CHARACTERIZING AND DETECTING COHESIVE SUBGROUPS WITH APPLICATIONS TO SOCIAL AND BRAIN NETWORKS

A Dissertation

by

MAKBULE ZEYNEP ERTEM OKTAY

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Sergiy Butenko
Committee Members,	Amarnath Banerjee
	Erick Moreno-Centeno
	Mark Lenox
Head of Department,	Cesar O. Malave

August 2015

Major Subject: Industrial Engineering

ABSTRACT

Many complex systems involve entities that interact with each other through various relationships (e.g., people in social systems, neurons in the brain). These entities and interactions are commonly represented using graphs due to several advantages. This dissertation focuses on developing theory and algorithms for novel methods in graph theory and optimization, and their applications to social and brain networks.

Specifically, the major contributions of this dissertation are three fold. First, this dissertation aims not only to develop a new clique relaxation model based on a structural metric, *clustering coefficient*, but also to introduce a novel graph clustering algorithm using this model. *Clique relaxations* are used in classical models of cohesive subgroups in social network analysis. *Clustering coefficient* was introduced more recently as a structural feature characterizing small-world networks. Leveraging the similarities between the concepts of cohesive subgroups and small-world networks (i.e., graphs that are highly clustered with small path lengths). The first part of this dissertation introduces a new clique relaxation, α-*cluster*, defined by enforcing a lower bound α on the clustering coefficient in the corresponding induced subgraph. Two different definitions of the clustering coefficient are considered, namely, the local and global clustering coefficient. Certain structural properties of α-clusters are analyzed, and mathematical optimization models for determining the largest size α-clusters in a network are developed and applied to several real-life social network instances. In addition, a network clustering algorithm based on local α-cluster is introduced and successfully evaluated.

Second, this dissertation explores a novel mathematical model called the *maximum independent union of cliques problem (max IUC problem)*, which arises as a

special case of α-clusters. It is an interesting problem for which both the maximum clique and maximum independent sets are feasible solutions and individually their corresponding sizes are lower bounds for the size of the IUC solution. After presenting the structural properties as well as the complexity results of different graph types (planar, unit disk graphs and claw-free graphs), an integer programming formulation is developed, followed by a branch-and-bound algorithm and several heuristic methods to approximate the *maximum independent union of cliques problem*.The developed methods have been empirically evaluated on many benchmark instances.

Finally, this dissertation, in collaboration with Texas Institute of Preclinical Studies (TIPS), applies clique relaxation models to explore a new experimental data to understand the effect of concussion on animal brains. Our research involves cohesive and robust clustering analysis of animal brain networks utilizing a unique and novel experimental data. In collaboration with TIPS, we have analyzed multiple pairs of fMRI data about animal brains that are measured before and after a concussion. We utilize network analysis to first identify the similar regions in animal brains, and then compare how these regions as well as graph structural properties change before and after a concussion. To the best of our knowledge, this study is unique in the literature in that it not only explicitly examines the relation between concussion level and the functional unit interaction but also uses very detailed and fine-grained fMRI measurements of brain data.

Dedicated to my beloved mother.

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the help of so many people in so many ways. I would like to express the deepest appreciation to my committee chair and my advisor Dr. Sergiy Butenko, without his guidance and persistent help this dissertation would not have been possible.

I would like to thank my committee members Dr. Amarnath Banerjee, Dr. Erick Moreno-Centeno and Dr. Mark Lenox for taking keen interest in my research and my professional development as well as their time.

I am ever grateful to my collaborator Dr. Mark Lenox for introducing me to brain networks. I would also thank my other collaborators Dr. Alexander Veremyev, Dr. Yiming Wang, Dr. Shahram Shahinpour, Dr. Anurag Verma who have been great colleagues to work with.

I would especially like to thank Drs. Wilbert Wilhem, Kiavash Kianfar, Georgia-Ann Klutke and Yu Ding for providing me with several opportunities to teach.

I would like to thank the staff members in the department for the learning opportunities they availed to me and the graduate student community in the department for their true friendship and support.

My special thanks goes to the love of my life, my husband for his unconditional love, patience and constant support. And thanks to my little sunshine, Serra for bringing joy and happiness to my life. Even a little sparkle in her eyes is more than enough to enlighten all the darkness in my day.

Words cannot express my thanks to my family whose deepest love, encouragement and continuous support made me who I am. I am ever grateful to my mom, my dad and my brother Sinan, for always being there for me on every journey. I would like

to thank my mom especially from the bottom of my heart for endless encouragement to pursue my dreams and for all the sacrifices she made.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Many complex systems may involve entities that interact with each other. For example, social systems include individuals who may interact with each other through various relationships such as friendship and co-occurrence. Another example can be a biological system such as a living brain that contains functional units that interact with each other to perform high-level functions such as hearing and walking.

Many interesting properties of these systems might require to identify closely-knit subgroups of entities within these systems. For example, in social systems, identifying interacting group of people (i.e., cohesive subgroup) might reveal interesting social circles. In biological systems, cohesive subgroups might identify anatomical parts that are responsible for specific functions.

These systems can be simply and effectively represented using graphs. Their entities can be represented as nodes, and pairwise interactions between entities can be represented as edges. One of the advantages of such representation is that graph theory, combinatorial optimization, and algorithms theory can be utilized to effectively answer interesting questions related to these systems. Focus of this dissertation is to model them as graphs and develop mathematical models as well as algorithms to obtain cohesive subgroups within these systems.

In the following sections, we first introduce a new mathematical model that corresponds to a new definition for cohesive subgroups based on a commonly used graph metric, and we develop a network clustering algorithm using this new model. Second, we develop exact and approximate algorithms for a special case of our first model, for which two classical canonical problems (i.e., maximum independent set and maximum clique) are lower bounds. Finally, this dissertation explores a unique

and novel experimental data set about animal brains. Using graph mining tools, we examine the effect of concussion on animal brains.

