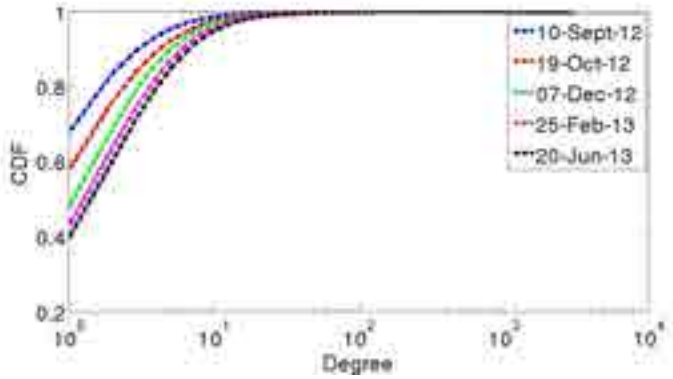
Degree Distribution: Figure 4.13 shows the degree distribution for all categories of edges and how they evolve across time. Figure 3.8(a) shows the CDF of the degree distribution for category 1 edges. From this figure, we observe that when new nodes (ΔH^2) join H_2^1, initially they create few connections, but the longer they stay in the system the number of connections to nodes already in the system increases significantly (as stated in Sect. 3.4.2). Furthermore, from our dataset, we observe that 72% of the nodes in ΔH^2 have only connections (edges) to nodes already in the system (H_2^1).

Figure 3.8(b) shows the CDF for the degree distribution of category 2 edges. From the results of Fig. 3.8(a) and Fig. 3.8(b), we infer that when new nodes (ΔH^2) join H_2^1, they create more connections with the nodes already in the system. Figure 3.8(c) shows the degree distribution for edges of category 3. We observe that the shape of the degree distribution is decreasing which implies that the network is become more dense (*densification*), as discussed above.

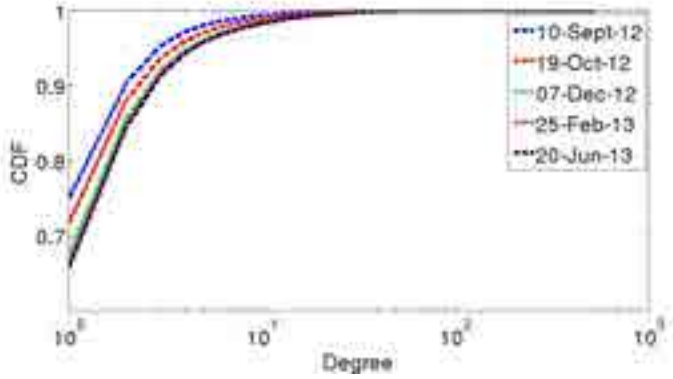
In summary, our analysis on the categories of nodes and edges yields the following key findings: *a) the majority of users that joins the reciprocal network of G+ are new users in the network and they tend to create reciprocal connections mostly to users who already have reciprocal connections to others; b) if a user does not create a reciprocal edge when he/she joins G+, there is a lower chance that he/she will create one later.*

3.5 Implication of our Results for G+ & Conclusion

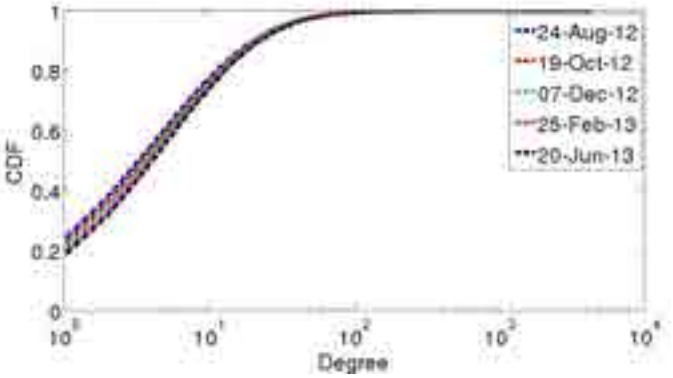
In this paper, we present the first study on the properties and evolution of a "reciprocal network", using a massive G+ dataset. Analyzing the connectivity of reciprocal edges is important because they are the most stable type of connections in directed network and they represent the strongest ties between nodes: users with large number of mutual edges are less likely to depart from the network and they may form the most relevant



(a) Category 1



(b) Category 2



(c) Category 3

Figure 3.8: Degree distribution per edge category

community structure[6](the intimacy community [19]) in directed OSN networks. Our analysis show that the reciprocal network of G+ reveals some important patterns of the user's behavior, for example: new users joining G+ are becoming less social as the network involves and they tend to create reciprocal connections mostly to users who already have reciprocal connections to others. Understanding these behaviors is important because they expose insightful information about how the social network is being adopted.

The findings here also provide hints that can help explain why G+ has so far failed to compete with Twitter and Facebook, as recently reported [62]. Firstly, we observe that although the numbers of nodes and edges increase as G+ evolves, the density of the network is decreasing. This result supports the claim that some users joined G+ because they need to access some of Google products but they weren't interested in creating connections in the network, in contrast to users in Twitter. Secondly, we observe a decrease in the reciprocity of G+ because the percentage of users with at least a reciprocal edge decrease as the network evolved. Furthermore, the users that joined the reciprocal network later always create fewer connections than the users who joined earlier. From this result, we infer that many users do not use G+ to connect and chat with friends, in contrast to users in Facebook[7]. Therefore, in its second year of existence, the G+ social network was already showing "signs" that it was failing to compete with others online social network, such as Twitter and Facebook. Many of the studies in the literature about G+ [55], [20, 21], [**?**] were done using dataset mostly collected in the first year of G+ existence. Thus, they either did not observe or failed to see these signs.

Our work is only a first step towards exploring the connectivity of reciprocal edges in social and other complex networks – reciprocal networks. There are several interesting directions for future work that we will pursue to uncover the properties of reciprocal networks so as to further understand the structural properties of directed graphs.

[6]We will analyse the community structure in a reciprocal network as future work
[7]The authors in [21] stated similar conclusion

Chapter 4

Uncovering the Nucleus of Complex Networks

4.1 Introduction

Networks are often abstractly modelled as a graph where vertices represent entities and edges capture the relations (e.g., connections) or interactions between them. In the context of (online) social networks, community identification has received a lot of attention. A community is often considered to be a subset of vertices that are densely connected internally but sparsely connected to the rest of the network [25, 26, 27, 28, 29]. The majority of studies on identifying communities structures in social networks have relied on clustering techniques, namely, by partitioning the underlying network/social graph into *disjoint* (sometimes *overlapping*) communities. For example, Newman proposes a measure of betweenness – modularity [27, 28] – for identifying disjoint communities in a social network. Andersen et al [29] design a local graph partitioning algorithm to indentify community structures. This algorithm is based on personalized PageRank vectors. Ahn et al [63] introduce a novel perspective for discovering hierarchical community structures by categorizing links only. To obtain an optimal partition and to find communities at multiple levels, an information-theoretic framework is proposed by the authors in [64, 65]. Several studies use link and content information for uncovering meaningful communities in networks [66, 67].

Although existing studies of community structure have been very successful, most

have not considered the existence of "core structure" in many networks. Intuitively, one expects that many social networks possess some sort of "core" as part of their meso-scale structure, which holds various parts of the network (or constituent "communities") together. We believe that it is just as important to uncover and extract the "core" structure – referred to as the "nucleus" – of a social network as identify its community structure [68, 69]: unlike "ordinary" constituent communities, the "core" structure plays a crucial role in the formation and evolution of a social network, to which other (constituent) "communities" are attached. Chung and Lu [70] show that power-law random graphs almost surely contain a core "subgraph" when the exponent β in the power-law degree distribution is such that $\beta \in (2, 3)$. This theoretical result suggests that many real-world social networks likely posess some sort of cohesive core structure.

One of the most popular notion of network core is given by the *k-shell decomposition* method [1]. This classical graph decomposition technique decomposes a network into hierarchically ordered layers from the periphery to the core. This method has also be extended to weighted graphs [71, 72] and dynamic networks [73]. The k-shell decomposition method has often been used as a visualization tool for studying the core structure of massive complex networks such as the Internet [1]. In addition, it has been used to identify influential spreaders in a network [74, 75].

When applying the standard k-shell decomposition to uncover the core of several example social networks (see § **??**), we find that the resulting "innermost" structure is unlikely to represent the "core" of these networks. For example, this "innermost" structure may contain the maximum clique of a network but which lies rather at its periphery, or it is simply a single vertex in a dense graph. This appears to the effect of the (iterative) degree-based pruning process of k-shell decomposition, where despite at some point we reach the vicinity of the core, the k-shell decomposition continues further, which then destroys the "core" structure of the network (see § 4.3 for more illustration). This raises the following important question: *When should we stop the k-shell decomposition pruning process in order to preserve the core graph G_C of a network?*

In an attempt to address this question, we develop an effective procedure to uncover the *nucleus* structure of a social network by building upon and generalizing ideas from the existing k-shell decomposition [1] approach, as follows. Firstly, we propose a new

metric, the *dependence value*, that measures the location importance of a node in a network. Intuitively, the dependence of node v captures the number of nodes recursively dependent of v that have been removed in earlier steps of the k-shell decomposition method. Secondly, we derive a new measure called *nucleon-index* (NI) that captures the extend to which a subgraph is a densely intra-connected and topological central core. This index can be used with a wide variety of functions to transition between core and peripheral nodes (e.g., dependence value, closeness [76] and betweenness [76] centralities, etc). Using these metrics, we therefore modify the standard k-shell decomposition method to stop the process earlier, in order to extract a meaningful "core" for social networks (see § 4.4). For a Facebook [77, 78] friendship network composed of 63,731 nodes and 817,035 edges, this process yields a dense "core" subgraph G_C with approximately 285 nodes and 9,616 edges. Given a dense core subgraph G_C, we investigate the importance of this substructure for the network by analysing the following metrics (see § 4.5): i) the distance between a node v to the core subgraph G_C; ii) the ratio of the distance between nodes u and v to their respective distance to G_C and iii) lastly, the impact of removing G_C in the structure of the network G ($G_C \subset G$).

We extend our definition of nucleon-index for massive complex networks and discuss implications in § 4.6 and § 4.7. Section 4.8 concludes the chapter. We summarize the major contributions of our paper as follows:

- We show that applying the conventional k-shell decomposition method to some complex networks produces inner-most structures that are not the "core" of these networks.

- We propose two *new* metrics: i) the *dependence value*, that measures the location importance of a node in the network; ii) the *nucleon-index* (NI) that captures the extend to which a subgraph is a densely intra-connected and topological central core . Using these metrics, we therefore modify the standard k-shell decomposition method to stop the process earlier, in order to extract a meaningful "core" for social networks.

- We apply our approach to uncover the core structure in example communication, computer, infrastructure, human-contact, collaboration, interaction, location-based and online social networks.

- We extend our definition of *nucleon-index* to extract the core structure of massive networks (hundreds million nodes and billion edges). Furthermore, we show the effectiveness of our approach by applying it to uncover the core structure of the reciprocal network of a massive Google+ dataset (with more than 40 million nodes and close to 200 million edges).

4.2 Datasets

This section presents a summary of the datasets that we use for our analysis:
Autonomous systems graph: This dataset are undirected graphs of the AS peering information inferred from Oregon route-views and CAIDA projects: *Oregon-1* [79], *Route views* [80, 81], *CAIDA* [82, 80] and *Internet* [83, 84]. Table 4.1 summarizes the main features.
Infrastructure systems graphs: This dataset is a collection of 3 undirected graphs of infrastructure systems [85, 86, 87, 88, 89]:

- *Euro-road*: European international E-road network – a graph contains an undirected edge (i, j), if city i is connected by E-road to city j.
- *US airports*: it is an undirected network of flights between US airports in 2010. Each edges (i, j) represents a connection from one airport to another, in 2010.
- *OpenFlights*: it is an undirected network of flights between airports of the world. An edge (i, j) represents a connection from one airport to another.

Social networks graphs: This dataset is a collection of 9 undirected graphs of communication, collaboration, interaction, human contact, location-based and online social networks [79, 77, 90, 78, 91, 92, 93, 94, 95, 96, 97, 98, 99](see Table 4.1 for a summary of the main features):

- *ca-AstroPh, ca-HepPh, ca-CondMat*: collaboration networks between authors for papers submitted to Astro Physics, High Energy Physics (Phenomenology category) and Condense Matter Physics – a graph contains an undirected edge (i, j), if author i co-authored a paper with author j.

- *arenas-jazz*: collaboration network between jazz musicians – the graph contains an undirected edge (i, j), if two musicians have played together in a band.
- *email-Enron*: email communication network – the graph contains an undirected edge (i, j), if address i sent at least one email to address j.
- *arenas-pgp*: interaction network of users of the Pretty Good Privacy (PGP) algorithm.
- *train bombing*: human contact network between suspected terrorists involved in the March 11, 2004 Madrid train bombing – the graph contains an undirected edge (i, j), if two terrorists were in contact.
- *infectious*: human contact network of people during the exhibition "Infectious: Stay Away" (2009) – the graph contains an undirected edge (i, j), if two exhibition visitors had face-to-face contacts that were active for a least 20 seconds.
- *dnc-corecipient*: online contact network for people having received the same email in the 2016 Democratic National Committee email leak – the graph contains an undirected edge (i, j), if two persons received the same email.
- *Facebook*: an undirected subgraph of the friendship network for the users in Facebook.
- *loc-brightkite*: an undirected graph for the friendship network for the users from loc-brightkite location-based online social network.

"Massive" social network graphs: this dataset[1] is a collection of three massive directed graphs of the social links of the users in $G+$, collected between August, 2012 and June, 2013. Table 4.2 summarizes the main features of this dataset, where each snapshot represents a complete graph of the social relations among all users in $G+$. In these dataset, a node represents an user and if user i x follows user j this relationship can be represented as a directed social edge (i, j); if user j also has a directed social edge (j, i), the relationship j, j is called reciprocal.

[1]We obtained our dataset from an earlier study on G+ [100]

Table 4.1: Main characteristics of the social networks and AS graphs: d - node degree; % LCC - percentage size of the largest connected component of the original network

ID	# nodes	# edges	max(d)	% LCC
train bombing	64	243	29	1.00
arenas-jazz	198	2,742	100	1.00
infectious	410	17,298	294	1.00
dnc-corecipient	906	20,858	368	0.94
Euro-road	1,174	1,417	10	0.89
US airports	1,574	28,236	596	1.00
OpenFlights	2,939	30,501	473	0.99
Route views	6,474	13,895	1,549	1.00
arenas-pgp	10,680	24,316	205	1.00
Oregon-1	11,174	23,409	2,389	1.00
ca-HepPh	12,008	118,521	491	0.93
ca-AstroPh	18,722	198,110	504	0.95
ca-CondMat	23,133	93,497	280	0.92
CAIDA	26,475	53,381	2,628	1.00
Internet	34,761	171,403	5,305	1.00
email-Enron	36,692	183,831	1,383	0.92
loc-brightkite	58,228	214,078	1,134	0.97
Facebook	63,731	817,035	1,098	0.99

Table 4.2: Main characteristics of Google+ snapshots: (start-date, duration) – Γ_1: (24-08-12, 17 days), Γ_2: (10-09-12, 11 days) and Γ_3: (20-06-13, N/A)

ID	# nodes	# edges	max(in)	max(out)
Γ_1	66,237,724	1,291,890,737	2,289,874	9,981
Γ_2	69,454,116	1,345,797,560	3,463,060	9,872
Γ_3	145,478,563	2,548,275,802	5,089,789	10,840

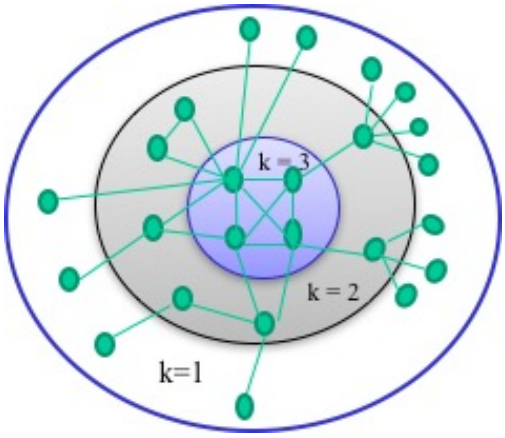


Figure 4.1: A schematic representation of a network under k-shell decomposition: the network can be viewed as the union of shell 1 up to $k_{max} = 3$ (network core).

4.3 k-shell network core

K-shell decomposition [1] is one of the most popular and scalable method to investigate and visualize the core-periphery structure in complex networks. This method assigns to each node an integer representing its coreness location according to successive layers or shells in the network. It works as follows: a) first, remove all nodes in the network with degree 1 (and their respective edges) – these nodes are assigned to the 1-shell; b) more generally, at step $k = 2, \ldots$, remove all nodes in the remaining network with degree k or less (and their respective edges) – these nodes are assigned to the k-shell; and c) the process stops when all nodes are removed at the last step. Small values of k define the periphery of the network and the *innermost network core* corresponds to the highest shell index (k_{max}) – see Fig. 4.1. (Note that this is distinct from *k-core decomposition*[2] defined in the literature [101, 102]).

In the k-shell decomposition process, at each step k, the remaining subgraph is referred to as "k-core" (C_k). The k-core subgraph is the union of all shells with indices larger or equal to k or it is the maximal induced subgraph $C_k \subseteq G$ such that if $v \in C_k$, then node v must have at least $k+1$ neighbors that belong to C_{k-1} and $deg^k(v) > 0$ (we use $deg(v)$ to denote the degree of v in the network and $deg^k(v)$ to denote the degree of v in C_k). Similarly, k-shell (S_k) can be defined as the subgraph induced by the set of

[2]Which simply removes all nodes with degree less than k in a graph.

nodes with $d^{k-1}(v) \leq k$ and if $v \in S_k \rightarrow deg^k(v) = 0$.