1.1 Clique Relaxations and Clustering Coefficients of Networks

A cohesive subgroup is a "tightly knit" subset of actors in a social network, which was originally modeled using the graph-theoretic concept of a clique [42]. The notion of a clique embodies a perfect cohesive group, it compels every two individuals in the subgroup to be directly connected to each other. However, this can often be impractical in real-life social networks. Indeed, requiring every possible pairwise connection between the individuals in a subgroup is often implausible and stringent. In addition, clique-detection algorithms may fail to identify cliques in which a few edges are absent due to imprecisions in collecting the data. To overcome these issues, clique relaxation models have been introduced, including the k-clique [41], relaxing direct interaction between individuals; the k-club [3, 44], relaxing reachability; the k-plex [53], allowing at most k non-neighbors; and s-defective clique [65], allowing at most s missing edges. Clique relaxation models have been extensively used in social network analysis [50, 45, 52, 61].

Section 3 proposes a novel clique relaxation based on the notion of *clustering coefficient*. This concept gained popularity in the study of the so-called *small-world networks* [62, 63], where it is used to model an observation that two people are more likely to be friends if they have a friend in common. For a given actor (node) with more than one friend (neighbor), its local clustering coefficient measures the probability that its two randomly picked friends are also friends with each other. Clustering coefficient is equal to one when the node's neighborhood is fully directly connected (forms a clique). On the other hand, a close to zero clustering coefficient means that there are hardly any connections in the neighborhood. Many real-life

networks have been empirically found to have nodes with rather high clustering coefficients [46], which also appears to be a natural property to expect of cohesive subgroups in social networks. In fact, according to [35, p. 35], clustering coefficient is "the most common way of measuring some aspect of cliquishness".

Hence, it is reasonable to define a cohesive subgroup by requiring that the corresponding subset of nodes induces a connected subgraph with a desired (high) clustering coefficient α. We will refer to such a structure as an α-cluster. If $\alpha = 1$, the connectivity requirement ensures that an α-cluster is a clique. Otherwise, if $\alpha < 1$, an α-cluster can be viewed as a clique relaxation.

Our study focuses on computing α-clusters of the *largest size* in networks, which is of interest for several reasons. Larger cohesive subgroups tend to have more influence on the overall network structure than their smaller counterparts. In fact, the largest size of a cohesive subgroup of a certain kind can be thought of as a *global measure of cohesiveness* of the whole network with respect to the imposed definition of cohesiveness. The presence of large cohesive subgroups consisting of considerable portions of a network implies a high level of cohesion in the network, whereas their absence indicates the opposite. Nevertheless, smaller cohesive subgroups may also be of interest, and the approaches proposed in this work can be easily modified to compute all α-clusters of a given size by introducing the corresponding constraints in the considered optimization models.

Dropping the connectivity requirement from the α-cluster definition results in a structure whose connected components are α-clusters. This motivates a novel clustering algorithm, which uses the multiple α-clusters as the "seeds", with the remaining nodes assigned to these seed clusters using a certain strategy. The algorithm yields encouraging results on the set of social network instances used in our experiments.

1.2 Maximum Clique and Maximum Independent Set on Networks

The maximum clique and maximum independent set problems are the two classical problems in combinatorial optimization [16, 2, 8, 1, 33]. Maximum clique problem is one of the Karp's 21 problems that are shown to be $\mathcal{NP}$-complete [37] and both problems remain hard to approximate [51, 6, 28, 67]. Solutions of these two problems are closely related, indeed, a set of vertices C is a clique if and only if C is an independent set in the complementary graph $\overline{G} = (V, \overline{E})$. Therefore, many computational approaches to one problem may be directly applied to the other problem. Although, these problems are not polynomial time solvable in general case (unless P=NP), in some special graphs like perfect graphs, maximum clique problem is shown to be polynomially solvable [7, 9]. Pardalos and Xue [49] give a summary of the special classes of graphs where the maximum clique and independent set problems have been studied.

We study another related problem to these classical canonical problems that could be stated as independent union of cliques in which cliques are independent. The *independent union of cliques* (*IUC* for short) model in a way combines maximum clique and maximum independent set problems. Given a graph G, an *IUC* is a subset of vertices inducing a subgraph with each connected component forming a clique. We can define the *maximum* IUC *problem* as finding the maximum set of vertices that forms an *IUC*. For the proposed problem, maximum clique and maximum independent set solutions are both feasible, hence they both provide lower bounds for the maximum IUC problem.

Definition 1 (Independent union of cliques (IUC for short))**.** *Given a graph G, an* IUC *is a subgraph such that each connected component in the subgraph forms a clique.*

Definition 2 (Maximum *IUC*)**.** *The maximum* IUC *problem is to find the maximum*

subset of vertices $C \subseteq V$ such that induced $G[C]$ is an IUC.

This problem is a special case of two general problems recently introduced in the literature. First, it is a variation of the *maximum α-cluster problem* [25]. Specifically, when $\alpha = 1$ and no connectivity constraints are imposed, a *maximum* α-cluster solution induces a set of cliques where the total cardinality is maximized, which is also the optimal solution for IUC. Similarly, this problem is a variation of the *s-plex cluster vertex deletion problem* when s is equal to 1 [15]. In this dissertation, we develop exact and efficient approximate algorithms for this problem.