Clearly, for a node to belong to the k-core (thus $shell(v) \geq k$), it must have at least degree k, i.e., $deg(v) \geq k$. However, $deg(v) \geq k$ is not sufficient to guarantee it to belong to the k-core. For example, a node v with only neighbors of degree 1 (i.e., v is the root of a star structure) belongs to the 2-shell, i.e., $shell(v) = 2$, no matter how high its degree is. On the other hand, it is easy to see that if a node v is part of a clique of k nodes, then $shell(v) \geq k$. However, a node v does not need to be part of a k-clique to have $shell(v) \geq k$. Consider a tree T of n nodes (the sparsest graph with n nodes). We can in fact provide a complete characterization of nodes in T to have $shell(v) \geq k$ in a recursive manner: for v to have $shell(v) \geq k$, it must have at least k-neighbors u's with $shell(u) \geq k - 1$ – this characterization also applies to a general graph. We see that in the case of a tree, nodes with higher k-shell indices must lie more at the "core" (i.e., the increasingly "denser" part) of the tree. For a general graph, however, a node with a high k-shell index may not lie at the "core" of the graph: it can be part of a large clique that is "isolated" on a periphery of a massive graph. In such a case, the large clique will break off from the "core" of the network (e.g., as represented by the largest connected component remaining in the k-core) in the early stage of the k-shell decomposition process.

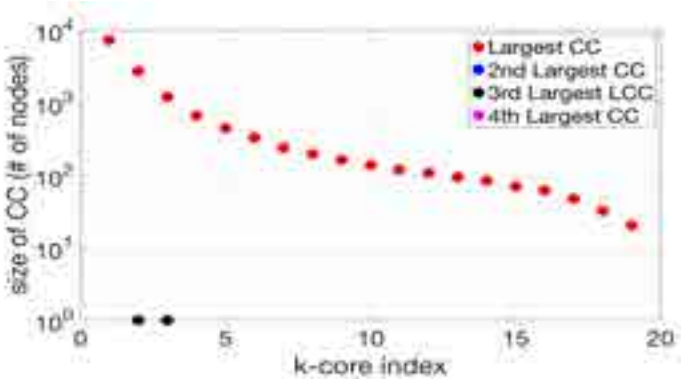
This method has been successfully used as a visualization tool for studying and uncovering the core structure of networks such as the Internet AS graph [1]. We apply it to the *Oregon-1* AS dataset. Fig. 4.2(a) shows the size of the largest as well as those of the 2nd, 3rd and 4th largest connected components in the k-core graph. We observe that the largest connected component decreases smoothly as k varies from 1 to 20. At $k_{max} = 20$, we are left with a very dense core subgraph composed of 20 nodes and 164 edges – the network nucleus. This result shows that for the AS graph, nodes with the highest k-shell indices indeed lie at the "core" (i.e., the increasingly "denser" part) of the graph. However, our experiments reveal that applying the k-shell decomposition for other types of graphs, especially social graphs, may not yield the same results. There are two possible reasons:

First, for some graphs the k_{max}-shell seems to contain some "residual" portions of the nucleus of a graph or simply a singleton node. For example, Fig. 4.2(b) shows the k-core graph for the 4 largest connected components in the *ca-AstroPh* dataset. We

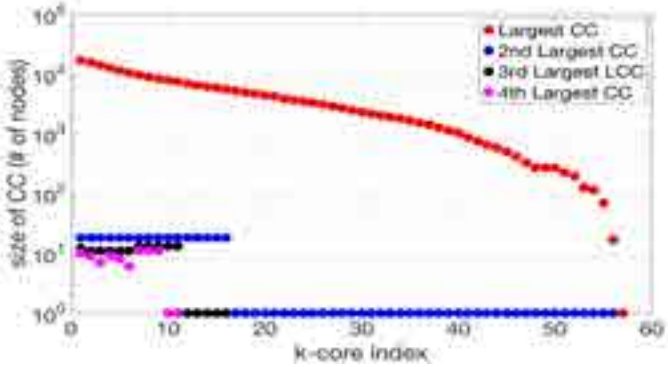
see that at k_{max}=57, we are left with just a single node in the k-core graph, which is unlikely to be the complete inner-core of the graph. Second, in other graphs the k_{max}-shell does not appear to lie at the "core" of the graph: it could be part of a large community structure (e.g. a maximum clique) that is "isolated" on a periphery of a graph. To illustrate this, we apply the k-shell decomposition method to a *Google+* reciprocal network[3] obtained from a previous study [14, 11] - it consists of more than 40 million nodes and $\approx$ 400 million edges. Figure 4.2(c) shows the size of the largest as well as those of the 2nd, 3rd and 4th largest connected components in the k-core, as k varies from 1 to 308. We note that at step $k = 121$, a small subgraph containing the maximum clique (of size 290) breaks off from the largest connected component which desolves after $k = 253$, whereas this subgraph containing the maximum clique persists after $k = 252$ and becomes the largest component; at $k_{max} = 308$, we are left with this maximum clique plus 10 additional nodes that are connected to the maximum clique. Closer inspection of the nodes in the maximum clique reveals that its users belong to a single institution in Taiwan, forming a close-knit community where each user follows everyone else – which is unlikely to be the network core of Google+.

From these results, we see that directly applying the standard k-shell decomposition to some graphs (especially, social networks) produces an "innermost" structure that does not represent "core" of these networks. This is due to the fact that at a certain k-index, we reach the vicinity of the core; but going far beyond this index would destroy the core structure of the network.

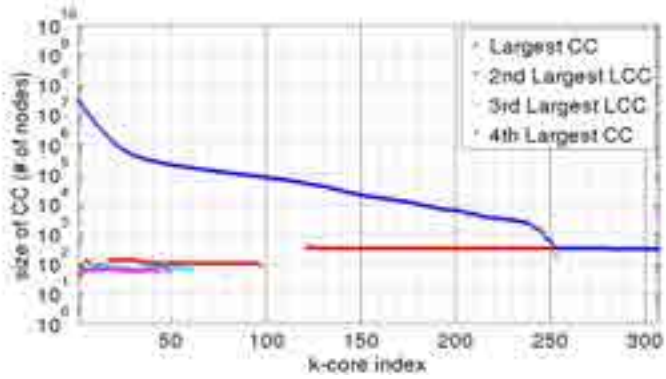
[3]A network composed with only bi-drectional edges, extracted from a directed social graph. A reciprocal network can be viewed as the stable "skeleton" network of a directed social network that holds it together and encodes its main topological characteristics [14]. For more on the reciprocal network of Google+ the reader is referred to [14, 11].



(a) Oregon-1



(b) ca-AstroPh



(c) Google+

Figure 4.2: The size of the largest as well as those of the 2nd, 3rd and 4th largest connected components in the k-core subgraphs

4.4 Node Depencence Values and Network Core

In order to extract a meaningful "core" for a general graph $G = (V, E)$ (e.g., social networks), we therefore modify the standard k-shell decomposition method to stop the process earlier. To achieve this, we propose a new metric that provides important information about the structural function of each node in the graph (we label it as "dependence" value) at each k-step. Then, we present a new measure called *nucleon-index* (NI) that captures the extend to which a subgraph is a densely intra-connected and topological central core – it can be used with a wide variety of functions to transition between core and peripheral nodes (e.g., dependence value, closeness and betweenness centralities, etc).

4.4.1 Node Depencence Values

The *dependence* value of node v at step k is defined as follows: for $v \in V$, $dep^0(v, \beta) = 0$ and for $k = 1, \ldots, c(v)$,

$$dep^k(v, \beta) := dep^{k-1}(v, \beta) + \delta^k(v) + \beta \times \Sigma_{u \in N^k(v)}[dep^{k-1}(u, \beta)] \tag{4.1}$$

where β is a control parameter, $0 \leq \beta \leq 1$; $N^k(v)$ is the set of neighbors of node v that are removed at step k, and $\delta^k(v) = |N^k(v)|$. The dependency of node v is recursively defined by measuring the number of nodes u (the h-hop neighbors of v, $h = 1, ..., k$) that are removed in earlier steps up to $k = c(v)$ –the *coreness* of node v (and for $k \geq c(v)$, by convention, we define $dep^k(v, \beta) = dep^{c(v)}(v, \beta)$).

Intuitively, $dep^k(v, \beta)$ captures the number of nodes recursively dependent on v that have been removed in earlier steps up to k. With $\beta = 0$, we note that $dep^k(v, \beta)$ captures the number of v's neighbors removed at each step up to k, and for $k \geq c(v)$, $dep^k(v, \beta) = \sum_k \delta^k(v) = deg(v)$, the degree of node v. With $\beta > 0$, $dep^k(v, \beta)$ captures not simply the dependence of its neighbors, but that of its neighbors' neighbors, and so forth. However, the number of nodes u removed at each step up to k does not influence the dependence value of the node v uniformly. Their contribution is weighted by the parameter β in eq.(4.1). The parameter β quantifies the contribution of node u to the total dependence value of node v. More precisely, at the kth-step, we multiply the number of h-step removed neighbors of v by β^{h-1} (see the proof in the appendix). Thus,

Table 4.3: $Arenas - jazz$: peak nucleon-indices (NI) and their respective k_C-indices (set SK) and β values

β	max(NI)	k_C
0.0	0.011019	26
0.1	0.006561	25
0.2	0.006125	24
0.3	0.006841	24
0.4	0.007256	24
0.5	0.007500	24
0.6	0.007818	25
0.7	0.008545	25
0.8	0.009222	25
0.9	0.009849	25
1.0	0.010433	25

the further a node u is to node v, the less it will contribute to the total dependence value of node v. Hence, a node v having more nodes u with high dependence values in its vicinity will also have a high dependence value, creating the *dependency propagation* effect. Therefore, we posit that the network core should contain only nodes with very high dependence because the $dep^k(v, \beta)$ values of any $v \in V$ grows as k increases (more nodes are removed as we move from the periphery of the graph to its core). In the next section, we use the dependence value of node v as a measure of its coreness.

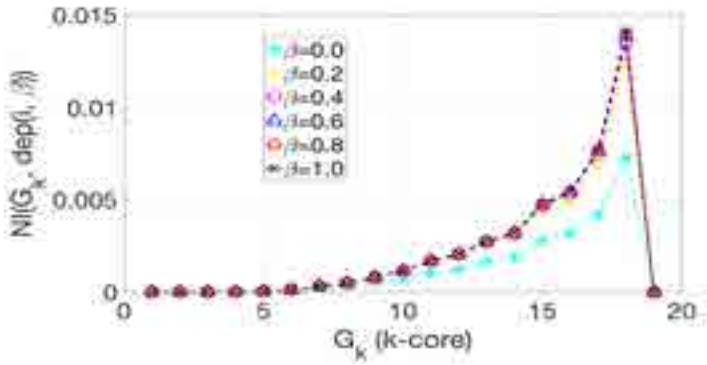
4.4.2 Nucleon Index and Network Nucleus

To derive a meaningful "core" structure in social networks, we postulate that the *nucleus* of a network $G(V, E)$ is an induced subgraph G_C having the following properties:

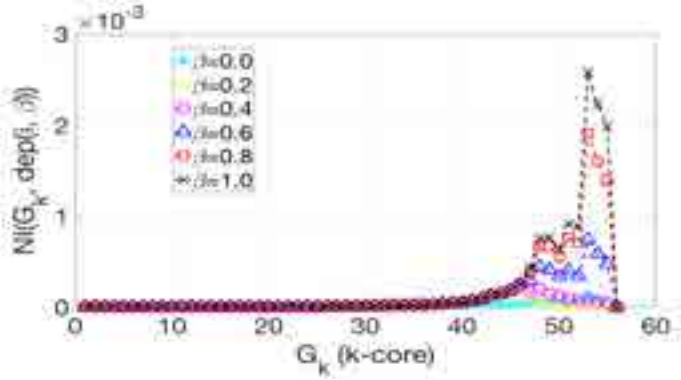
1. Subgraph $G_C(V_C, E_C)$ is *connected* and composed of a collection of nodes in G with *dense* aggregate centralities by some measure.

2. The set V_C is fundamental for the *structural properties* of the network, e.g., in terms of connecting nodes via short paths through the network.

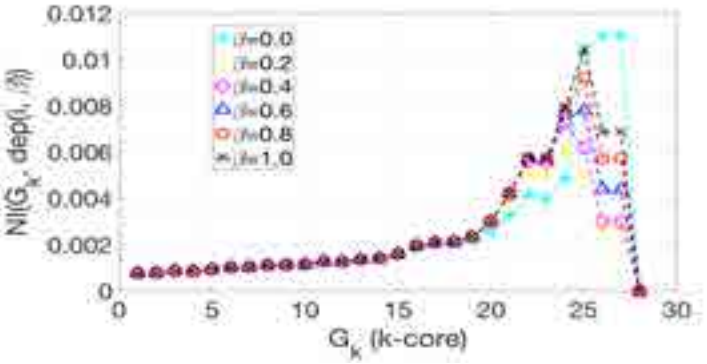
3. G_C is the minimal subgraph with these properties.



(a) Oregon-1

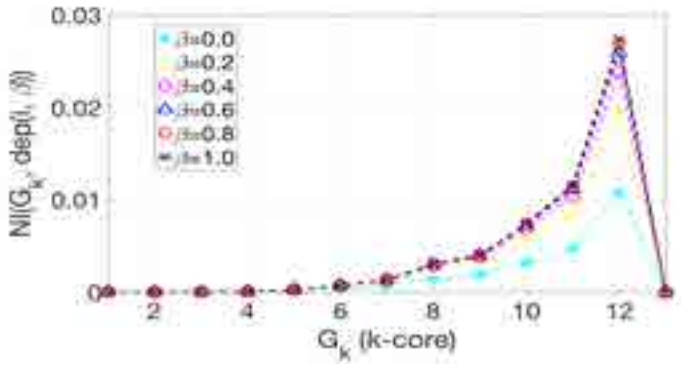


(b) ca-AstroPh

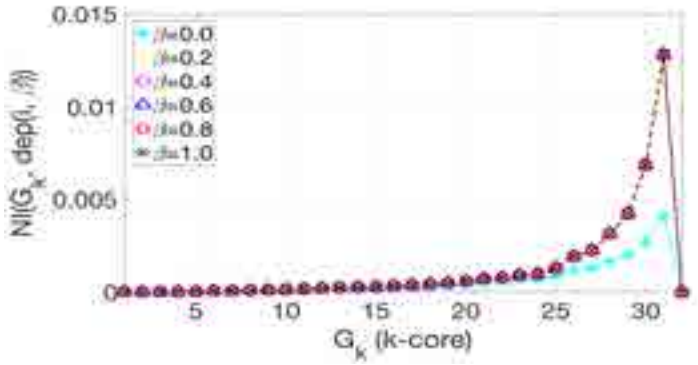


(c) arenas-jazz

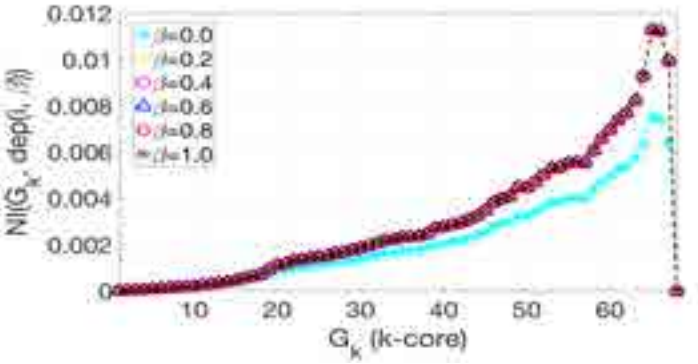
Figure 4.3: Variation of the nucleon-index per k-core index for several β parameters in the dependence computation: Oregon-1, ca-AstroPh and arenas-jazz



(a) Route views



(b) OpenFlights



(c) US airports

Figure 4.4: Variation of the nucleon-index per k-core index for several β parameters in the dependence computation: Route views, OpenFlights and US airports

To find a subgraph G_C with the above properties, we consider an appropriately defined "decomposition" process (e.g., the k-shell decomposition) which yields a (filtration) sequence of (sub)graphs $\{G_k\}$'s of G: $G_0 := G \supset G_1 \supset \cdots \supset G_K = \emptyset$. Given a node centrality measure $\theta(i)$, $i \in V$, we define the *nucleon-index* (NI) to capture the extent to which a subgraph constitutes a "densely connected", topological central core in this sequence:

$$NI(G_k, \theta(i)) := \frac{V_k}{V_{k-1}} \times \frac{E_k}{V_k \times (V_k - 1)} \times \{\frac{1}{V_k} \times \sum_{i \in G_k} \theta(i)\} \tag{4.2}$$

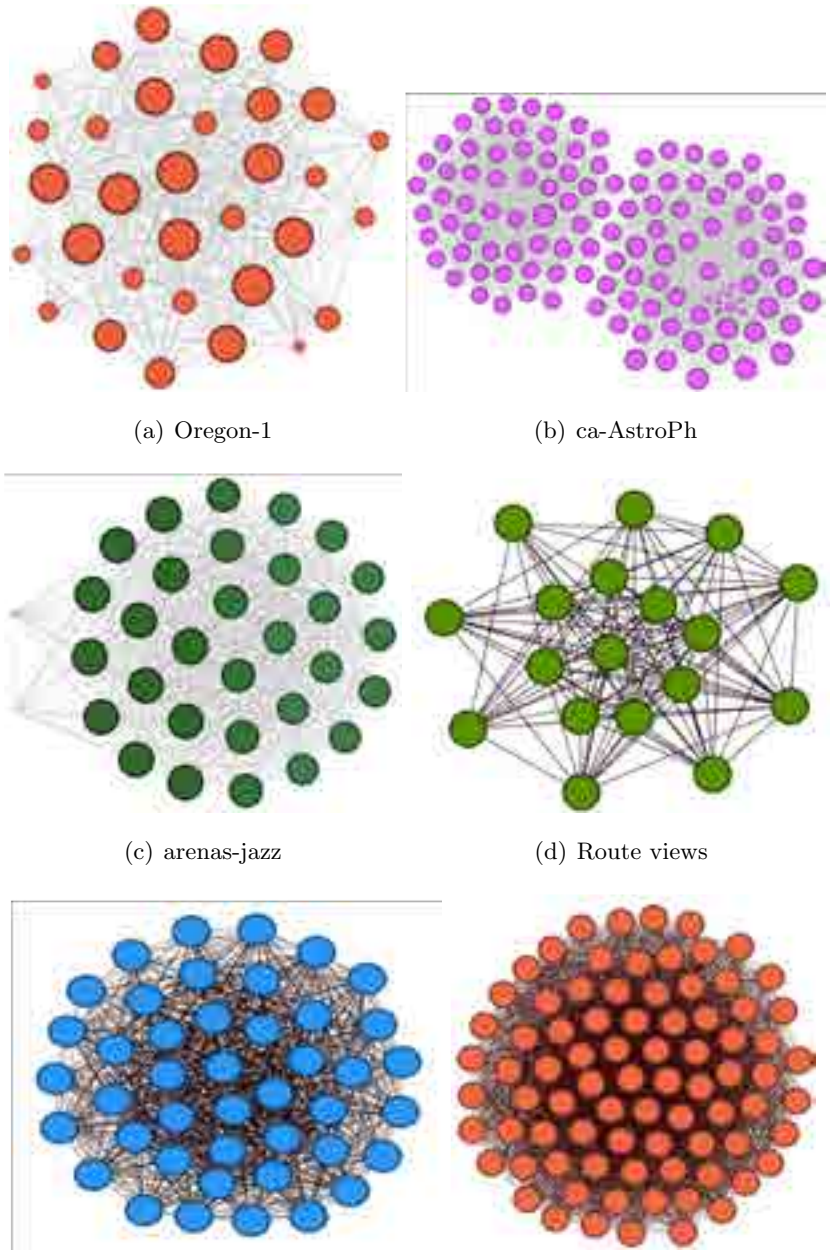
where by abuse of notation, we use E_k to denote the number of edges between nodes in G_k and V_k the number of nodes in G_k (and $|V_K| = 0$). The second term in eq.(4.2) measure the density of G_k and the last term the average centrality of G_k. Ideally, if G_k is a "dense core" of G, the product of these two terms should be large. The first term controls the rate of changes in size from G_k to G_{k+1}: intuitively, if G_k is the "nucleus" of G, going from G_{k-1} to G_k should not drastically change its size; but going from G_k to G_{k+1} amounts to breaking G_k apart, yielding a collection of small connected components. In other words, V_{k+1} would fall off quickly, as G_{k+1} is a small connected subgraph or an empty graph. Hence, G_k with the largest NI represents the *nucleus* of G (as produced by the decomposition process).

Considering the node dependence value as a centrality measure, we define $\theta(i)$ as follows:

$$\theta(i) := \frac{dep^{c(i)}(i, \beta)}{\sum_{j \in G} dep^{c(j)}(j, \beta)}. \tag{4.3}$$

Using $\theta(i)$ defined above and applying the nucleon-index to the k-shell decomposition procedure, we develop the following *stop rule* for core extraction.

Stopping rule for core extraction: For any graph G with a dense core structure, we should stop the k-shell decomposition method at the induced subgraph of the k_C-core with maximal nucleon-index. Thus, we seek a k_C-index that maximizes the nucleon-index (NI).



(a) Oregon-1 (b) ca-AstroPh

(c) arenas-jazz (d) Route views

(e) OpenFlights (f) US airports

Figure 4.5: Visualization of the core subgraphs of example networks: the size of a node is proportional to its degree. Oregon-1 (32 nodes, 362 edges); ca-AstroPh (126 nodes, 3,378 edges); arenas-jazz (24 nodes, 144 edges); Route views (18 nodes, 127 edges); OpenFlights (42 nodes, 742 edges); US airports (81 nodes, 3,073 edges).

Table 4.4: maximum k-shell index (k_{max}); β parameter; k-index to stop the shells pruning process (k_C); number of nodes and edges in the core subgraph N(G_C) and E(G_C)

Network	k_{max}	β	k_C	$N(G_C)$	$E(G_C)$
train bombing	11	0.5	7	24	144
arenas-jazz	29	0.6	25	32	466
infectious	20	0.5	17	29	303
dnc-corecipient	75	0.5	67	87	3,118
Euro-road	5	0.5	3	14	16
US airports	69	0.5	65	81	3,073
OpenFlights	33	0.5	31	42	742
Route views	14	0.5	12	18	127
arenas-pgp	33	0.5	31	38	658
Oregon-1	20	0.5	18	32	362
ca-HepPh	238	0.5	99	239	28,441
ca-AstroPh	57	0.6	53	126	3,378
ca-CondMat	51	0.5	37	37	382
CAIDA	26	0.5	23	50	765
Internet	67	0.5	64	115	4,578
email-Enron	51	0.5	48	150	4,395
loc-brightkite	58	0.5	56	66	1,893
Facebook	64	0.5	61	285	9,616

Figure 4.4 plots the nucleon-indices per k-core (C_k) for Oregon-1, ca-AstroPh and arenas-jazz networks. To select the optimal β parameter for eq. (4.1), we use the following criteria: let's assume that SK is the set of the k-indices corresponding to the maximum nucleon-indices, as β varies in the interval [0, 1] and k increases from 1 up to k_{max}. Then, we select any β associated with the k-index which appears most often in the set SK. For example, Table 4.3 shows the set SK for arenas-jazz. We select a β corresponding to the mode k_C-index value of 25 (i.e., $\beta = 0.1$; $\beta = 0.6$; $\beta = 1.0$).

Table 4.4 shows the (k_{max}, β, k_C) indices for our social network and Internet AS datasets and Fig. 4.5 provides a visualization of our extracted core subgraphs (G_C) for several example networks. The smallest subgraph has 32 nodes and 362 edges (Oregon-1), whereas the largest one has 239 nodes and 28,441 edges (ca-HepPh). We will further investigate the structure of these core subgraphs (network nuclei) in the remaining sections.

Table 4.5: k-index to stop the shells pruning process (k_C) for several centralities: c_c - closeness centrality; b_c - betweenness centrality; e_c - eigenvector centrality; *dep* - dependence

Network	k_C			
	$\theta(i) = c_c$	$\theta(i) = b_c$	$\theta(i) = e_c$	$\theta(i) = dep$
train bombing	7	6	7	7
arenas-jazz	26	25	26	25
infectious	16	16	16	17
dnc-corecipient	68	65	68	67
Euro-road	3	3	–	3
US airports	65	65	65	65
OpenFlights	31	31	31	31
Route views	12	12	12	12
arenas-pgp	31	30	31	31
Oregon-1	18	18	18	18
ca-HepPh	99	99	99	99
ca-AstroPh	53	53	53	53
ca-CondMat	42	37	37	37
CAIDA	23	23	23	23
Internet	62	64	64	64
email-Enron	48	48	48	48
loc-brightkite	55	48	56	56
Facebook	60	60	60	61

4.4.3 Other Centralities and Nucleus

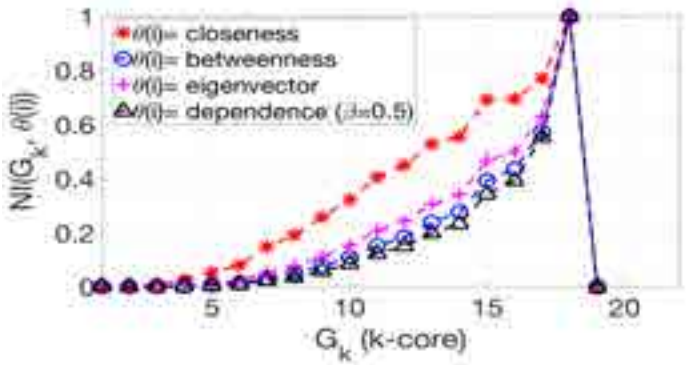
Nodes are more likely to be part of a network's core if they have high centrality score and if they are connected to other core nodes. Equation (4.2) can be used with a wide variety of $\theta(i)$ functions to transition between core and peripheral nodes. Thus, it allows one to use different ways to compute the nucleon-index (NI) and measure core quality. Here, we compute the nucleon-index using some of the most common centrality metrics: closeness centrality (c_c) [103, 104, 76], betweenness centrality (b_c) [105, 103, 76] and eigenvalue centrality (e_c) [103, 106, 107, 76] – we compare the obtained k_C-indices with the values computed in the previous section.

The closeness centrality measures how central a node is in terms of its distance (shortest path) from all other nodes [76], while the betweenness centrality for a node measures the number of shortest paths that pass through that node [76]. The eigenvalue

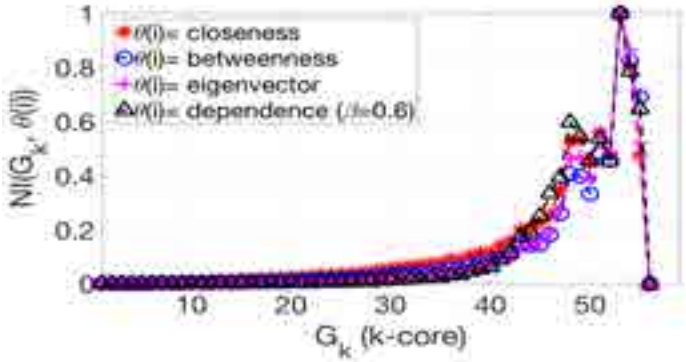
centrality computes the centrality for a node based on the centrality of its neighbors. It is based on the notion that a node should be viewed as important if it is linked to other important nodes, where a node importance (or centrality score) corresponds to the largest eigenvector of the adjacency matrix [76]. Table 4.5 shows the k_C-indices for the different centrality measures and Fig. 4.6 and Fig. 4.7 plot the nucleon-indices versus k-core indices of several example networks. In general, we observe that all the centralities give consistent k_C-indices or core structures for our datasets. In particular, we observe that our dependence metric, $dep(i, \beta)$, derives similar core structure when compared to the other metrics. From the consistency of the results given by the studied centrality metrics, we can infer that our social networks (see § **??**) truly have a core structure.

All the centrality metrics discussed here are designed to measure notions of node importance in a network. Nevertheless, they have different computational complexity and require different network information. For example, the closeness and eigenvalue centralities need the full network information and have a high complexity of $\mathrm{O}(V^3)$. The betweenness centrality has a lower complexity of $O(VE)$ [105]. Our approach to calculate the $dep(v, \beta)$ score for node v is dependent on the k-shell decomposition method and degree computation which have a complexity of $O(V + E)$. Then, given that the degree and coreness of each node are known, our procedure has a complexity of $O(E)$. For a large sparse social network with $O(n)$ edges, this yields a linear time algorithm. Therefore, our methodology is highly scalable and can be applied to massive networks.

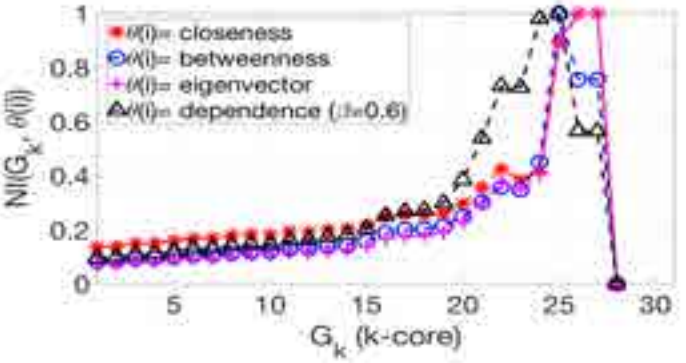
We compare our methodology to extract core subgraphs to the classical k-shell decomposition [1], rich club [108, 109] and Holme core [22] methods. Table 4.6 provides statistics for the structure of the derived core subgraphs (G_C) for six of our networks (i.e., *Oregon-1*, *ca-AstroPh*, *arenas-jazz*, *Route views*, *OpenFlights* and *US airports*) – we omit the others networks here due to space constraint. In general, for our dataset, we observe that the classical k-shell decomposition method (KS) is bias toward small and highly dense core subgraphs, G_C^{KS}, (i.e., a clique) which may not represent the "network core" (see § 4.3). In contrast, our modified k-shell decomposition method ($NI + KS$) generates larger core subgraphs than KS. In fact, our core subgraphs are supersets of the cores extracted using KS: $G_C^{NI+KS} \supset G_C^{KS}$. When compared to rich-club, we see



(a) Oregon-1

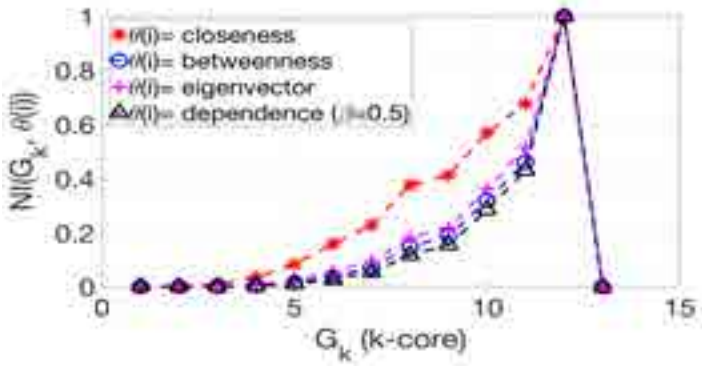


(b) ca-AstroPh

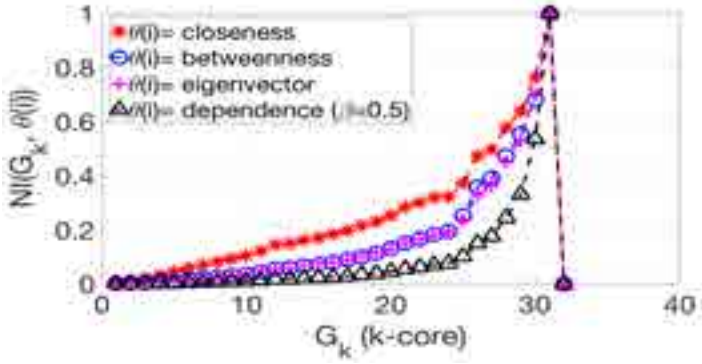


(c) arenas-jazz

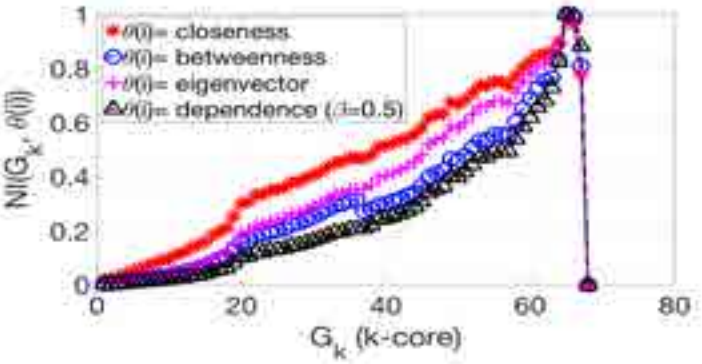
Figure 4.6: Variation of the nucleon-index (NI) per k-core index for several centrality metrics for Oregon-1, ca-AstroPh and arenas-jazz : the value of NI is normalized; the k-index to stop the shells pruning process (k_C) corresponds to the max(NI)



(a) Route views



(b) OpenFligths



(c) US airports

Figure 4.7: Variation of the nucleon-index (NI) per k-core index for several centrality metrics for Route views, OpenFligths and US airports: the value of NI is normalized; the k-index to stop the shells pruning process (k_C) corresponds to the max(NI)

that for some networks our modified k-shell decomposition method ($NI+KS$) generates core subgraphs of similar size (e.g., *Oregon-1*). However, our core subgraphs have more compact structure: small diameter, small path length and high density. For other networks, our methodology generates larger and denser core subgraphs than the rich-club method (e.g., *US airports*). This can be explained due to the fact that the rich-club is bias toward nodes with higher degree[4]. Differently, our definition of core is more general, and it allows low-degree nodes to belong to the core, as long as, they are important components in the structure of the network. When compared to Holme-core, we observe that our methodology ($NI + KS$) extracts larger core graphs for some network (e.g., *Route views*, *OpenFlights* and *US airports*), while for others it extracts denser and more compacted core structures (e.g., *Oregon-1*, *ca-AstroPh* and *arenas-jazz*).

4.5 Analysis of the Network Core Structure

Given the dense structures of our core subgraphs, illustrated in Figure 4.5, we now investigate the importance of this substructure for the network. To achieve this, we define and analyse the following metrics:

Core Path Length: To understand how much the network core contributes towards the small path lengths, we measure how many hops there are between any user to the core subgraph: $\delta(u, G_C) = min_{y \in G_C}\{d(u, y)\}$; $G_C \subset G$. Figure 4.9 presents the core path length and network path length distribuitions for *Oregon-1*, *ca-AstroPh* and *arenas-jazz*[5], whereas Table 4.7 shows the average values and the diameter for all the networks. From these results, we can see that most users are approximately 4 hops away from a random user and at most 2 hops away from the core (G_C), which implies that our core subgraphs are important structure for the connectivity of the nodes in the network.

Core Centrality: We now investigate the importance of the core subgraph for communication and information diffusion in the network. To achieve this, we use the following procedure: first, we randomly sample k unique pairs of nodes (u, v). Then, we measure, $R(u, v)$, the ratio of the distance between nodes u and v to their respective distance to

[4]Rich-club is a group of high-degree nodes in a network that preferentially connect to one another. This structure might be the core subgraph for power law networks

[5]We obtain similar results for the other datasets. We omit the plots here due to space constraint.