Jansen et. al., [36] introduce the *maximum disjoint union of clique problem (max DUC)*, in which for a given D, max DUC finds D disjoint cliques (a set of vertices and a set of edges) where the total number of nodes in the solution is maximized. They also provide polynomial time algorithms for special graphs like interval graphs, bipartite graphs and directed path graphs. A similar study was conducted recently by Ames and Vavasis [4]. These closely related papers introduce the union of cliques in which the cliques in the solution may contain an edge between them in the original graph. However in *max IUC*, cliques are independent from each other in the subgraph induced by the solution.

1.3 Network Analysis of Large Scale Brain Networks

Neuroimaging data for brains have been used in many studies to understand complex structure of a brain and its connectivity patterns. Neuroimaging data is usually represented in two distinct ways in the literature. The first set of studies represents this data as a list of temporal vectors and use Euclidean distance to determine the similarity between them [19, 39, 13, 12]. The second set of studies represents this data as a graph by using temporal correlation to determine the relationships between voxels. Recent developments in this direction include a large body of theo-

retical and experimental research on the healthy and diseased brain networks such as brains affected by epilepsy, schizophrenia, Parkinson's disease, and traumatic brain injuries [5, 14, 55, 18, 27]. A detailed analysis on graph theoretical modeling of brain connectivity can be found in the survey paper [34].

There has been extensive empirical evidence in the literature that brain networks are small-world networks [62, 63]. They have the properties like high clustering coefficient and low length of the shortest paths between pairs of nodes. Our research involves extensive graph theoretic analysis on animal brain networks created from a unique and novel experimental data to explore the effect of concussion. To the best of our knowledge, this study is unique in that it explicitly examines the relation between concussion level and structural properties of animal brain networks. Specifically, we perform two types of analysis. First, we analyze the changes in the structural properties of these graphs before and after the concussion (referred to as treatment). For this analysis, we focus on the comparison of basic graph structural properties such as edge density, degree distribution and clustering coefficient. Second, we identify the change in cohesive subgroups in these graphs before and after the treatment.

2. DETECTING LARGE COHESIVE SUBGROUPS WITH A HIGH CLUSTERING COEFFICIENT

This section focuses on novel clique relaxation models based on clustering coefficient. We utilized local clustering coefficient and global clustering coefficient metrics. Since clustering coefficients are commonly used to asses small-world properties of networks, imposing a high lower bound α on the clustering coefficient (local or global) within a cluster ensures that the corresponding subnetwork has strong small-world properties. We develop mathematical models in order to find the largest α-clusters. Furthermore, we introduce a novel clustering method based on local α-clusters.

The remainder of this section is organized as follows. In the next section, we introduce the necessary definitions and study some basic structural properties of α-clusters. Section 2.2 provides optimization models for finding the largest α-clusters in a network. We present the cubic and the quadratic models in Section 2.2.1 and the triangular model in Section 2.2.4. In Section 2.3, the proposed models are applied to analyze several well-known social network instances. The proposed local α-clustering algorithm is described and evaluated in Section 2.4. This section is based on the submitted paper by Ertem et al. [25].

2.1 Definitions and Properties

This section presents basic graph-theoretic definitions and notations used throughout this section. Let $G = (V, E)$ be a simple graph with set V of n vertices (nodes) and set E of edges (links), $E \subset \{\{i, j\} : i, j \in V\}$. If $\{i, j\} \in E$, we say that vertices i and j are *adjacent* to each other; we also call i and j *neighbors*. We can represent G using its *adjacency matrix* A_G, whose elements $a_{ij}, i, j = 1, \ldots, n$ indicate whether

or not there is an edge between nodes i and j:

$$a_{ij} = \begin{cases} 1, & \{i,j\} \in E, \\ 0, & \text{otherwise.} \end{cases}$$

Let $N_G(i) = \{j : \{i,j\} \in E\}$ be the *neighborhood* of i in G. The *degree* $d_G(i)$ of vertex i is the number of neighbors of i in G. The edge density $\rho(G)$ of G is the ratio of the number of edges in G over the possible number of edges in the graph, that is $\rho(G) = |E|/\binom{n}{2}$.

A *path* between vertices i and j in G is a subgraph of G defined by an alternating sequence of distinct vertices and edges, with the first and the last elements of the sequence given by i and j, respectively, and with the edges defined by pairs of consecutive vertices in the sequence. The length of a path is given by the number of edges in the corresponding sequence. Two vertices i and j are *connected* in G if there exists a path between i and j in G. A graph is *connected* if all its vertices are pairwise connected; the graph is *disconnected* otherwise. The *distance* $d_G(i,j)$ between vertices i and j in G is the shortest path length between i and j in G; $d_G(i,j) = \infty$ if i and j are disconnected. The diameter of G, denoted as $diam(G)$, is the maximum distance between a pair of nodes in V. Given a subset $V' \subset V$, the corresponding induced subgraph $G[V']$ is defined as $G[V'] = (V', E')$, where E' is the subset of edges of G connecting pairs of vertices from V'.

Watts and Strogatz [63] define the *local clustering coefficient* for a node of degree at least 2 as the proportion of links between the vertices within its neighborhood divided by the number of links that could possibly exist between the nodes in the neighborhood.

Definition 3 (Local clustering coefficient)**.** *The local clustering coefficient C_i of node*

i of degree $d_G(i) \geq 2$ *in* G *is given by*

$$C_i = \frac{\sum\limits_{j,k \in N_G(i), j<k} a_{jk}}{\binom{d_G(i)}{2}}. \tag{2.1}$$

The global clustering coefficient $\mathcal{C}$ of G can be thought of as the probability that two randomly chosen neighbors of an arbitrary node of degree at least 2 are adjacent to each other. It can be expressed mathematically as follows.