Table 4.6: Comparing classical k-shell decomposition (KS), Nucleon Index (NI) + k-shell decomposition (KS), Rich-Club network core and Holme-Core in real-world networks : N - number of nodes; E - number of edges; D - diameter; P - path length; ρ - density

method	dataset	N	E	D	P	ρ
Classical KS	Oregon-1	20	164	2.0	1.14	0.86
	ca-AstroPh	17	136	1.0	1.00	1.00
	arenas-jazz	30	435	1.0	1.00	1.00
	Route views	11	53	2.0	1.04	0.964
	OpenFlights	29	385	2.0	1.05	0.948
	US airports	51	1,274	2.0	1.00	1.00
NI + KS	Oregon-1	32	363	2.0	1.27	0.73
	ca-AstroPh	126	3,378	3.0	1.87	0.43
	arenas-jazz	32	466	2.0	1.06	0.94
	Route views	18	127	2.0	1.17	0.87
	OpenFlights	42	742	2.0	1.18	0.82
	US airports	81	3,073	2.0	1.05	0.95
Rich-Club	Oregon-1	37	314	3.0	1.57	0.47
	ca-AstroPh	82	994	3.0	1.80	0.30
	arenas-jazz	48	536	3.0	1.56	0.48
	Route views	21	125	3.0	1.44	0.60
	OpenFlights	45	587	3.0	1.41	0.59
	US airports	73	2,343	2.0	1.11	0.89
Holme-Core	Oregon-1	33	365	2.0	1.31	0.69
	ca-AstroPh	2,827	78,870	6.0	2.88	0.02
	arenas-jazz	46	659	2.0	1.36	0.64
	Route views	5	10	1.0	1.00	1.00
	OpenFlights	18	153	1.0	1.00	1.00
	US airports	66	2,123	2.0	1.01	0.99

Table 4.7: Summary of path length (P) and diameter (D) characteristics: $\delta(u, G_C)$ - shortest path from node u to the core subgraph G_C

Network	P	D	$Avg(\delta(u, G_C))$
train bombing	2.63	6	1.35
arenas-jazz	2.21	6	1.27
infectious	3.57	9	2.55
dnc-corecipient	2.27	8	1.63
Euro-road	19.18	62	11.60
US airports	3.14	8	1.48
OpenFlights	4.18	14	2.04
Route views	3.67	9	1.64
arenas-pgp	7.65	24	4.27
Oregon-1	3.62	10	1.54
ca-HepPh	4.67	13	2.38
ca-AstroPh	4.17	14	2.24
ca-CondMat	5.35	14	3.25
CAIDA	3.91	17	1.61
Internet	3.78	10	1.84
email-Enron	4.03	13	1.74
loc-brightkite	4.92	18	3.41
Facebook	4.31	15	2.42

Table 4.8: Ratio of the distance between nodes u and v to their respective distance to the core subgraph G_C: $R(u, v)$

Network	k	Avg($R(u, v)$)
train bombing	64	1.23
arenas-jazz	70	0.96
infectious	410	0.75
dnc-corecipient	700	0.90
Euro-road	1,039	0.85
US airports	1,574	1.13
OpenFlights	2,939	1.04
Route views	6,474	1.17
arenas-pgp	8,000	0.89
Oregon-1	8,000	1.21
ca-HepPh	8,000	1.03
ca-AstroPh	8,000	0.96
ca-CondMat	20,000	0.84
CAIDA	26,475	1.24
Internet	34,761	1.05
email-Enron	20,000	1.21
loc-brightkite	20,000	0.73
Facebook	20,000	0.92

the core subgraph, as expressed in eq.(4.4), where $d(u, v)$ represents the shortest path between u and v, and $d(u, G_C)$ or $d(v, G_C)$ represents the shortest path between u or v to the core subgraph G_C.

Table 4.8 shows the average $R(u, v)$ for $k = 70$, $k = 700$, $k = 8,000$ and $k = 20,000$ respectively. We observe that the avg($R(u, v)$) is very close to the optimal value of 1.0, which implies that our core subgraph G_C contains the nodes with the highest *betweeness* in the network and they act as "bridges" for the connectivity between the other nodes in the network.

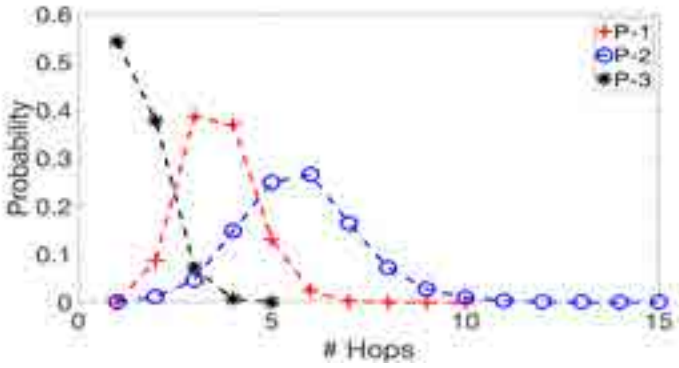
$$R(u, v) = \frac{d(u, v)}{d(u, G_C) + d(v, G_C)} \tag{4.4}$$

Core Removal: we also investigate the impact of removing the core subgraph G_C in the structure of the studied networks. We observe that all the neworks described in § **??** have a giant connected component (GCC) contaning more than 90% of all the nodes and more than 85% of all edges in the network. After the core removal, we see

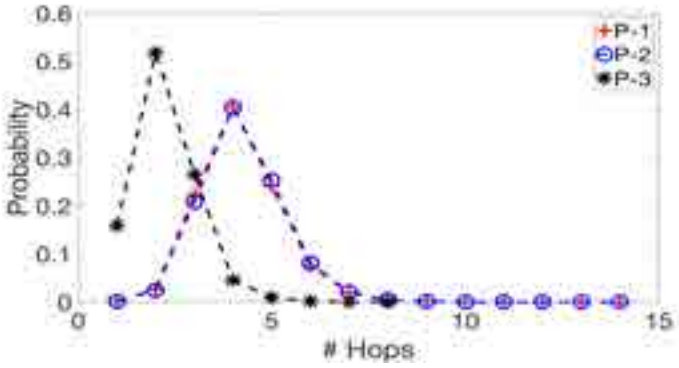
that, for some networks (i.e., arenas-jazz, dnc-corecipient, Oregon-1 and email-Enron), at least 20% of the nodes break away from GCC, forming many isolated components of smaller sizes. Table 4.9 shows the number of these new connected components per network as well as the ratio of the size of the GCC after and before call removal in terms of the number of nodes and edges. From these results, we deduce that removing G_C significantly affects the connectivity and density for some of the networks.

Figure 4.9 shows the path length distribuition after we remove the core from our networks. We observe that the average path length increases after the core removal for most of the networks. For example, *ca-AstroPh*, *email-Enron* and *Oregon-1* have average path length of 4.17, 4.03 and 3.62 before core removal, and 4.25, 4.49 and 5.72 after core removal. This result provides further evidence that the core subgraph G_C is an important structure for reachability, communication and information diffusion in these networks. Next, we discuss the implications of our results.

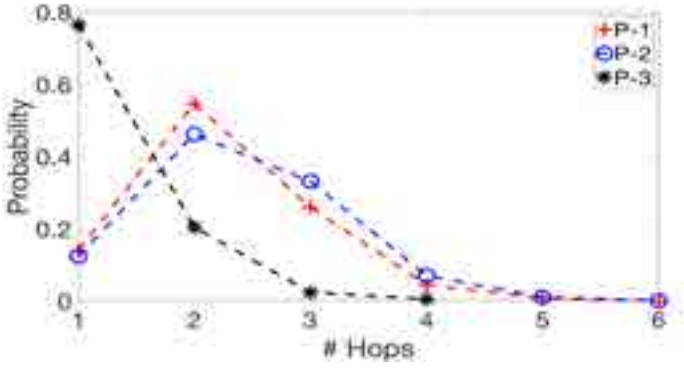
Community Structure vs Network Core: Lastly, we now investigate the importance of the core subgraph (G_C) for the formation of communities structure in a dynamic system. To achieve this, we use Newman's modularity-based algorithm [110, 27] to identify the communities structure in our network – recall that a "community" is often considered to be a subset of vertices that are densely connected internally but sparsely connected to the rest of the network. Figure 4.14(a) shows the structure of our "Route views" network. It is the Internet graph at the level of autonomous systems. Its general structure typically consists of client autonomous system and a small number of well-connected backbone nodes. This figure shows that the bulk of the nodes are placed in the periphery (yellow nodes), while a small fraction of central hubs are placed in the core (red nodes). In contrast, Fig. 4.14(b) shows the obtained Newman's communities. There are 3 large communities (i.e., blue, purple and dark grey nodes). However, most of the communities are heavily blended with each other and the core nodes are spread across the communities. Thus, our results shows that traditional community detection algorithms may not discover the core structure of the network but instead they might break the core structure of the network. Figure 4.14(c) shows the community structure of the network after removing the core nodes. We observe that the community structure of the network is destroyed in the absence of the core nodes. Hence, this results provides



(a) Oregon-1

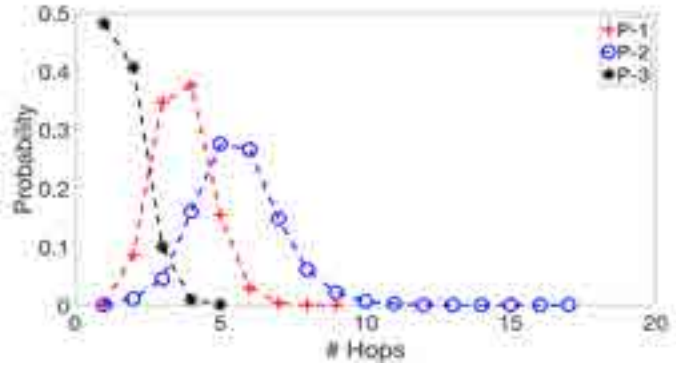


(b) ca-AstroPh

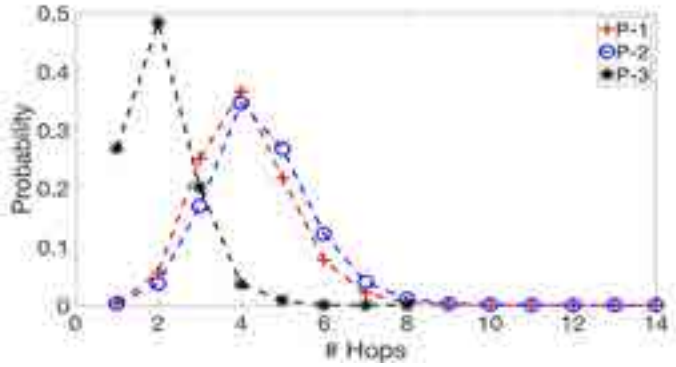


(c) arenas-jazz

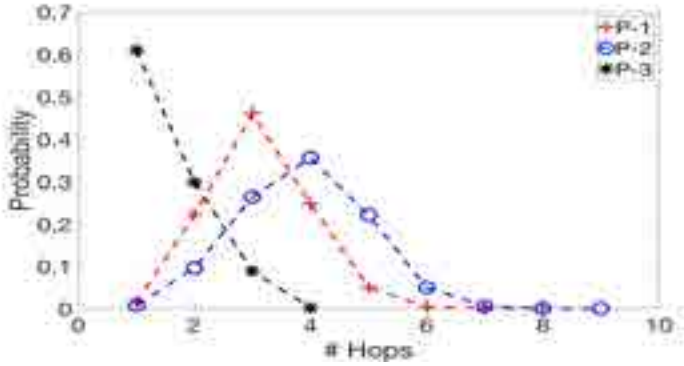
Figure 4.8: Path length distribuitions for Oregon-1, ca-AstroPh and arenas-jazz: P-1: distance between nodes in the original network; P-2: distance between nodes in the original network, after core removal; P-3: nodes distance to the core subgraph G_C



(a) Route views



(b) OpenFlights



(c) US airports

Figure 4.9: Path length distribuitions Route views, OpenFlights and US airports: P-1: distance between nodes in the original network; P-2: distance between nodes in the original network, after core removal; P-3: nodes distance to the core subgraph G_C

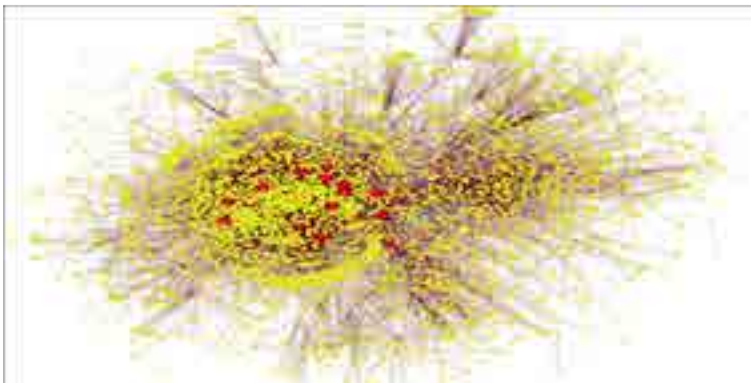
Table 4.9: Basic stats of the giant (largest) connected components (GCC) after core removal: c_n - number of connected components; n_j and n_i- number of nodes in GCC before and after core removal; e_j and e_i - number of edges in GCC before and after core removal; P_r - path length after core removal

Network	# c_n	n_i/n_j	e_i/e_j	P_r
train bombing	12	0.31	0.16	2.66
arenas-jazz	2	0.833	0.612	2.39
infectious	2	0.93	0.83	3.71
dnc-corecipient	104	0.757	0.404	2.97
Euro-road	3	0.98	0.96	19.93
US airports	246	0.79	0.33	3.86
OpenFlights	78	0.96	0.73	4.45
Route views	1,530	0.75	0.57	5.60
arenas-pgp	26	0.993	0.940	7.61
Oregon-1	3,183	0.688	0.503	5.72
ca-HepPh	73	0.967	0.645	4.87
ca-AstroPh	12	0.946	0.929	4.25
ca-CondMat	2	0.997	0.978	5.37
CAIDA	5,724	0.77	0.55	6.44
Internet	1.473	0.95	0.62	4.13
email-Enron	3,350	0.800	0.711	4.49
loc-brightkite	65	0.972	0.957	4.92
Facebook	66	0.994	0.930	4.36

further evidence that our core subgraph G_C is a fundamental structure for the formation of the network. Additionally, Fig. 4.11(a) shows the structure of our "OpenFlights" network. Our methodology uncovers the core-periphery structure of the network: green (periphery) and red (core) nodes. Figure 4.11(b) shows he obtained Newman's communities. There are 3 large communities (i.e., blue, purple and green nodes) and the core nodes are spread across the communities. Lastly, Fig. 4.11(c) shows the community structure of the network after removing the core nodes. We observe that the modularity value of the network increases after core removal. Hence, from this result, we can infer that for some networks the core subgraph masks or hides the true community structure of a network – we obtained similar results for the others networks (see Table 4.10).

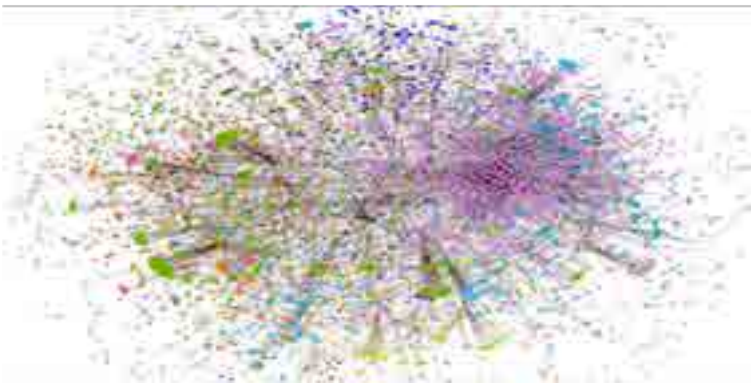
Table 4.10: Modularity values of the giant (largest) connected components (GCC) before (M_j) and after (M_i) core removal

Network	M_j	M_i
train bombing	0.417	0.594
arenas-jazz	0.294	0.392
infectious	0.699	0.613
dnc-corecipient	0.411	0.256
Euro-road	0.860	0.864
US airports	0.222	0.517
OpenFlights	0.596	0.880
Route views	0.570	0.743
arenas-pgp	0.869	0.882
Oregon-1	0.559	0.778
ca-HepPh	0.600	0.727
ca-AstroPh	0.569	0.584
ca-CondMat	0.710	0.700
CAIDA	0.599	0.887
Internet	0.549	0.722
email-Enron	0.52	0.668
loc-brightkite	—	—
Facebook	—	—



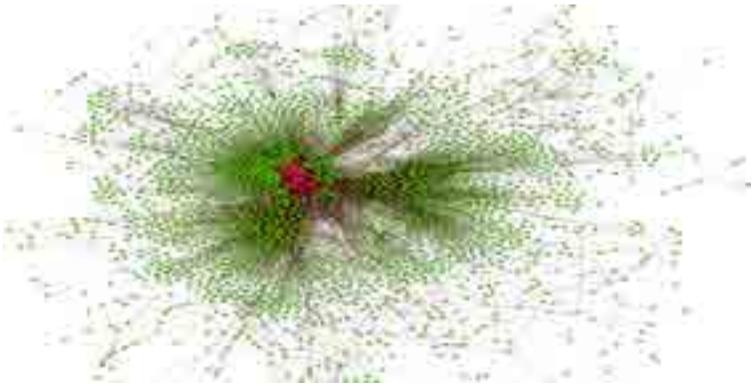
(a) Core-periphery structure

(b) Newman communities

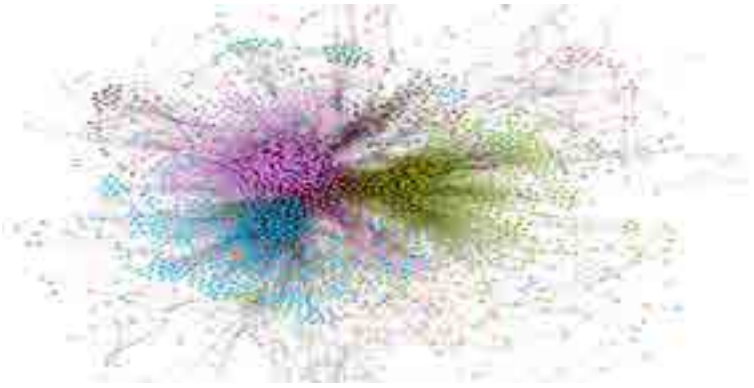


(c) Newman communities after core removal

Figure 4.10: Visualization of the network structure for the "Route views" network: a) core-periphery structure: red (core) and yellow (periphery) nodes; b) Newman community structure before core removal; c) Newman community structure after core removal



(a) Core-periphery structure



(b) Newman communities

(c) Newman communities after core removal

Figure 4.11: Visualization of the network structure for the "OpenFlights" network: a) core-periphery structure: red (core) and green (periphery) nodes; b) Newman community structure before core removal; c) Newman community structure after core removal

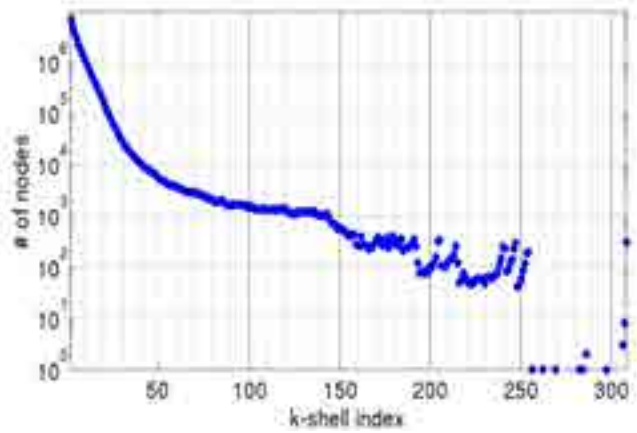


Figure 4.12: The k-shell decomposition method on the reciprocal network of Google+ (subgraph H_1). For each k-shell, we plot the number of nodes belonging to the k-shell as k varies from 1 to $k_{max} = 308$.

4.6 Scalability Analysis

In this subsection, using Google+ (G+) as a case study, we show that our algorithm can be applied to massive graphs with 10s or 100s of millions of nodes and more than 1 billion edges. Using our Γ_i graphs[6] (see Section 4.2), we first apply our procedure to extract the core subgraph of G+. We then analyze how the core structure evolves over time using three different snapshots ($\Gamma_{i=1,2,3}$).

In extracting the core subgraph of G+, we focus our analysis on its *reciprocal* network – namely, the *bidirectional* subgraph formed by the reciprocal edges among users in G+. Based on a massive Google+ dataset (see Sect. 4.2 for a brief overview of Google+ and a description of the dataset), we find that out of more than 74 million nodes and $\approx$ 1.4 billion edges in (a snapshot of) the directed Google+ OSN, more than two-third of the nodes are part of Google+'s *reciprocal* network and more than a third of the edges are reciprocal edges (with a reciprocity value of roughly 0.31). This reciprocal network contains a *giant connected subgraph* with more than 40 million nodes and close to 200 million edges (see Sect. 3.4.1 for more details). The main characteristics of the reciprocal network of Google+ (subgraphs $H_{i=1,2,3}$) are summarized in Table 4.11, where density is

[6]For clarity of notation, we sometimes drop the subscript index i, unless we are referring to a specific snapshot $i > 1$

Table 4.11: Main characteristics of the reciprocal network of Google+: H

ID	# nodes	# edges	max(degree)	density
H_1	40,403,216	197,838,519	4,294	2.42×10^-7
H_2	49,161,409	226,373,003	4,425	1.87×10^-7
H_3	74,539,728	327,204,637	4,743	1.78×10^-7

defined as $|E|/[|V|(|V|-1)$ for a *directed* graph, and $2|E|/[|V|(|V|-1)$ for an *undirected* graph here $|V|$ is the number of nodes and $|E|$ is the number of edge.

In a sense, a reciprocal network can be viewed as the stable "skeleton" network of the directed OSN that holds it together. Hence, we are interested in analyzing and uncovering the *core* structural properties of the reciprocal network of a directed OSN, as they could reveal the possible organizing principles shaping the observed network topology of an OSN [17]. For example, using the *core*, we can build network models that can help us to understand the topological features of the nodes and structural properties of the network, as well as, to predict the topological growth of the network and provide upper bounds of the distance between the nodes – see the jellyfish model of the Internet in [33]. Furthermore, unveiling the core structure (referred to as the "nucleus") of a reciprocal network may have implications in the design of algorithms for information flow, and in development of techniques for analyzing the vulnerability or robustness of OSNs.

We apply the classical k-shell decomposition method to the Google+ reciprocal network for subgraph H_1 (we analyze the other subgraphs in Sect. 5.4). We find that the $k_{max} = 308$, and the k_{max}-core is a clique of size 290 nodes (the maximum clique in the Google+ reciprocal network). Figure 4.12 shows the number of nodes belonging to the k-shell as k varies from 1 to 308: we see that 99% of the nodes in our network fall in the lower k-shells (from $k = 1$ to 100). This is not surprising, as the majority of the nodes in our network have degree less than 100. Figure 4.13(a) shows the average degree of nodes in the k-shell, whereas in Fig. 4.13(b) we zoom in on nodes with $deg(v) \geq 1000$, and illustrate how they distribute across various k-shells. We see that while a large portion of high-degree nodes belong to higher k-shells, in fact the highest degree nodes belong to lower k-shells, suggesting that they do not lie at the "core" of the Google+ reciprocal network. However, as we discussed in Sect. 4.3 directly applying this method

to the Google+ reciprocal yields a final graph – a clique of 290 nodes (the maximum clique of the Google+ reciprocal network) that consists of a close-knit community of users in Taiwan – which is unlikely to lie at the "core" of the Google+ reciprocal network (see discussion on the next chapter, where we show this clique in fact lies more at the outer ring of Google+'s dense core structure).

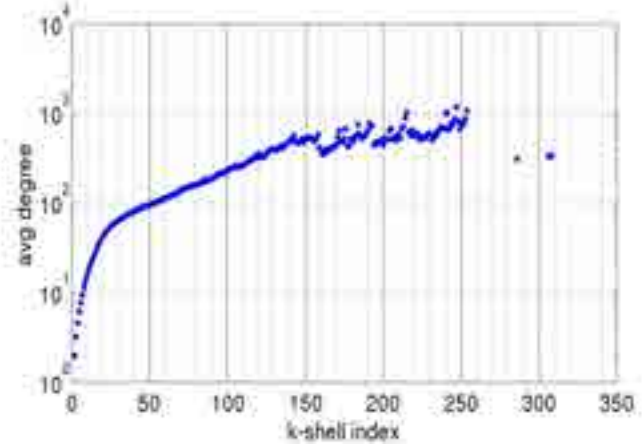
To extract a meaningful core for our Google+ dataset, we re-formulated our nucleon-index (NI) for massive graphs as follows:

$$NI(G_k, \theta(i)) := \frac{V_k}{V_G} \times \rho(D_k) \times \{\sum_{i \in G_k} \theta(i)\} \tag{4.5}$$

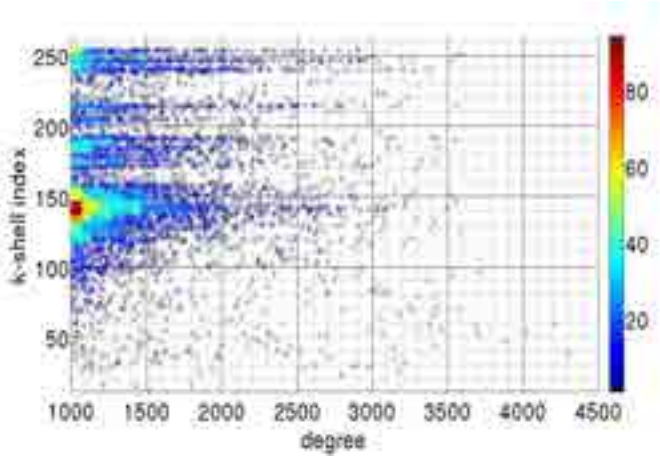
$$D_k := \frac{E_k}{V_k \times (V_k - 1)} \tag{4.6}$$

$$\rho(D_k) := 1 - e^{S \times D_k} \tag{4.7}$$

where by abuse of notation, we use E_k to denote the number of edges between nodes in G_k and V_k the number of nodes in G_k (and $|V_K| = 0$). The first term is penalty parameter and it takes into account the proportion of nodes excluded from G - it favors large cores. The second term in eq.(4.2) measure the density of G_k and the last term is the sum of the centrality values of the nodes in G_k (see also appendix B). Ideally, if G_k is a "dense core" of G, the product of these two terms should be large. Hence, G_k with the largest NI represents the *nucleus* of G (as produced by the decomposition process). Applying eq.(4.2) to our dataset and the statistics of our core subgraphs are illustrated in Table 4.12. In the next chapter, we dissect the structure of these subgraph in order to understand how these networks are formed. Figure **??** shows the variation of the nucleon-index per k-core index for our subgraphs $H_{i=1,2,3}$.



(a) Average degree of nodes in the k-shells

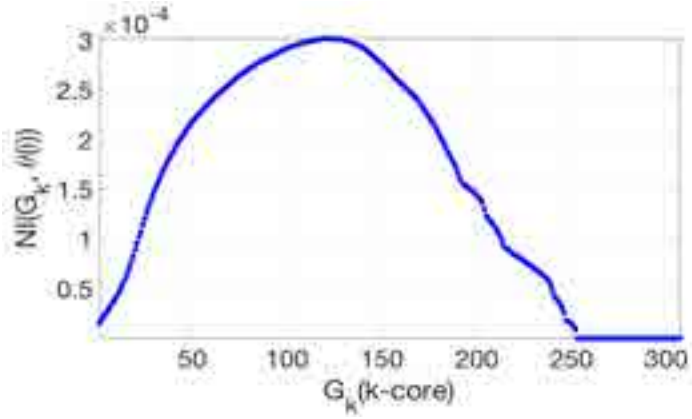


(b) K-shell distribuition of the nodes with $deg(v) \geq 1000$

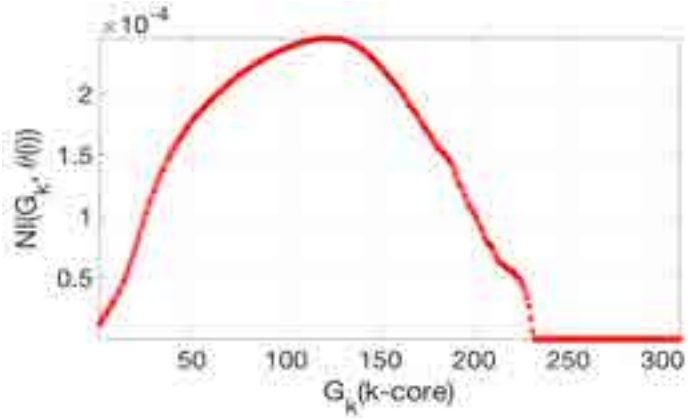
Figure 4.13: The k-shell decomposition method on the reciprocal network of Google+ (subgraph H_1). We plot the degree distributions for nodes in the k-shells, as k varies from 1 to $k_{max} = 308$: a) average degree of nodes in the k-shells, b) we zoom in on nodes with $deg(v) \geq 1000$, and illustrate how they distribute across various k-shells.

Table 4.12: Main characteristics of the core subgraph (G_C) for the reciprocal network of Google+ across several snapshots.

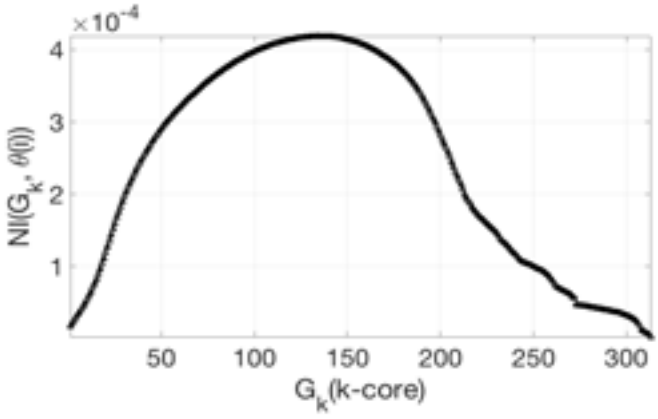
H_i	k_C	# nodes	# edges	avg(d)	density
1	120	48,229	6,378,596	132	0.00548
2	120	52,904	6,737,630	127	0.00482
3	130	94,112	14,260,691	152	0.00322



(a) H_1: maximum NI at $k_C = 120$



(b) H_2: maximum NI at $k_C = 123$



(c) H_3: maximum NI at $k_C = 134$

Figure 4.14: Variation of the nucleon-index(NI) per k-core index: the k-index to stop the shells pruning process (k_C) corresponds to the max(NI)

4.7 Discussion

Using examples from communication networks as well as collaboration, location-based, interaction, and online social networks, we have demonstrated that our method can effectively uncover and extract the nucleus of these networks. In this section, we discuss the limitations and implications of our method and results.

First, our proposed methodology to uncover the nucleus of networks can also be applied to weighted and directed networks by using a variation of the k-shell decomposition method: Garas et al. [71] presented a weighted k-shell decomposition method and Batagelj et al. [111] generalized the k-shell decomposition to directed networks. Our method can be applied with these generalized algorithms because our dependence and nucleon-index metrics are independent to the k-shell decomposition method. Once the k-shells are provided by decomposing the network into k-layers, the dependence and nucleon-index values can be computed.

Second, the "coreness" centrality or k-shell index has been argued to be a better measure than node degree for identifying influential spreaders in a network [74, 75]. However, our results show that using k-shell indices as a predictor of spreading influence of a node can be misleading. This is due to the fact that for a node to have a high k-shell index, it just needs to be a part of a very strong structure (e.g., a clique). This structure, however, may be isolated and lie at the edge or periphery of the network, instead of its core (see § 4.3). Our analysis shows that the dependency value of a node, $dep^k(i)$, provides important information about the structure function of each node in the graph. Thus, we believe that by using a node dependency value along with its k-shell index (dep^k, k), we can better predict the spreading influence of a node than simply using its k-shell index. We will investigate this in the future.

Third, unveiling the core structure of social networks may have implications in the design of algorithms for information flow, and in development of techniques for analysing the vulnerability or robustness of networks. In addition, analysis of the core structure of social networks can help us uncover and understand possible organizing principles shaping the observed network topological structure and network formation.

4.8 Summary

In this paper, we have advanced and developed an effective procedure to extract the *core* structure of social networks. First, we introduce a new metric – the node "dependence value" – that measures the location importance of a node in a network. Second, we define a new measure called *nucleon-index* that captures the extend to which a subgraph is a densely intra-connected and topological central core. Then, using these metrics, we proposed a modified version of the k-shell decomposition method by identifying the k_C-index where we should stop pruning the network in order to preserve its core structure. For our social network datasets, we found that they contain very dense core subgraphs G_C. The smallest core has 32 nodes and 362 edges (Oregon-1), whereas the largest one has 239 nodes and 28,441 edges (ca-HepPh). Finally, given a dense core subgraph G_C, we investigate the importance of this substructure for the network by analysing the following metrics: i) the distance between a node v to the core subgraph G_C; ii) the ratio of the distance between nodes u and v to their respective distance to G_C and iii) lastly, the impact of removing G_C in the structure of the network G ($G_C \subset G$).

As part of ongoing and future work, we will provide a more in-depth analysis of the dense core subgraph G_C of social networks. We also plan to apply our method to a massive Google+ dataset [14, 11, 100] (with more than 170 million nodes and ≈ 3 billion edges), a massive Twitter dataset [112] (with more than 500 million nodes and ≈ 23 billion edges) and other social networks.

Chapter 5

Dissecting the Nucleus of Complex Networks using (Hyper)Graphs

5.1 Introduction

Many complex networks are observed to have a core-periphery structure [22, 23, 24]. In Chapter 4, we present our scheme to extract this meso structure in complex networks. For a massive Google+ reciprocal network with more than 40 million nodes and close to 200 million edges (see Chapter 4 for more details) with uncovered very dense core subgraphs (G_C) – from 48,229 nodes and 6,378,596 edges to 94,112 nodes and 14,260,691 edges. Existence of this dense core sub(graph) in the reciprocal network of Google+ raises many interesting and challenging questions. How is this network core formed? What does this structure look like?[1].

In an attempt to address these questions (or challenges), we develop an effective two-step procedure to *hierarchically* extract and unfold the *core* structure of Google+'s

[1]To answer these questions, for networks of tens or hundreds of vertices, it is a relatively straightforward matter of drawing and examining a picture of the network either by hand or with computer rendering tools [113]. However, for networks of million or a billion vertices, however, this approach is useless.

reciprocal network[2], building up and generalizing ideas from the clique percolation approach [**?**] as follows: i) Given this dense "core" subgraph of the Google+ reciprocal network, we first compute the maximal clique that each node is part of (using a simplified Bron-Kerbosh algorithm), and then form a new *directed* (hyper)graph – a form of clique percolation [**?**], where the vertices are (unique) cliques of various sizes, and there exists a directed edge from clique C_i to clique C_j if half of the nodes in C_i are contained in C_j (see Sect. **??**). This new (hyper)graph provides a higher-level representation of the dense core graph of the Google+ reciprocal network: the intuition is that the maximal clique containing each node v represents the most stable structure that node v is part of, and the directed edge in a sense reflects the "attraction" (or "gravitational pull") that one clique (constellation) has over the other. We find that this (hyper)graph of cliques comprises of 1700+ connected components (CCs). ii) Considering these CCs as the core "community" structures (a dense cluster of cliques) of the Google+ reciprocal network, we define three metrics to study the relations among these CCs in the underlying Google+ reciprocal network: the number of nodes shared by two CCs, the number of nodes that are neighbors in the two CCs, and the number of edges connecting these neighboring nodes (see Sect. **??**). These metrics produce a set of new (hyper)graphs that succinctly summarize the (high-level) structural relations among the core "community" structures and provide a "big picture" view of the core structure of the Google+ reciprocal network and how it is formed. In particular, we find that there are ten CCs that lie at the center of this core structure through which the other CCs are most richly connected. Additionally, our results shows that directly applying standard k-shell decowomposition method to the Google+ reciprocal yields a final graph – a clique of 290 nodes (the maximum clique of the Google+ reciprocal network) that consists of a close-knit community of users in Taiwan – which is unlikely to lie at the "core" of the Google+ reciprocal network (see discussion in Sect. **??**, where we show this clique in fact lies more at the outer ring of Google+'s dense core structure). We also find that the core structure of the Google+ reciprocal network is very stable as the network evolves (see Sect. 5.4). We discuss implications and related work in Sect. 3.5 and Sect **??**. In Sect. **??**, we conclude the paper with a brief discussion of the future work.

[2]Our methodology can also be applied to others massive online social networks.

We summarize the major contributions of our paper as follows. To the best our knowledge, our paper is the first study on the core structure of a "reciprocal network" extracted from a massive *directed* social graph. While this paper focuses on Google+, our approach is also applicable to other directed OSNs.

- We develop an effective two-step procedure to *hierarchically* extract and unfold the *core* structure of a reciprocal network arising from a directed OSN.
- We apply our method to the reciprocal network of the massive Google+ social network, and unfold its core structure. In particular, we find that there are ten subgraphs ("communities") comprising of dense clusters of cliques that lie at the center of the core structure of the Google+ reciprocal network, through which other communities of cliques are richly connected; together they form the core to which other nodes and edges that are part of sparse subgraphs on the peripherals of the network are attached.
- We observe that the core structure of the Google+ reciprocal network is very stable as the network evolves: the size of the core communities (hyper)graph increases as the network evolves, as well as, its density. Additionally, the set of nodes that participates in the core is very stable over time, with few percentage of nodes (e.g: 5% and 9%) that move away from the core to the periphery as the network evolves.
- We observe that the number of communities lying at the center of the core structure of the Google+ reciprocal network is also very stable: it increases from 10 to 11 core communities across snapshots $H_1 \rightarrow H_2$ and from 11 to 13 core communities across snapshots $H_2 \rightarrow H_3$ in the core communities (hyper)graphs.