Definition 4 (Global clustering coefficient)**.** *The global clustering coefficient* $\mathcal{C}$ *of graph* G *that has at least one connected component with more than 2 vertices is given by*

$$\mathcal{C} = \frac{\sum\limits_{i \in V} \sum\limits_{j,k \in N_G(i), j<k} a_{jk}}{\sum\limits_{i \in V} \binom{d_G(i)}{2}}. \tag{2.2}$$

Relatively unexpectedly, both local and global clustering coefficients of a graph can decrease with an increase in edge density. Indeed, Figure 2.1 shows a graph where adding an edge between nodes 3 and 5 decreases the clustering coefficients. Before the dashed edge is added, local clustering coefficients of the corresponding nodes are $\{1, \frac{1}{3}, 1, \frac{1}{3}, 1, 1\}$, and the global clustering coefficient is $\mathcal{C} = \frac{6}{10}$. After the addition, local clustering coefficients change to $\{1, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 1\}$, and the global clustering coefficient value is $\frac{6}{14}$.

We define a *local* α*-cluster* as a subset of vertices that induces a subgraph in which each node's local clustering coefficient is at least α.

Definition 5 (Local α-cluster)**.** *Given a graph* $G = (V, E)$*, a subset of vertices* $C \subseteq V$ *is called a local* α*-cluster if* $G[C]$ *is connected and every node in* C *has the*

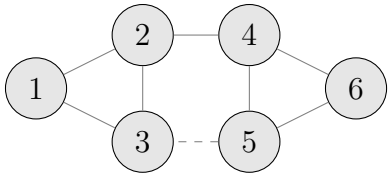


Figure 2.1: Addition of an edge decreases global clustering coefficient.

local clustering coefficient at least α in $G[C]$, that is,

$$\sum_{j,k \in N_{G[C]}(i)} a_{jk} \geq \alpha \binom{d_{G[C]}(i)}{2} \quad \forall i \in C. \tag{2.3}$$

Note that the definition of local clustering coefficient implies that for an α-cluster C the degree of each node in $G[C]$ is at least 2. Also, it is easy to see that the edge density of the subgraph induced by any local α-cluster is at least α. However, the set of vertices inducing a subgraph with the edge density α may not be a local α-cluster.

Similarly, we can define a *global α-cluster* as follows.

Definition 6 (Global α-cluster)**.** *Given a graph $G = (V, E)$, a subset of vertices $C \subseteq V$ is called a global α-cluster if $G[C]$ is connected and $G[C]$ has the global clustering coefficient at least α, that is,*

$$\sum_{i \in C} \sum_{j,k \in N_{G[C]}(i)} a_{jk} \geq \alpha \sum_{i \in C} \binom{d_{G[C]}(i)}{2}. \tag{2.4}$$

It is easy to see that the edge density of the subgraph induced by a global α-cluster can be less than α. For example, the graph in Figure 2.1 (before the edge {3,5} is added) has the global clustering coefficient of 6/10 while its edge density is

7/15. Also, It is obvious that a local α-cluster is also a global α-cluster, whereas the converse does not hold in general. Hence, the definition of a local α-cluster guarantees stronger cohesiveness properties than those enforced by the definition of a global α-cluster. This is also evident from the experiments with real-life networks reported in Section 2.3, which led us to focus primarily on local α-clusters in this study.

2.2 Mathematical Models

Next we formulate optimization models for detecting the largest local and global α-clusters in a given graph G. Our objective is to find a maximum size subset of nodes with the local (global) clustering coefficient above the given threshold value α. We start by describing formulations for finding largest local α-clusters, which can then be altered to address the maximum global α-cluster problem as discussed in Section 2.2.5.

Decision variables: Let us define x_i as a binary decision variable indicating whether node i is in the set C sought, that is,

$$x_i = \begin{cases} 1, & \text{if node } i \text{ is in the local (global) } \alpha\text{-cluster } C; \\ 0, & \text{otherwise.} \end{cases}$$

The vector x consisting of x_i defined as above is called the *characteristic vector* of the subset C of vertices.

Objective function: Our objective is to maximize the number of vertices in the α-cluster C:

$$\text{maximize} \sum_{i \in V} x_i. \tag{2.5}$$

We consider three different approaches to modeling the *local α-cluster constraints*,

which ensure that the set of nodes in C satisfies the local α-cluster definition. The first two approaches, referred to as the cubic and quadratic model, respectively, are based on alternative straightforward representations of Equation (2.3) in terms of the decision variables (Section 2.2.1), whereas the third model, called the triangle model, is based on somewhat more sophisticated arguments (Section 2.2.4). As we will observe later, the triangle model outperforms the other two models in terms of running times when used in conjunction with an optimization solver.

2.2.1 The Cubic and Quadratic Models

To ensure that the subset of vertices C induces a local α-cluster, we impose three sets of constraints. The first set of constraints formalizes the requirement that each node must have a degree value at least 2 in $G[C]$. This requirement can be expressed as follows:

$$\sum_{j \in V} a_{ij} x_j \geq 2x_i \quad \forall i \in V. \tag{2.6}$$

If $x_i = 1$ (meaning that $i \in C$), this constraint guarantees that at least 2 neighbors of i are included in C. We call the constraints in (2.6) the *degree constraints*.