5.2 Constructing the Core Clique (Hyper)Graph

Given the dense "core" subgraph G_{120} (extracted in the previous chapter), we use "maximal cliques" as the basic atomic structures of the network nucleus[3]. Using these

[3]In this paper we use the terms "core" and "nucleus" interchangeable

substructures, we build (hyber)graphs that provide us with a higher-level representation of the dense core graph of the networks. To achieve this, we proceed as following:

First, to find the largest maximal clique containing a given vertex in a network, we implement algorithm 1. It uses a variation of the popular Bron-Kerbosh algorithm [**?**] (we denote it as Simplified Bron-Kerbosh (SBK)) to extract maximal cliques. During the search for the largest maximal clique containing a given vertex v (thereafter referred to as C^v in short), our heuristic removes the vertices that cannot form cliques larger than the clique stored in the variable C_{max}. Furthermore, our algorithm considers only the set of neighbors of v that share at least one edge to another vertex adjacent to v at each step, instead of recursively considering all neighbors of v, and thus is much faster. This set (denoted by $N^i(v)$) is sorted in decreasing order based on the number of shared neighbors between v and $u \in N^i(v)$ for the following reason: in a relatively fairly connected subgraph, a vertex with the largest number of shared nodes with v is more likely to be a member of C^v compared to any other. Then, in the worst case, algorithm 1 loops over the complete set $N^i(v)$ at most $\triangle$ (max degree in the graph), calling the subroutine SBK at most $\triangle$. Thus, the time complexity of our heuristic is bounded by $O(\triangle^2)$. Using algorithm 1, we develop a procedure to extract the minimal set of the largest maximal cliques that cover every node in a given graph (algorithm 2). The resulting set of cliques returned from this method is always guaranteed to contain at least a unique node per clique. We apply this procedure to subgraph G_{120} and obtain 34,501 maximal cliques with an average clique size of 23.03 nodes. Figure 5.1 shows the clique size distribution.

Second, using the extracted 34,501 maximal cliques, we generate a new *directed* (hyper)graph, where the vertices are (unique) cliques of various sizes, and there exists a *directed* edge from clique C_i to clique C_j if more than half of the nodes in C_i are contained in C_j, i.e., $C_i \rightarrow C_j$ if $(|C_i| \cap |C_j|)/|C_i| \geq \theta = 0.5$. We vary the parameter θ from 0.5 to 0.7, and find that it does not fundamentally alter the connectivity structure of the (hyper)graph of cliques thus generated. We remark that the maximal clique containing each node v can be viewed as the most stable structure that node v is part of. The directed (hyper)graph of cliques captures the relations among these stable structures each node is part of: intuitively, each directed edge in a sense reflects the attraction (or gravitational pull) that one clique (a constellation of nodes) has over the other. Hence,

Algorithm 1 Largest Maximal Clique Extraction algorithm (LC)

Input: node u
Output: largest maximal clique containing u
R : currently growing maximal clique
$P := N[u]$: set of neighbors of vertex u
procedure LC(u)
    $N^i(u) = \{w_i, w_i, ... | w_{k=i,j..} \in N(u) \bigwedge d^u(w_i) > d^u(w_j)\}$
    $C_{max} = 0$
    $max = 0$
    for $w \in N^i(u)$ **do**
        $R = [u]$
        $P = N[w]$
        $C = SBK(R, P, max)$
        $k = size(C)$
        if $k > max$ **then**
            $C_{max} = C$
            $max = k$
    return C_{max}

Subroutine: Simplified Bron-Kerbosh (SBK)

procedure SBK(R, P, max)
    if $size(R) + size(P) \leq max$ **then**
        return ▷ it is not possible to find a clique larger than max
    else if $P := 0$ **then**
        report R as a maximal clique
    else
        Let u_{new} be the vertex with highest number of neighbors in P
        $R_{new} := R \cup \{u_{new}\}$
        $P_{new} := P \cap N[u_{new}]$
        SBK(R_{new}, P_{new}, max)

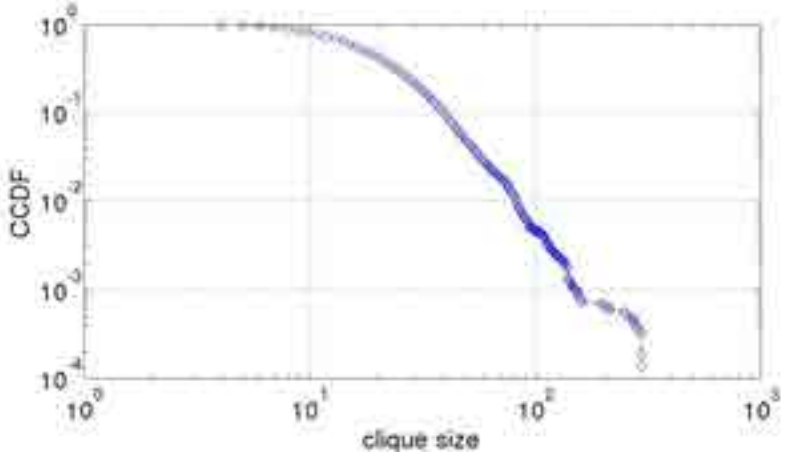


Figure 5.1: Log-log plot of clique size complementary cumulative distribution function (CCDF) for the core subgraph G_{120} (extracted from H_1) – we extract these cliques using algorithms 1 and 2.

Algorithm 2 Extract Minimal Set of Maximal Cliques from a Graph

1: **procedure** EMC($G(V, E)$)
2: construct a set W and $W := V$
3: construct a ordered list S of the nodes in V based on their degree (decreasing order)
4: select the first item in S, vertex i, as the pivot
5: apply the LC algorithm using i as the pivot vertex
6: add the reported maximal clique c_i containing i to the clique set $C_{total} = [c_n, c_m, ..]$
7: remove the nodes in c_i from $W : W_j = W_i - c_i$
8: select the next item in S, vertex j, as the next pivot vertex such that $j \notin C_{total}$ and repeat steps(5), (6) and (7) until $W = \emptyset$

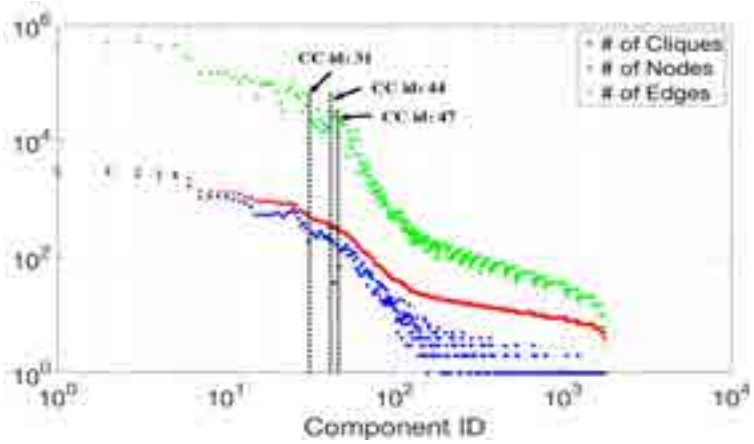
this (hyper)graph of cliques provides us with a higher-level representation of the dense core graph of the Google+ reciprocal network – how the most stable structures are related to each other. This procedure can be viewed as a form of clique percolation [**?**].

We find that this (hyper)graph of cliques comprises of 1,758 connected components (CCs). The largest component has 2,618 cliques, 3,295 nodes and 437,867 edges, while the smallest has 1 clique, 3 nodes and 3 edges respectively. We regard these connected components (CCs) as forming the *core communities* of the core graph of the Google+ reciprocal graph: each CC is composed of either one single clique (such a CC shares few than half of its members with other cliques or CCs), or two or more cliques (stable structures) (where one clique shares at least half of its member with another clique in the same CC, thus forming a closely knit community). Figure 5.2(a) shows the distributions of these components in terms of the number of cliques, the number of nodes and the number of edges. We observe that for CC id's from 1 to 100 (which contains 30 or more cliques), there is a strong correlation between the number of cliques, nodes and edges: in general the connected components with the highest number of cliques also have the highest number of nodes and edges.

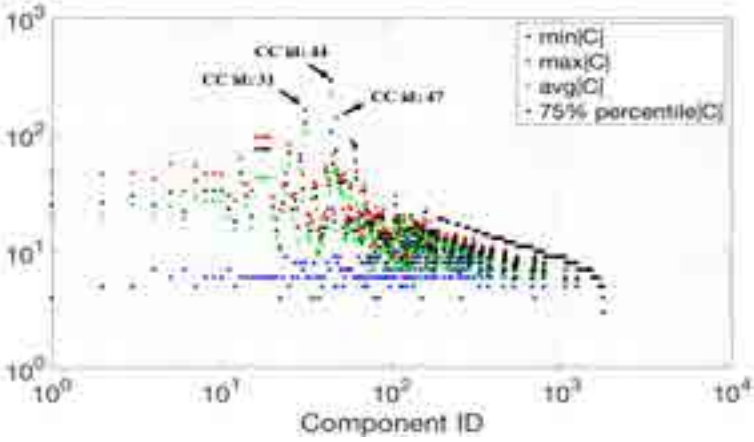
Figure 5.2(b) shows the maximum, minimum, average and 75% percentile of clique size for each CC. We observe that there is not a relationship between the number of cliques and their respective sizes in the CCs. We observe that most cliques have sizes between 10 and 100 nodes. There are largest CCs composed with a huge number of cliques of small size (e.g., CC ids from 1 to 10), whereas there are also small CCs composed with few number of cliques but with very large sizes (e.g. CC ids: 31, 44, and 47). We note also that there are a number of CCs which contain only one clique, but some of these cliques are of large size also.

5.3 Analysis of the Core Community (Hyper)Graph & its Structure

We now investigate the relationship between the connected components (CCs) in our clique (hyper)graphs constructed in the previous section (Sect. **??**), in particular the 70th largest CCs. Recall that we regard the CCs in the clique (hyper)graphs as forming the core communities within Google+ reciprocal network nucleus – each CC represents



(a) Number of cliques, nodes and edges



(b) Clique size: maximum, minimum, average and 75% percentile

Figure 5.2: Statistics of the connected components in the (hyper)graph of cliques constructed from the core subgraph G_{120} (extracted from H_1): a) distribution of the number of cliques, nodes and edges and b) distribution of the clique size in terms of the maximum, minimum, average and 75% percentile of the clique size.

a dense cluster of cliques. In this section, we define three metrics to study the relations among these CCs in the underlying Google+ reciprocal network:

- **Shared Nodes**: the number of nodes that CC_i and CC_j have in common:

$$S(CC_i, CC_j) = |\{u \in V | u \in CC_i, u \in CC_j\}| \tag{5.1}$$

- **Shared Neighbors**: the number of nodes in CC_i that have an edge to another node in CC_j:

$$N(CC_i, CC_j) = |\{u \in CC_i, |\exists v \in CC_j : (u, v) \in E\}| \tag{5.2}$$

- **Cross-Edges**: the number of cross edges between two connected components (CC_i and CC_j):

$$B(CC_i, CC_j) = |\{(u, v) \in E | v \in CC_i, u \in CC_j\}| \tag{5.3}$$

These metrics produce a set of three new (hyper)graphs that succinctly summarize the (high-level) structural relations among the core community structures: 1st) a node represents a CC and an undirected edge $CC_i - CC_j$ denotes that both components share at least one node; 2nd) a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ denotes that CC_i has the largest number of cross edges to nodes in CC_j; 3rd) a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of neighboring nodes to nodes in CC_j. These (hyper)graphs provide a "big picture" view of the core graph of the Google+ reciprocal network and yield insights as to how it is formed.

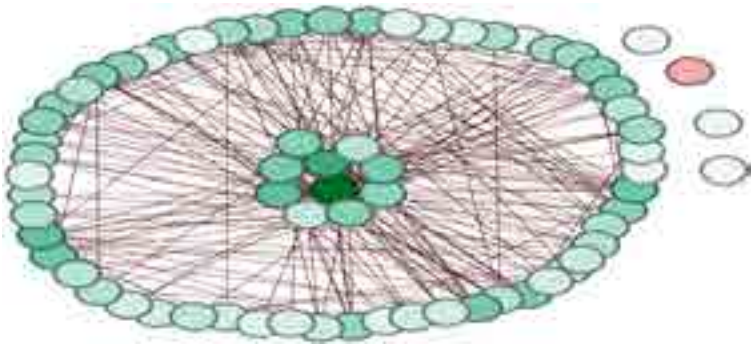
Figures 5.3(a), 5.3(b), 5.3(c) show the (hyper)graphs of the relationship between the components based on the number of shared nodes, cross-edges and shared neighbors. These figures show that there are ten subgraphs (core communities") comprising of dense clusters of cliques that lie at the center of the nucleus of the Google+ reciprocal network, through which other communities of cliques are richly connected. Then, the 1,758 connected components (CCs) in the clique (hyper)graph form the core graph of the Google+ reciprocal network, to which other nodes and edges that are part of sparse subgraphs on the peripherals of the network are attached. Table 5.1 shows a summary of the statistics for the ten CCs, respectively. We observe that the largest CC has 2,618

Table 5.1: Summary of the statistics for the ten components that lie at the center in the core graph of the reciprocal network of Google+. Together they form the core to which peripheral sparse subgraphs are attached.

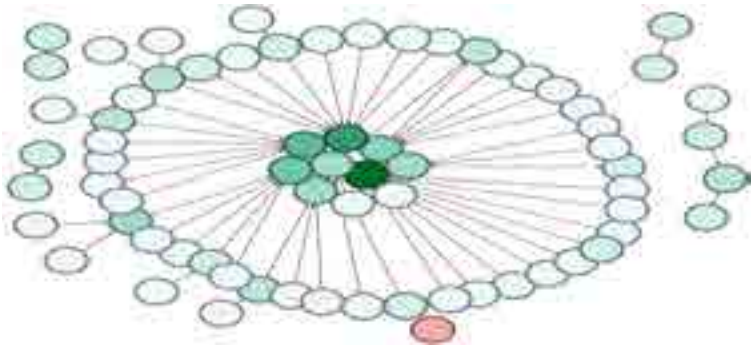
ID	# c	# nodes	# edges	avg $\|c\|$	max $\|c\|$	min $\|c\|$	75% percentile
1	2,618	3,295	437,867	30.0	47	4	25
2	2,745	3,256	494,867	20.2	46	5	26
3	2,437	3,059	499,356	25.5	47	5	30
4	2,324	2,877	416,098	20.2	42	7	25
5	2,340	2,737	449,225	24.3	56	6	32
7	1,040	1,362	146,151	29.2	55	5	40
15	513	923	60,191	16.0	33	6	20
22	473	808	32,031	10.0	23	4	11
37	262	396	14,324	9.2	15	4	10
47	69	297	22,629	50.3	139	5	73

cliques, 3,295 nodes and 437,867 edges, while the smallest has 69 cliques, 297 nodes and 22,629 edges. The set of components in table 5.1 contains some of the largest CC in our clique (hyper)graph.

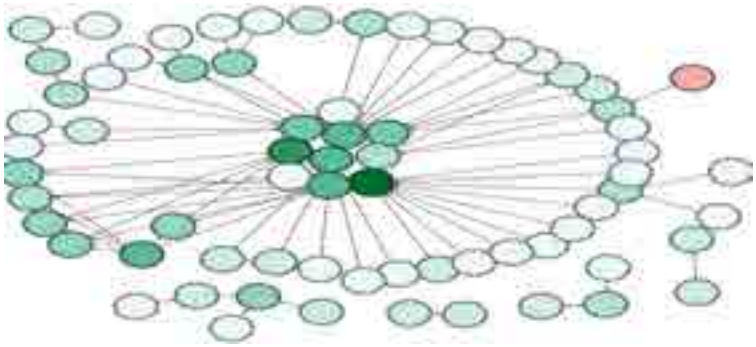
From figures 5.3(a), 5.3(b) and 5.3(c), we observe that in the periphery of our core communities (hyper)graphs, we find a small CC composed with 36 of the largest cliques in the Google+ reciprocal network. The average, minimum and maximum sizes of the cliques in this CC are 227, 105 and 290 – the latter is the maximum clique of the Google+ reciprocal network. This CC is highlighted by a "red circle" in the (hyper)graphs in Fig. 5.3. It shows this CC lies more at the outer ring of Google+'s dense core structure. As mentioned earlier in Sect. 4.3, the 290 users in this maximum clique of the Google+ reciprocal network belong to a single institution in Taiwan where every user follows every other. The users in this clique also form close relations with many other users, forming 35 other cliques. Together, these 35 cliques form a close-knit community. However, we see that this community in fact does not lie at the very "center" – instead lies more at the outer ring – of the core graph of the Google+ reciprocal network. Hence, we see that simply applying the conventional k-shell decomposition method to the Google+ reciprocal network would yield the maximum clique in the Google+ reciprocal network, but not its *core* structure. In contrast, the ten CCs mentioned above more likely lie at the "center" of the core graph of the Google+ reciprocal network.



(a) Shared nodes: a node represents a CC and an undirected edge $CC_i - CC_j$ denotes that both components share at least one node.



(b) Cross-edges: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of cross edges to nodes in CC_j.



(c) Neighboring nodes: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of neighboring nodes with CC_j.

Figure 5.3: (Hyper)Graphs for the core communities (extracted from G_{120}) of the reciprocal network of Google+: snapshot - H_1. The color intensity of a CC is proportional to its degree. The CC highlighted in "red" is the core subgraph yielded by directly applying the standard k-shell decomposition to Google+'s reciprocal network. However, our core communities (hyper)graphs show that this structure in fact does not lie at the very "center" – instead lies more at the outer ring – of the core graph of the Google+ reciprocal network.