The second set of constraints expresses equation (2.3) from the definition of a local α-cluster in terms of the decision variables. Namely, for each node i in the α-cluster, we need to make sure that

$$\sum_{j,k \in N_G(i), j<k} a_{ij} a_{ik} a_{kj} x_j x_k \geq \alpha \binom{\sum_{k \in V} a_{ik} x_k}{2} \quad \forall i \in C. \tag{2.7}$$

These constraints are called the *local clustering coefficient constraints*. Since C is unknown, we need to rewrite equation (2.7) to avoid using C explicitly. We consider two alternative representations for equation (2.7), one (cubic) obtained by multiplying both sides of (2.7) written for all $i \in V$ by x_i, and the other (quadratic) based on

introducing a term in (2.7) that makes the constraint redundant when $i \notin C$. The corresponding equations are given by

$$x_i \sum_{j,k \in N_G(i), j<k} a_{ij} a_{ik} a_{kj} x_j x_k \geq \alpha \binom{\sum_{k \in V} a_{ik} x_k}{2} x_i \quad \forall i \in V \tag{2.7a}$$

and

$$\sum_{j,k \in N_G(i), j<k} a_{ij} a_{ik} a_{kj} x_j x_k \geq \alpha \binom{\sum_{k \in V} a_{ik} x_k}{2} - \alpha \binom{d_G(i)}{2} (1 - x_i) \quad \forall i \in V, \tag{2.7b}$$

respectively. The term $\alpha \binom{d_G(i)}{2}(1 - x_i)$ vanishes if $x_i = 1$ (that is, $i \in C$), in which case (2.7) is satisfied, and makes the constraint (2.7b) redundant if $x_i = 0$ (that is, $i \notin C$). Obviously, only one set of the local clustering coefficient constraints (2.7a), (2.7b) needs to be used, leading to two alternative models, which we will refer to as the *cubic model* and the *quadratic model*, respectively.

Finally, the third set of the local α-cluster constraints assures that the induced subgraph $G[C]$ is connected and is referred to as the *connectivity constraints*. While there are several ways to describe connectivity mathematically, here we adopt a compact model of [57] recently proposed to bound the diameter of a graph when searching for k-clubs. A k-club is defined as a subset of vertices inducing a subgraph of diameter at most k [44]. By simply setting $k = n - 1$, the model can be used to enforce connectivity (see 2.2.2).

To make use of off-the-shelf binary linear optimization software for solving the proposed models, we linearize the nonlinear constraints (2.7a) and (2.7b), respectively. The resulting mixed integer programming (MIP) formulations are derived in 2.2.3.

Note that if we drop the connectivity requirement from the definition of an α-

cluster, a feasible solution to each of the considered optimization models corresponds to a disjoint union of α-clusters. That is, the vertices corresponding to each connected component of the subgraph corresponding to a feasible solution form an α-cluster. In Section 3, we focus on a special case of α-cluster with no connectivity constraints are imposed.

2.2.2 Connectivity Constraints

Let $p_{ij}^{(\ell)}$ be the binary variable indicating that there is a path of length ℓ in between the nodes i and j in the induced subgraph $G[C]$. Also, let L be an upper bound on the desired diameter of $G[C]$ (we can put $L = n - 1$ if we are only concerned with connectivity). Then the following set of constraints ensures that the subset of vertices C defined by the characteristic vector x induces a subgraph $G[C]$ of diameter at most L [57]:

$$\sum_{l=2}^{L} p_{ij}^{(\ell)} \geq x_i + x_j - 1 \;\; \forall \{i,j\} \notin E \tag{2.8}$$

$$p_{ij}^{(2)} \leq x_i \;\; \forall i,j \in V \tag{2.9}$$

$$p_{ij}^{(2)} \leq x_j \;\; \forall i,j \in V \tag{2.10}$$

$$p_{ij}^{(2)} \leq \sum_{k \in V} a_{ik} a_{kj} x_k \;\; \forall i,j \in V \tag{2.11}$$

$$p_{ij}^{(2)} \geq \frac{1}{n}(\sum_{k \in V} a_{ik} a_{kj} x_k) + x_i + x_j - 2 \;\; \forall i,j \in V \tag{2.12}$$

$$p_{ij}^{(\ell)} \leq x_i \;\; \forall i,j \in V, \;\; \ell \in \{3,\ldots,L\} \tag{2.13}$$

$$p_{ij}^{(\ell)} \leq \sum_{k \in V} a_{ik} p_{kj}^{(\ell-1)} \;\; \forall i,j \in V, \;\; \ell \in \{3,\ldots,L\} \tag{2.14}$$

$$p_{ij}^{(\ell)} \geq \frac{1}{n}(\sum_{k \in V} a_{ik} p_{kj}^{(\ell-1)}) + x_i - 1 \;\; \forall i,j \in V, \;\; \ell \in \{3,\ldots,L\}. \tag{2.15}$$

The reader is referred to [57] for a detailed explanation and derivation of these constraints.

2.2.3 Linearization of the Cubic and Quadratic Models

We use the reformulation linearization technique (RLT) introduced by [54] for linearization. RLT has two steps, reformulation and linearization by the constraint generation technique. In the reformulation step, we introduce two new sets of decision variables: ω_{ij} and y_{ijk} as follows:

$$\omega_{ij} = \begin{cases} 1, & \text{if } x_i = x_j = 1; \\ 0, & \text{otherwise,} \end{cases} \tag{2.16}$$

$$y_{ijk} = \begin{cases} 1, & \text{if } x_i = x_j = x_k = 1; \\ 0, & \text{otherwise.} \end{cases} \tag{2.17}$$

To apply the reformulation process, we multiply the bound constraints for binary variables as follows. For two binary variables x_j and x_k, we have the bound constraints

$$x_i \geq 0, \;\; 1 - x_i \geq 0, \quad x_j \geq 0, \;\; 1 - x_j \geq 0. \tag{2.18}$$

We multiply each constraint involving x_i with each constraint involving x_j and then replace $x_i x_j$ product term with a new variable ω_{ij}. We obtain the following set of constraints:

$$\omega_{ij} \geq 0, \;\; \omega_{ij} - x_j - x_i \geq -1, \;\; \omega_{ij} - x_i \geq 0, \;\; \omega_{ij} - x_j \geq 0. \tag{2.19}$$