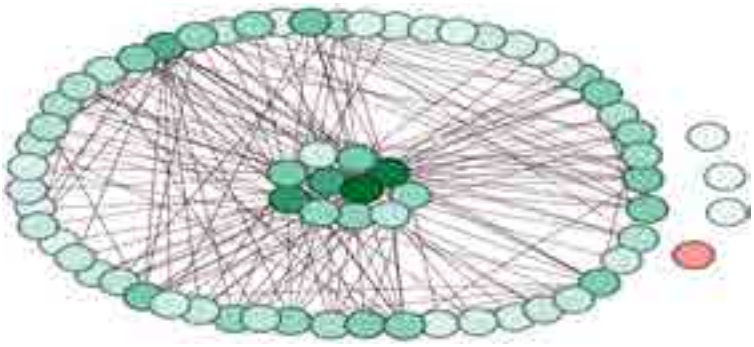
5.4 Evolution of the Core Community (Hyper)Graph

We now analyze how the core structure of the Google+ reciprocal network evolves over time using the remaining snapshots of subgraph H ($H_{i=2,3}$). To achieve this, we apply our methodology to uncover the core communities (hyper)graph for H_i. Table **??** shows the k_C-indices where we stop the k-shell decomposition method and provides statistics for the core subgraph (G_C) of the reciprocal network of Google+ across three different snapshots. We observe that the size of the nucleus increases as the network evolves, as well as, its density – although, we see a slight decrease at H_2 (this correlates with the release of a new Google+ feature reported by the authors in [**?**]). Table 5.2 provides statistics for the core communities (hyper)graphs. We observe that the number of cliques in the core subgraph (G_C) increases as the network evolves. Similarly, the number of core communities (CC) and the size of the largest CC in the clique (hyper)graph increase as the network evolves. In contrast, the size of the smallest CC remains the same across all the snapshots.

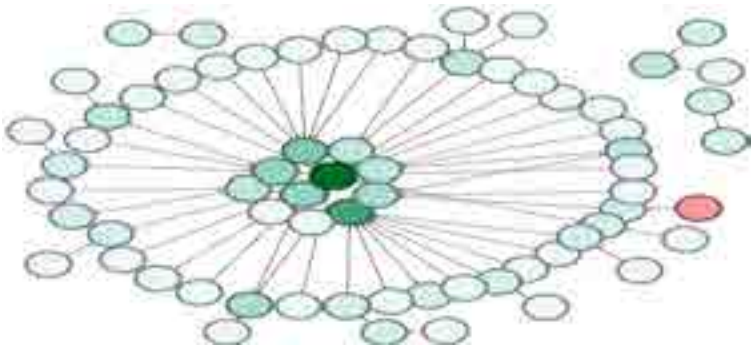
Analyzing the nodes that are found in the nucleus, we find that the set that participates is very stable over time. We find changes consisting of a few percentage of nodes that moved from the nucleus to a lower k-shell as the network evolves: 9% from $H_1 \rightarrow H_2$ and 5% from $H_2 \rightarrow H_3$. We also observe that the main structure of the core communities (hyper)graph is stable across all the snapshots: it consists of dense clusters of cliques that lie at the center of the core graph, through which other communities of cliques are richly connected. Additionally, we observe that the number of the most central communities in the core communities (hyper)graphs is also very stable: it increases from 10 to 11 across snapshots $H_1 \rightarrow H_2$ and from 11 to 13 across snapshots $H_2 \rightarrow H_3$. Lastly, we see that the community containing the "maximum clique" remains in the periphery of the core subgraph as the network evolves – see Fig. 5.4 and Fig. 5.5 for illustrations.

5.5 Summary

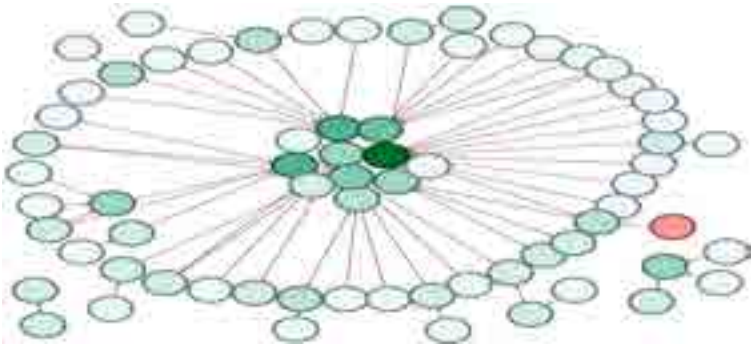
In this paper, we have developed an effective three-step procedure to *hierarchically* extract and unfold the *core* structure of the reciprocal network of Google+. We first applied a modified version of the k-shell decomposition method to prune nodes and



(a) Shared nodes: a node represents a CC and an undirected edge $CC_i - CC_j$ denotes that both components share at least one node.

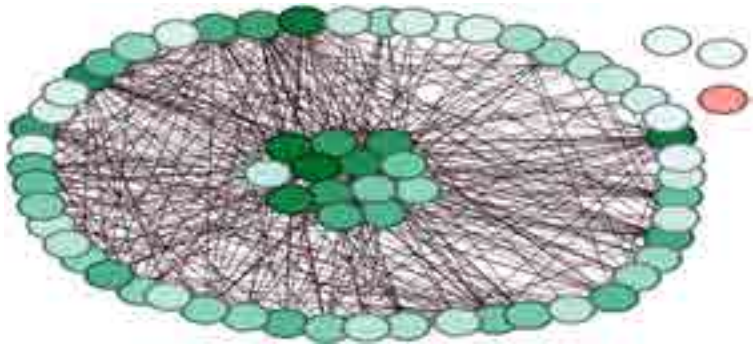


(b) Cross-edges: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of cross edges to nodes in CC_j.

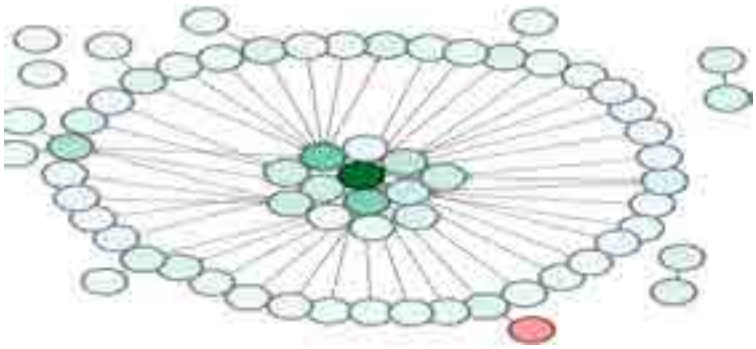


(c) Neighboring nodes: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of neighboring nodes with CC_j.

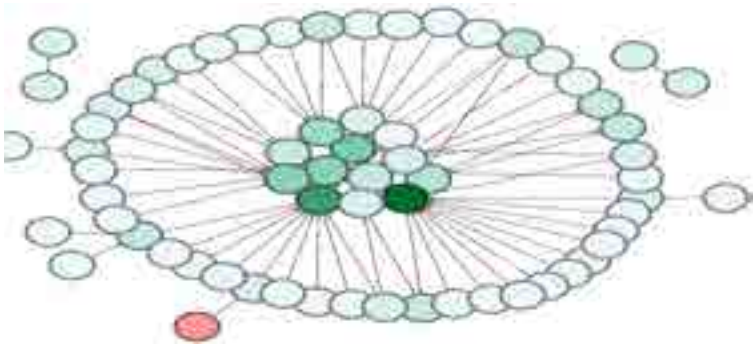
Figure 5.4: (Hyper)Graphs for the core communities (extracted from G_{120}) of the reciprocal network of Google+: snapshot - H_2. The color intensity of a CC is proportional to its degree. The CC highlighted in "red" is the core subgraph yielded by directly applying the standard k-shell decomposition to Google+'s reciprocal network. However, our core communities (hyper)graphs show that this structure in fact does not lie at the very "center" – instead lies more at the outer ring – of the core graph of the Google+ reciprocal network.



(a) Shared nodes: a node represents a CC and an undirected edge $CC_i - CC_j$ denotes that both components share at least one node.



(b) Cross-edges: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of cross edges to nodes in CC_j.



(c) Neighboring nodes: a node represents a CC and a directed edge $CC_i \rightarrow CC_j$ implies that CC_i has the largest number of neighboring nodes with CC_j.

Figure 5.5: (Hyper)Graphs for the core communities (extracted from G_{120}) of the reciprocal network of Google+: snapshot - H_3. The color intensity of a CC is proportional to its degree. The CC highlighted in "red" is the core subgraph yielded by directly applying the standard k-shell decomposition to Google+'s reciprocal network. However, our core communities (hyper)graphs show that this structure in fact does not lie at the very "center" – instead lies more at the outer ring – of the core graph of the Google+ reciprocal network.

Table 5.2: Main statistics of the core communities (hyper)graphs for H_i: c - cliques; CC - connected components

H_i	# c	avg$\|c\|$	# CC	max$\|CC\|$	min$\|CC\|$
1	34,501	23.03	1,758	2,618	1
2	38,055	20.68	2,221	2,487	1
3	65,101	24.96	3,802	6,217	1

edges of sparse subgraphs that are likely to lie at the peripherals of the Google+ reciprocal network. We then performed a form of clique percolation to generate a new *directed* (hyper)graphs where vertices are maximal cliques containing the nodes in the dense "core" graph generated in the previous step, and there exists a directed edge from clique C_i to clique C_j if half of the nodes in C_i are contained in C_j. We found that this (hyper)graph of cliques comprises of 1700+ connected components (CCs), which represent the core "communities" of the Google+ reciprocal network. Finally, we introduced three metrics to study the relations among these CCs in the underlying Google+ reciprocal network: the number of nodes shared by two CCs, the number of nodes that are neighbors in the two CCs, and the number of edges connecting these neighboring nodes. These metrics produce a set of new (hyper)graphs that succinctly summarize the (high-level) structural relations among the core "community" structures and provide a "big picture" view of the core structure of the Google+ reciprocal network and how it is formed. In particular, we found that there are ten CCs that lie at the center of this core structure through which the other CCs are most richly connected.

Our proposed three-step hierarchical procedure assumes that the core subgraph of a network has a large number of cliques. Hence, it may fail to yield a meaningful structure for graphs with just a small number of cliques. To address this limitation, we can relax the notion of clique by constructing substructures which are *clique-like*. For example, a *k-relaxed clique* [114] is a set of nodes that connect to every node in the set except for at most k nodes (a 1-relaxed clique is a clique) [33];*k-clique* is a maximal subgraph such that the distance between each pair of its vertices is not larger than k; k-club [115, 116] is a subgraph with diameter $\leq k$. There are others definitions of relaxed cliques in the literature such as k-plex [115, 116], k-block [115, 116], γ-quasi-clique [115, 116] and $((\alpha, \gamma))$-quasi-clique [115, 116]. As part of ongoing and future work, we will develop

a more rigorous characterization of the core graph of the Google+ reciprocal network based on the (modified) k-shell decomposition, and provide a more in-depth analysis of the (hyper)graph structures of the clique core graph and the (high-level) structural relations among the core "community" structures. We also plan to apply our method to a massive Twitter dataset (with more than 500 million nodes and ≈ 23 billion edges) and other OSNs.

Chapter 6

Conclusion

In this dissertation, we propose new tool to understand the structural properties and formation of complex networks. Our developed schemes are capable of: i) helping to understand possible organizing principles shaping the observed network topology of a directed complex network; ii) extracting the core structure of massive complex networks; and iii) dissecting the structure of the dense nucleus of complex networks.

6.1 Summary of Contributions

Our main contribution in this dissertation are as follows:

1. We present a comprehensive measurement-based characterization of the connectivity among reciprocal edges in a directed complex network – using the online social network (OSN) Google+ as case study – and their evolution over time, with the goal to gain insights into the structural properties of a complex network. In a sense, the reciprocal network can be viewed as the stable skeleton network of a directed network that holds it together. Thus, they could reveal the possible organizing principles shaping the observed network topology of a directed complex network. Moreover, understanding the dynamic structural properties of the reciprocal network provides us with additional information to characterize or compare directed networks that go beyond the classic reciprocity metric, a single static value currently used in many studies.

2. We develop an effective procedure to extract the core structure of complex networks. To achieve this, we introduce a new metric the node "dependence value" that measures the location importance of a node in a network. Then, we define a new measure called "nucleon-index" that captures the extend to which a subgraph is a densely intra-connected and topological central core. Then, using these metrics, we proposed a modified version of the traditional k-shell decomposition method by identifying the k_C-index where we should stop pruning the network in order to preserve its core structure and extract a meaningful "core" for complex networks.

3. We propose a two-step procedure to hierarchically unfold the nucleus of complex networks by building up and generalizing ideas from the existing clique percolation approaches. Using maximal cliques as the basic atomic structures of the network nucleus, we build (hyper)graphs that provide us with a higher-level representation of the dense core graph of complex networks. Hence, our scheme provides a "big picture view of the core structure of a complex network and how it is formed. Our methodology is very scalable and can also be applied to massive complex networks (hundreds million nodes and billion edges).

References

[1] Shai Carmi, Shlomo Havlin, Scott Kirkpatrick, Yuval Shavitt, and Eran Shir. A model of internet topology using k-shell decomposition. *Proceedings of the National Academy of Sciences*, 104(27):11150–11154, 2007.

[2] Romualdo Pastor-Satorras and Alessandro Vespignani. *Evolution and structure of the Internet: A statistical physics approach*. Cambridge University Press, 2007.

[3] Steven H Strogatz. Exploring complex networks. *nature*, 410(6825):268, 2001.

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

[5] Sergey N Dorogovtsev, Alexander V Goltsev, and José Ferreira F Mendes. Pseudofractal scale-free web. *Physical review E*, 65(6):066122, 2002.

[6] MEJ Newman. Mej newman, siam rev. 45, 167 (2003). *SIAM Rev.*, 45:167, 2003.

[7] Duncan J Watts. *Small worlds: the dynamics of networks between order and randomness*. Princeton university press, 1999.

[8] Stefan Bornholdt and Heinz Georg Schuster. *Handbook of graphs and networks: from the genome to the internet*. John Wiley & Sons, 2006.

[9] SN Dorogovtesev and JFF Mendes. Evolution of networks oxford university press. *New York*, 2003.

[10] EO Wilson. Consilience p. 85, 1998.

[11] Braulio Dumba, Golshan Golnari, and Zhi-Li Zhang. Analysis of a reciprocal network using google+: Structural properties and evolution. In *International Conference on Computational Social Networks*, pages 14–26. Springer, 2016.

[12] Braulio Dumba and Zhi-Li Zhang. Uncovering the nucleus of social networks. In *Proceedings of the 10th ACM Conference on Web Science*, pages 37–46. ACM, 2018.

[13] Braulio Dumba and Zhi-Li Zhang. Uncovering the nucleus of a massive reciprocal network. *World Wide Web*, pages 1–26, 2018.

[14] Braulio Dumba and Zhi-Li Zhang. Unfolding the core structure of the reciprocal graph of a massive online social network. In *International Conference on Combinatorial Optimization and Applications*, pages 763–771. Springer, 2016.

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

[16] Neil Zhenqiang Gong and Wenchang Xu. Reciprocal versus parasocial relationships in online social networks. *Social Network Analysis and Mining*, 4(1):184, 2014.

[17] Diego Garlaschelli and Maria I Loffredo. Patterns of link reciprocity in directed networks. *Physical review letters*, 93(26):268701, 2004.

[18] Bo Jiang, Zhi-Li Zhang, and Don Towsley. Reciprocity in social networks with capacity constraints. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 457–466. ACM, 2015.

[19] Phan Nhat Hai and Hyoseop Shin. Effective clustering of dense and concentrated online communities. In *Web Conference (APWEB), 2010 12th International Asia-Pacific*, pages 133–139. IEEE, 2010.

[20] Neil Zhenqiang Gong, Wenchang Xu, Ling Huang, Prateek Mittal, Emil Stefanov, Vyas Sekar, and Dawn Song. Evolution of social-attribute networks: measurements, modeling, and implications using google+. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 131–144. ACM, 2012.

[21] Roberto Gonzalez, Ruben Cuevas, Reza Motamedi, Reza Rejaie, and Angel Cuevas. Google+ or google-?: dissecting the evolution of the new osn in its first year. In *Proceedings of the 22nd international conference on World Wide Web*, pages 483–494. ACM, 2013.

[22] Petter Holme. Core-periphery organization of complex networks. *Physical Review E*, 72(4):046111, 2005.

[23] Puck Rombach, Mason A Porter, James H Fowler, and Peter J Mucha. Core-periphery structure in networks (revisited). *SIAM Review*, 59(3):619–646, 2017.

[24] Peter Csermely, András London, Ling-Yun Wu, and Brian Uzzi. Structure and dynamics of core/periphery networks. *Journal of Complex Networks*, 1(2):93–123, 2013.

[25] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 695–704. ACM, 2008.

[26] Mark EJ Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, 2004.

[27] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[28] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

[29] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 475–486. IEEE, 2006.

[30] Stephen P Borgatti and Martin G Everett. Models of core/periphery structures. *Social networks*, 21(4):375–395, 2000.

[31] Fabio Della Rossa, Fabio Dercole, and Carlo Piccardi. Profiling core-periphery network structure by random walkers. *Scientific reports*, 3:1467, 2013.

[32] Marcio Rosa Da Silva, Hongwu Ma, and An-Ping Zeng. Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks. *Proceedings of the IEEE*, 96(8):1411–1420, 2008.

[33] Georgos Siganos, Sudhir Leslie Tauro, and Michalis Faloutsos. Jellyfish: A conceptual model for the as internet topology. *Journal of Communications and Networks*, 8(3):339–350, 2006.

[34] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42. ACM, 2007.

[35] Murray Shanahan and Mark Wildie. Knotty-centrality: finding the connective core of a complex network. *PLoS One*, 7(5):e36579, 2012.

[36] Liaoruo Wang, John Hopcroft, Jing He, Hongyu Liang, and Supasorn Suwajanakorn. Extracting the core structure of social networks using (α, β)-communities. *Internet Mathematics*, 9(1):58–81, 2013.

[37] Wentao Tang, Davood Babaei Pourkargar, and Prodromos Daoutidis. Relative time-averaged gain array (rtaga) for distributed control-oriented network decomposition. *AIChE Journal*, 64(5):1682–1690, 2018.