Similarly, for three binary variables x_i, x_j and x_k, we add the bound constraints on x_k,

$$x_k \geq 0, \;\; 1 - x_k \geq 0, \tag{2.20}$$

to the constraints in (2.18) and consider products of triples of constraints from (2.18), (2.20) involving each of the three variables. We replace each product $x_i x_j x_k$ with a new variable y_{ijk}. This will add $2^3 = 8$ linear class of constraints:

$$y_{ijk} \geq 0, \;\; \omega_{ij} - y_{ijk} \geq 0, \;\; \omega_{ik} - y_{ijk} \geq 0, \;\; \omega_{jk} - y_{ijk} \geq 0, \tag{2.21}$$

$$y_{ijk} + x_i - \omega_{ik} - \omega_{ij} \geq 0, \;\; y_{ijk} + x_j - \omega_{jk} - \omega_{ij} \geq 0, \tag{2.22}$$

$$y_{ijk} + x_k - \omega_{ik} - \omega_{jk} \geq 0, y_{ijk} + x_i - \omega_{ik} - \omega_{ij} - \omega_{jk} + x_j + x_k \leq 1. \tag{2.23}$$

The Cubic Model

The cubic model consists of the objective function (2.5) and the constraints (2.6), (2.7a), (2.8)–(2.15). A linearized version of first formulation after applying RLT is:

$$\text{maximize} \sum_{i \in V} x_i \tag{2.24}$$

subject to

$$\sum_{j \in J} a_{ij} x_j \geq 2x_i \quad \forall i \in V \tag{2.25}$$

$$\sum_{j \in N_G(i)} \sum_{j < k \in N_G(i)} a_{ij} a_{ik} a_{kj} y_{ijk} \geq 2\alpha \sum_{k \in N_G(i)} a_{ij} a_{ik} \omega_{kj} \quad \forall i \in V \tag{2.26}$$

$$\omega_{ij} - y_{ijk} \geq 0, \;\; \omega_{ik} - y_{ijk} \geq 0, \;\; \omega_{jk} - y_{ijk} \geq 0 \quad \forall i, j, k \in V \tag{2.27}$$

$$y_{ijk} + x_i - \omega_{ik} - \omega_{ij} \geq 0, \;\; y_{ijk} + x_j - \omega_{jk} - \omega_{ij} \geq 0 \quad \forall i, j, k \in V \tag{2.28}$$

$$y_{ijk} + x_k - \omega_{ik} - \omega_{kj} \geq 0 \quad \forall i, j, k \in V \tag{2.29}$$

$$y_{ijk} + x_i - \omega_{ik} - \omega_{ij} - \omega_{jk} + x_j + x_k \leq 1 \quad \forall i, j, k \in V \tag{2.30}$$

$$\sum_{l=2}^{L} p_{ij}^{(\ell)} \geq x_i + x_j - 1 \quad \forall \{i, j\} \notin E \tag{2.31}$$

$$p_{ij}^{(2)} \leq x_i, \quad p_{ij}^{(2)} \leq x_j, \quad p_{ij}^{(2)} \leq \sum_{k \in V} a_{ik} a_{kj} x_k \quad \forall i, j \in V \tag{2.32}$$

$$p_{ij}^{(2)} \geq \frac{1}{n}(\sum_{k \in V} a_{ik} a_{kj} x_k) + x_i + x_j - 2 \quad \forall i, j \in V \tag{2.33}$$

$$p_{ij}^{(\ell)} \leq x_i, \quad p_{ij}^{(\ell)} \leq \sum_{k \in V} a_{ik} p_{kj}^{(\ell-1)} \quad \forall i, j \in V \quad \ell \in \{3, \ldots, L\} \tag{2.34}$$

$$p_{ij}^{(\ell)} \geq \frac{1}{n}(\sum_{k \in V} a_{ik} p_{kj}^{(\ell-1)}) + x_i - 1 \quad \forall i, j \in V \quad \ell \in \{3, \ldots, L\} \tag{2.35}$$

$$x_i, \omega_{ij}, y_{ijk}, p_{ij}^{(\ell)} \in \{0, 1\}, \forall i, j, k \in V. \tag{2.36}$$

The Quadratic Model

The quadratic model is composed of the objective function (2.5) and the constraints (2.6), (2.7b), (2.8)–(2.15). As mentioned earlier, the main difference between the cubic and quadratic model is in the equations (2.7a) and (2.7b). Linearized version of the quadratic model is:

$$\text{maximize} \sum_{i \in V} x_i \tag{2.37}$$

subject to

$$\sum_{j \in J} a_{ij} x_j \geq 2x_i \quad \forall i \in V \tag{2.38}$$

$$\sum a_{ij} a_{ik} a_{kj} \omega_{kj} \geq 2\alpha \sum_{k \in N_G(i)} a_{ij} a_{ik} \omega_{kj} - M(1 - x_i) \tag{2.39}$$

$$\omega_{ij} - x_j - x_i \geq -1, \;\; \omega_{ij} - x_i \geq 0, \;\; \omega_{ij} - x_j \geq 0 \tag{2.40}$$

$$\sum_{\ell=2}^{L} p_{ij}^{(\ell)} \geq x_i + x_j - 1 \quad \forall \{i,j\} \notin E \tag{2.41}$$

$$p_{ij}^{(2)} \leq x_i, \quad p_{ij}^{(2)} \leq x_j, \quad p_{ij}^{(2)} \leq \sum_{k \in V} a_{ik} a_{kj} x_k \quad \forall i,j \in V \tag{2.42}$$

$$p_{ij}^{(2)} \geq \frac{1}{n}(\sum_{k \in V} a_{ik} a_{kj} x_k) + x_i + x_j - 2 \quad \forall i,j \in V \tag{2.43}$$

$$p_{ij}^{(\ell)} \leq x_i, \quad p_{ij}^{(\ell)} \leq \sum_{k \in V} a_{ik} p_{kj}^{(\ell-1)} \quad \forall i,j \in V \quad \ell \in \{3,\ldots,L\} \tag{2.44}$$

$$p_{ij}^{(\ell)} \geq \frac{1}{n}(\sum_{k \in V} a_{ik} p_{kj}^{(\ell-1)}) + x_i - 1 \quad \forall i,j \in V \quad \ell \in \{3,\ldots,L\} \tag{2.45}$$

$$x_i, \omega_{kj} \in \{0,1\}, \forall i,j,k \in V. \tag{2.46}$$

2.2.4 The Triangle Model

In this section we consider an alternative formulation for solving the considered problem, referred to as the *triangle model*. In this formulation, we use a value-disjunction technique to model the required number of triangles in the subgraph induced by an α-cluster. Moreover, we use a slightly modified version of k-club constraints [58] and show that we can relax the integrality requirements for almost all variables in the corresponding formulation to enhance the performance of MIP solvers.

For any $i \in V$, let

$$z_{id} = \begin{cases} 1, & \text{if } \sum\limits_{j \in V} a_{ij} x_j = d, \\ 0, & \text{otherwise.} \end{cases}$$

Namely, $z_{id} = 1$ if and only if the degree of node i in the subgraph $G[C]$ is d. This

requirement can be enforced by the following constraints:

$$\sum_{d=0}^{d_G(i)} dz_{id} = \sum_{j \in V} a_{ij} x_j, \quad \sum_{d=0}^{d_G(i)} z_{id} = 1, \quad \forall i \in V. \tag{2.47}$$

Since we are only concerned with $i \in C$ (i.e., $x_i = 1$), we use constraints (2.47) in a relaxed form:

$$\sum_{d=1}^{d_G(i)} dz_{id} \geq \sum_{j \in V} a_{ij} x_j - M(1 - x_i), \quad \sum_{d=1}^{d_G(i)} z_{id} = 1, \quad \forall i \in V, \tag{2.48}$$

where M is a sufficiently large constant, e.g., $M = d_G(i)$.

Next, for each edge $\{i, j\} \in E$, we introduce a variable ω_{ij} such that $\omega_{ij} = 1$ if and only if $x_i = 1$, $x_j = 1$. Then, for any node $i \in V$ the clustering coefficient constraint can be written as

$$\sum_{\{k,j\} \in E, k<j} a_{ik} a_{ij} \omega_{kj} \geq \alpha \sum_{d=2}^{d_G(i)} \frac{d(d-1)}{2} z_{id}. \tag{2.49}$$

Observe that for any $i \in V$ the number of variables z_{id} is $d_G(i)$. Hence, we only need $2|E|$ of variables z_{id} and $|E|$ of variables ω_{ij}.

Also, for every pair of nodes i and j define $u_{ij}^{(\ell)} \in \{0, 1\}$ to be 1 if and only if both nodes i and j are in local α-cluster C and there is a path of length *at most* ℓ between i and j in $G[C]$. Note that in the formulations in 2.2.3, we use variables $p_{ij}^{(\ell)}$ indicating that there is a path of length (exactly) ℓ between i and j in $G[C]$. As it is can be seen from the formulation below, using variables $u_{ij}^{(\ell)}$ instead of $p_{ij}^{(\ell)}$ allows for a simpler modeling of recursive distance-based constrains. Then the triangle formulation can

be stated as follows:

$$\text{maximize} \quad \sum_{i \in V} x_i \tag{2.50}$$

$$\text{subject to} \quad \sum_{(k,j) \in E, k<j} a_{ik} a_{ij} \omega_{kj} \geq \alpha \sum_{d=2}^{d_G(i)} \frac{d(d-1)}{2} z_{id}, \quad \forall i \in V \tag{2.51}$$

$$\omega_{ij} \leq x_i, \quad \omega_{ij} \leq x_j, \quad \forall \{i,j\} \in E, \;\; i < j \tag{2.52}$$

$$\sum_{d=1}^{d_G(i)} d z_{id} \geq \sum_{j \in V} a_{ij} x_j - M(1 - x_i), \qquad \sum_{d=1}^{d_G(i)} z_{id} = 1, \quad \forall i \in V \tag{2.53}$$

$$u_{ij}^{(L)} \geq x_i + x_j - 1, \quad \forall i,j \in V, \quad i \neq j \tag{2.54}$$

$$u_{ij}^{(1)} = 0, \quad \forall \{i,j\} \notin E, \quad i \neq j \tag{2.55}$$

$$u_{ij}^{(\ell)} = u_{ij}^{(1)}, \quad \forall \{i,j\} \in E, \quad \ell \in \{2, \ldots, L\} \tag{2.56}$$

$$u_{ij}^{(\ell)} \leq \sum_{t:\ (i,t) \in E} u_{tj}^{(\ell-1)}, \quad \forall \{i,j\} \notin E, \quad \ell \in \{2, \ldots, L\} \tag{2.57}$$

$$u_{ij}^{(\ell)} \leq x_i, \quad u_{ij}^{(\ell)} \leq x_j, \quad u_{ij}^{(\ell)} = u_{ji}^{(\ell)}, \quad \forall i,j \in V, \quad \ell \in \{1, \ldots, L\} \tag{2.58}$$

$$u_{ij}^{(\ell)} \in \mathbb{R}_+, \quad \forall i,j \in V, \quad \ell \in \{1, \ldots, L\} \tag{2.59}$$

$$x_i \in \{0,1\} \qquad z_{id}, \omega_{ij} \in \mathbb{R}_+, \quad \forall i,j \in V, \quad d \in \{1, \ldots, d_G(i)\}. \tag{2.60}$$

Note that we relaxed the integrality requirements for variables $u_{ij}^{(\ell)}$, ω_{ij} and z_{id}. For the variables $u_{ij}^{(\ell)}$ and ω_{ij} this can be done without altering the optimal solutions due to the maximization nature of the problem. We show that z_{id} variables can also be relaxed in Proposition 1.