[38] Wentao Tang, Andrew Allman, Davood Babaei Pourkargar, and Prodromos Daoutidis. Optimal decomposition for distributed optimization in nonlinear model predictive control through community detection. *Computers & Chemical Engineering*, 2017.

[39] Wentao Tang and Prodromos Daoutidis. Network decomposition for distributed control through community detection in input–output bipartite graphs. *Journal of Process Control*, 64:7–14, 2018.

[40] Sujit Suresh Jogwar and Prodromos Daoutidis. Community-based synthesis of distributed control architectures for integrated process networks. *Chemical Engineering Science*, 172:434–443, 2017.

[41] Gianluca Antonelli. Interconnected dynamic systems: An overview on distributed control. *IEEE Control Systems*, 33(1):76–88, 2013.

[42] John Baillieul and Panos J Antsaklis. Control and communication challenges in networked real-time systems. *Proceedings of the IEEE*, 95(1):9–28, 2007.

[43] Rachana Ashok Gupta and Mo-Yuen Chow. Networked control system: Overview and research trends. *IEEE transactions on industrial electronics*, 57(7):2527–2535, 2010.

[44] Fu Lin, Makan Fardad, and Mihailo R Jovanović. Design of optimal sparse feedback gains via the alternating direction method of multipliers. *IEEE Transactions on Automatic Control*, 58(9):2426–2431, 2013.

[45] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[46] Nader Motee and Ali Jadbabaie. Optimal control of spatially distributed systems. *IEEE Transactions on Automatic Control*, 53(7):1616–1629, 2008.

[47] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

[48] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[49] Yang Tang, Feng Qian, Huijun Gao, and Jürgen Kurths. Synchronization in complex networks and its application–a survey of recent advances and challenges. *Annual Reviews in Control*, 38(2):184–198, 2014.

[50] Hassan Farhangi. The path of the smart grid. *IEEE power and energy magazine*, 8(1), 2010.

[51] Kyoung-Dae Kim and Panganamala R Kumar. Cyber–physical systems: A perspective at the centennial. *Proceedings of the IEEE*, 100(Special Centennial Issue):1287–1308, 2012.

[52] Michael Baldea and Prodromos Daoutidis. *Dynamics and nonlinear control of integrated process systems*. Cambridge University Press, 2012.

[53] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42. ACM, 2007.

[54] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[55] Gabriel Magno, Giovanni Comarela, Diego Saez-Trumper, Meeyoung Cha, and Virgilio Almeida. New kid on the block: Exploring the google+ social graph. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 159–170. ACM, 2012.

[56] Alvin W Wolfe. Social network analysis: Methods and applications. *American Ethnologist*, 24(1):219–220, 1997.

[57] Mohsen Jamali, Gholamreza Haffari, and Martin Ester. Modeling the temporal dynamics of social rating networks using bidirectional effects of social relations and rating patterns. In *Proceedings of the 20th international conference on World wide web*, pages 527–536. ACM, 2011.

[58] Yanhua Li, Zhi-Li Zhang, and Jie Bao. Mutual or unrequited love: Identifying stable clusters in social networks with uni-and bi-directional links. In *WAW*, volume 12, pages 113–125. Springer, 2012.

[59] Google+ Platform. `http://www.google.com/intl/en/+/learnmore/`.

[60] Google+. `http://en.wikipedia.org/wiki/Google+`.

[61] Google+ New Feature. `http://googledrive.blogspot.com/2012/10/share-your-stuff-from-google-drive-to.html`.

[62] Google Strips Down Google Plus. `http://blogs.wsj.com/digits/2015/11/17/google-strips-down-google-plus/`.

[63] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[64] Martin Rosvall and Carl T Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences*, 104(18):7327–7331, 2007.

[65] Spiros Papadimitriou, Jimeng Sun, Christos Faloutsos, and S Yu Philip. Hierarchical, parameter-free community discovery. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 170–187. Springer, 2008.

[66] Jing Gao, Feng Liang, Wei Fan, Chi Wang, Yizhou Sun, and Jiawei Han. On community outliers and their efficient detection in information networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 813–822. ACM, 2010.

[67] Tianbao Yang, Rong Jin, Yun Chi, and Shenghuo Zhu. Combining link and content for community detection: a discriminative approach. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 927–936. ACM, 2009.

[68] M Puck Rombach, Mason A Porter, James H Fowler, and Peter J Mucha. Core-periphery structure in networks. *SIAM Journal on Applied mathematics*, 74(1):167–190, 2014.

[69] Jaewon Yang and Jure Leskovec. Structure and overlaps of communities in networks. *arXiv preprint arXiv:1205.6228*, 2012.

[70] Patrick Doreian. Structural equivalence in a psychology journal network. *Journal of the American Society for Information Science*, 36(6):411–417, 1985.

[71] Antonios Garas, Frank Schweitzer, and Shlomo Havlin. A k-shell decomposition method for weighted networks. *New Journal of Physics*, 14(8):083030, 2012.

[72] Bo Wei, Jie Liu, Daijun Wei, Cai Gao, and Yong Deng. Weighted k-shell decomposition for complex networks based on potential edge weights. *Physica A: Statistical Mechanics and its Applications*, 420:277–283, 2015.

[73] Daniele Miorandi and Francesco De Pellegrini. K-shell decomposition for dynamic complex networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*, pages 488–496. IEEE, 2010.

[74] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. Identification of influential spreaders in complex networks. *Nature physics*, 6(11):888–893, 2010.

[75] Antonios Garas, Panos Argyrakis, Céline Rozenblat, Marco Tomassini, and Shlomo Havlin. Worldwide spreading of economic crisis. *New journal of Physics*, 12(11):113043, 2010.

[76] Diego F Rueda, Eusebi Calle, and Jose L Marzo. Robustness comparison of 15 real telecommunication networks: Structural and centrality measurements. *Journal of Network and Systems Management*, 25(2):269–289, 2017.

[77] Facebook friendships network dataset – KONECT, September 2016.

[78] Jrme Kunegis. KONECT – The Koblenz Network Collection. In *Proc. Int. Conf. on World Wide Web Companion*, pages 1343–1350, 2013.

[79] Stanford Large Network Dataset Collection. `https://snap.stanford.edu/data/`.

[80] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowledge Discovery from Data*, 1(1):1–40, 2007.

[81] Route views network dataset – KONECT, September 2016.

[82] Caida network dataset – KONECT, September 2016.

[83] Internet topology network dataset – KONECT, September 2016.

[84] Beichuan Zhang, Raymond Liu, Daniel Massey, and Lixia Zhang. Collecting the Internet AS-level topology. *SIGCOMM Computer Communication Review*, 35(1):53–61, 2005.

[85] Euroroad network dataset – KONECT, September 2016.

[86] Lovro Šubelj and Marko Bajec. Robust network community detection using balanced propagation. *Eur. Phys. J. B*, 81(3):353–362, 2011.

[87] Us airports network dataset – KONECT, September 2016.

[88] Tore Opsahl. Why anchorage is not (that) important: Binary ties and sample selection, 2011.

[89] Openflights network dataset – KONECT, September 2016.

[90] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interaction in Facebook. In *Proc. Workshop on Online Social Networks*, pages 37–42, 2009.

[91] Jazz musicians network dataset – KONECT. `http://konect.uni-koblenz.de/networks/arenas-jazz`, 2016.

[92] Pablo M. Gleiser and Leon Danon. Community structure in jazz. *Advances in Complex Systems*, 6(4):565–573, 2003.

[93] Marin Bogu, Romualdo Pastor-Satorras, Albert Daz-Guilera, and Alex Arenas. Models of social networks based on social distance attachment. *Phys. Rev. E*, 70(5):056122, 2004.

[94] Pretty Good Privacy network dataset – KONECT.

[95] DNC emails co-recipients network dataset – KONECT. `http://konect.uni-koblenz.de/networks/dnc-corecipient`, 2016.

[96] Infectious network dataset – KONECT, September 2016.

[97] Lorenzo Isella, Juliette Stehl, Alain Barrat, Ciro Cattuto, Jean-Franois Pinton, and Wouter Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *J. of Theoretical Biology*, 271(1):166–180, 2011.

[98] Train bombing network dataset – KONECT, September 2016.

[99] Brian Hayes. Connecting the dots. can the tools of graph theory and social-network studies unravel the next big plot? *American Scientist*, 94(5):400–404, 2006.

[100] Roberto Gonzalez, Ruben Cuevas, Reza Motamedi, Reza Rejaie, and Angel Cuevas. Google+ or google-?: dissecting the evolution of the new osn in its first year. In *Proceedings of the 22nd international conference on World Wide Web*, pages 483–494. ACM, 2013.

[101] José Ignacio Alvarez-Hamelin, Luca Dall'Asta, Alain Barrat, and Alessandro Vespignani. K-core decomposition of internet graphs: hierarchies, self-similarity and measurement biases. *arXiv preprint cs/0511007*, 2005.

[102] J Ignacio Alvarez-Hamelin, Luca Dall'Asta, Alain Barrat, and Alessandro Vespignani. Large scale networks fingerprinting and visualization using the k-core decomposition. In *Advances in neural information processing systems*, pages 41–50, 2006.

[103] Lei Tang and Huan Liu. Community detection and mining in social media. *Synthesis lectures on data mining and knowledge discovery*, 2(1):1–137, 2010.

[104] Gert Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

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

[106] Phillip Bonacich. Power and centrality: A family of measures. *American journal of sociology*, 92(5):1170–1182, 1987.

[107] Mark Newman. *Networks: an introduction*. Oxford university press, 2010.

[108] Shi Zhou and Raúl J Mondragón. The rich-club phenomenon in the internet topology. *IEEE Communications Letters*, 8(3):180–182, 2004.

[109] Julian J McAuley, Luciano da Fontoura Costa, and Tibério S Caetano. Rich-club phenomenon across complex network hierarchies. *Applied Physics Letters*, 91(8):084103, 2007.

[110] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[111] Vladimir Batagelj and Matjaž Zaveršnik. Generalized cores. *arXiv preprint cs/0202039*, 2002.

[112] Maksym Gabielkov, Ashwin Rao, and Arnaud Legout. Studying social networks at scale: macroscopic anatomy of the twitter social graph. In *ACM SIGMETRICS Performance Evaluation Review*, volume 42, pages 277–288. ACM, 2014.

[113] Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. *Icwsm*, 8:361–362, 2009.

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

[115] Jeffrey Pattillo, Nataly Youssef, and Sergiy Butenko. On clique relaxation models in network analysis. *European Journal of Operational Research*, 226(1):9–18, 2013.

[116] Timo Gschwind, Stefan Irnich, Fabio Furini, Roberto Wolfler Calvo, et al. Social network analysis and community detection by decomposing a graph into relaxed cliques. Technical report, 2015.

Appendix A

Beta Parameter Selection

Beta Parameter Selection: we now proof that the number of n-step removed neighbors of i is multiplied by β^{n-1}. We also present a discussion on how the selection of values for the β parameter in (4.1) impacts our criteria to stop the k-shell decomposition method presented in Sect. 4.4.1:

Given that $dep^0(i) = 0$ and $dep^1(i) = \delta^1(i)$, we can write an expression for $dep^2(i)$ as following:

$$\begin{aligned} dep^2(i) &= dep^1(i) + \delta^2(i) + \beta \times \Sigma_{j \in N^2(i)} dep^1(j) \\ &= \delta^1(i) + \delta^2(i) + \beta \times \Sigma_{j \in N^2(i)} \delta^1(j) \end{aligned} \tag{A.1}$$

Let's assume that node i has $c(i) = 4$, then $dep^4(i)$ is computed as following:

$$dep^4(i) = dep^3(i) + \delta^4(i) + \beta \Sigma_{j \in N^4(i)} [dep^3(j)] \tag{A.2}$$

Expanding (A.2) gives:

$$dep^4(i) = dep^3(i) + \delta^4(i) + \beta \Sigma_{j \in N^4(i)} [dep^2(j) + \delta^3(j) + \beta \Sigma_{j' \in N^3(j)} dep^2(j')]$$

Substituting (A.1) gives:

$$dep^4(i) := dep^3(i) + \delta^4(i) + \beta\Sigma_j[M^3(j) + \beta\delta^2(j)\rho^1(j'^*) + \beta\Sigma_{j'}[M^2(j') + \beta\delta^2(j')\rho^1(j'')]]$$

where $M^k(i) = \Sigma_k\delta^k(i)$ and $\delta^k(i) = \rho^k(i), \forall i \in V$.

Further simplify $dep^4(i)$ gives:

$$dep^4(i) := dep^3(i) + \delta^4(i) + \Sigma_j[\beta M^3(j) + \beta^2\delta^2(j)\rho^1(j'^*) + \Sigma_{j'}[\beta^2 M^2(j') + \beta^3\delta^2(j')\rho^1(j'')]]$$

We can rewrite the above expressions as:

$$dep^4(i) := dep^3(i) + \beta^0 A + \Sigma_j[\beta B + \beta^2 C + \Sigma_{j'}[\beta^2 D + \beta^3 E]] \tag{A.3}$$

Where:

- $A = \delta^4(i)$: 1-step neighbors of i removed at $k = 4$
- $B = M^3(j)$: 2-step neighbors of i removed at $k = 1, 2, 3$
- $C = \delta^2(j)\rho^1(j'^*)$: 3-step neighbors of i removed at $k = 1$
- $D = M^2(j')$: 3-step neighbors of i removed at $k = 1, 2$
- $E = \delta^2(j')\rho^1(j'')$: 4-step neighbors of i removed at $k = 1$

By generalizing equation (A.3) ($k = 5, ..., n$), we observe that at every k-index, the number of n-step removed neighbors of i is multiplied by β^{n-1}. This concludes our proof. Essentially, the parameter β quantifies the contribution of node j to the total dependence value of node i. Thus, varying β in the range $]0, 1[$ will not have any impact on the value of the k-index where we should stop the k-shell decomposition method — by varying β, we are impacting the contribution of any node j to the total dependence value of node i by the same proportion. Thus varying the β^{n-1} does not have any impact in our criteria to stop the k-shell decomposition method introduced in Sect. 4.4.1.

Appendix B

Parameter S: steepness of the curve

The parameter S controls the steepness of the curve in eq.(4.5) (see Fig. B.1). For our dataset, we obtained the best results with $S = 50$ (i.e., H_1 and H_2) and $S = 110$ (H_3).

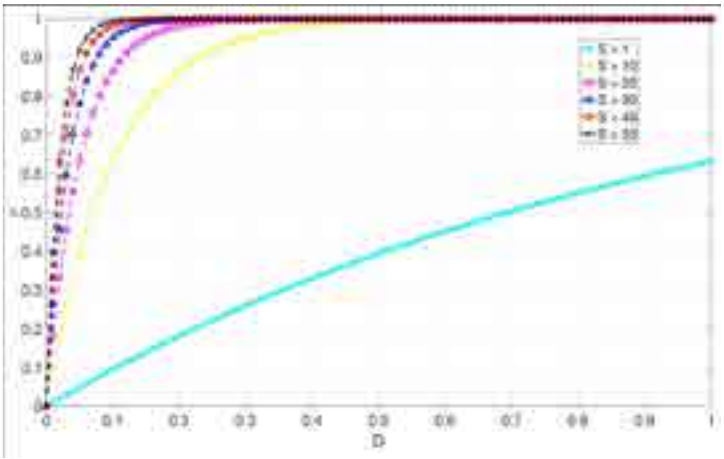


Figure B.1: ρ curve for several values of the parameter S.

Appendix C

Publications

In addition to this dissertation, the presented results are also documented in the following published papers.

- **Braulio Dumba**, Zhi-Li Zhang. "Uncovering the Nucleus of Social Networks". Proceedings of the 10th ACM Conference on Web Science (WebSci'18), May 27-30, 2018, Amsterdam, Netherlands.
- **Braulio Dumba**, Zhi-Li Zhang, "Uncovering the Nucleus of a Massive Reciprocal Network", World Wide Web Journal - Special issue on Social Computing and Big Data Applications, (2018): doi.org/10.1007/s11280-018-0609-7.
- **Braulio Dumba,** Zhi-Li Zhang, "Unfolding the Core Structure of the Reciprocal Graph of a Massive Online Social Network.", Proceedings of the 10th Annual International Conference on Combinatorial Optimization and Applications (COCOA'16), Hong Kong, China, December 16-18, 2016.
- **Braulio Dumba**, Golshan Golnari, Zhi-Li Zhang, "Analysis of a Reciprocal Network Using Google+: Structural Properties and Evolution.", Proceedings of the 5th International Conference on Computational Social Networks (CSoNet'16), Ho Chi Minh City, Vietnam, August 2-4, 2016.
- Eman Ramadan, Hesham Mekky, **Braulio Dumba**, Zhi-Li Zhang, "Adaptive Resilient Routing via Preorders in SDN", Proceedings of the 4th Workshop on Distributed Cloud Computing (DCC'16), Chicago, IL, July 25, 2016.

- **Braulio Dumba**, Hesham Mekky, Sourabh Jain, Guobao Sun, Zhi-Li Zhang, "A Virtual Id Routing Protocol for Future Dynamics Networks and Its Implementation Using the SDN Paradigm.", Journal of Network and Systems Management, 24(3), 578-606. doi:10.1007/s10922-016-9373-0.

- **Braulio Dumba**, Hesham Mekky, Guobao Sun, Zhi-Li Zhang, "In-Network Dynamic Pathlet Switching with VIRO for SDN Networks", International Workshop on Computer and Networking Experimental Research using Testbeds (CNERT'15), co-located with IEEE ICDCS'15, Columbus, Ohio June 19, 2015.

- **Braulio Dumba**, Guobao Sun, Hesham Mekky, Zhi-Li Zhang, "Experience in Implementing & Deploying a Non-IP Routing Protocol VIRO in GENI." International Workshop on Computer and Networking Experimental Research using Testbeds (CNERT'14), co-located with IEEE ICNP'14 , The Research Triangle, NC, Oct 24, 2014. (Best Paper)

- **Braulio Dumba**, Guobao Sun, Hesham Mekky, Zhi-Li Zhang, "Poster: VIRO-GENI: Deployment of a plug & play, scalable, robust virtual Id routing in GENI.", The 20th GENI Engineering Conference (GEC20) , Davis, CA, June 21-24, 2014.