Proposition 1. *There exists an optimal solution of the triangle formulation denoted by* ($\mathbf{x}^*$, $\mathbf{y}^*$, $\mathbf{u}^*$, $\mathbf{z}^*$) *such that* $\mathbf{z}^*$ *has only binary components.*

Proof. Consider an instance of the triangle formulation with an optimal solution given by $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{u}^*, \bar{\mathbf{z}})$, where $\bar{\mathbf{z}}$ may not have only 0–1 components. Define $\mathbf{z}^* = \{z^*_{it} | i \in V, t = 1, \ldots, d_G(i)\}$ as follows.

$$z^*_{it} = \begin{cases} 1, \text{ if } \sum\limits_{j \in V} x^*_j = t, \text{ and } x_i = 1, \\ 0, \text{ if } \sum\limits_{j \in V} x^*_j \neq t, \text{ and } x_i = 1. \end{cases} \tag{2.61}$$

and

$$z^*_{it} = \begin{cases} 1, \text{ if } t = 1, \text{ and } x_i = 0, \\ 0, \text{ if } t > 1, \text{ and } x_i = 0. \end{cases} \tag{2.62}$$

Obviously, $(\mathbf{x}^*, \mathbf{z}^*)$ satisfy (2.53). Moreover, for any node $i \in V$ such that $x_i = 0$, $(\mathbf{x}^*, \mathbf{z}^*)$ satisfy (2.50). Also, without loss of generality we can assume that $\sum\limits_{t=1}^{d_G(i)} t\bar{z}_{it} = \sum\limits_{j \in V} x^*_j$, for all $i \in V$ such that $x_i = 1$. Then, for any convex function $f()$ and node $i \in V$ such that $x_i = 1$, applying Jensen's inequality we can conclude that

$$\sum_{t=1}^{d_G(i)} f(t)\bar{z}_{it} \geq f\left(\sum_{t=1}^{d_G(i)} t\bar{z}_{it}\right) = f\left(\sum_{j \in V} x^*_i\right) = f\left(\sum_{t=1}^{d_G(i)} t z^*_{it}\right) = \sum_{t=1}^{d_G(i)} f(t) z^*_{it},$$

since $\sum\limits_{t=1}^{d_G(i)} t z^*_{it} = \sum\limits_{t=1}^{d_G(i)} t\bar{z}_{it} = \sum\limits_{j \in V} x^*_j$. Hence, for the function $f(t) = \alpha\frac{t(t-1)}{2}$

$$\sum_{\{k,j\} \in E, k<j} a_{ik} a_{ij} y^*_{kj} \geq \alpha \sum_{t=2}^{d_G(i)} \frac{t(t-1)}{2} \bar{z}_{it} \geq \alpha \sum_{t=2}^{d_G(i)} \frac{t(t-1)}{2} z^*_{it}$$

Therefore, $(\mathbf{y}^*, \mathbf{z}^*)$ satisfy (2.50), and since $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{u}^*, \bar{\mathbf{z}})$ is an optimal solution, thus $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{u}^*, \mathbf{z}^*)$ is also an optimal solution of the triangle formulation. □

2.2.5 Finding Global α-Clusters

The cubic and triangle formulations for the maximum global α-cluster problem can be easily obtained from the cubic and triangle models above. In the cubic model, we can do so by replacing the constraints (2.26) with their surrogate constraints [32], which are obtained by summing up the corresponding inequalities over all $i \in V$. As for the triangle model, we first introduce nonnegative real variables y_{ijk} for any triplet of nodes (i, j, k) that form a triangle in G, i.e., $\{i, j\}, \{i, k\}, \{k, j\} \in E$. Let

$$\Delta = \{(i, j, k) : i < j < k, \{i, j\}, \{i, k\}, \{k, j\} \in E\}$$

be the set of all such triangles in G. Then we replace the constraints (2.51)–(2.52) with

$$3 \sum_{(i,j,k)\in\Delta} y_{ijk} \geq \alpha \sum_{i\in V} \sum_{d=2}^{d_G(i)} \frac{d(d-1)}{2} z_{id}, \tag{2.63}$$

$$y_{ijk} \leq x_i, y_{ijk} \leq x_j, y_{ijk} \leq x_k, \qquad \forall (i, j, k) \in \Delta. \tag{2.64}$$

2.3 Analysis of α-Clusters in Real-life Social Networks

We implemented the proposed formulations using FICO Xpress-IVE Version 1.24.04 solver. All computations were performed on a *Dell Precision WorkStation T7500®* computer with eight 2.40 GHz Intel Xeon® processors and 12 GB RAM.

In preliminary experiments, we verified that all three proposed models yield identical solutions and observed that the triangle model outperforms its competitors in terms of running time on all instances. Hence, we use the triangle model in all experiments discussed below.

We analyze four well-known social network instances by varying the clustering

coefficient threshold level, as described in the following four subsections.

2.3.1 Zachary's Karate Club

The first social network we analyze is the karate club example first studied by Zachary [66]. The corresponding graph describes social relationships among 34 members of a university karate club, where the nodes represent its members. If two members are friends outside the club, then an edge is present between them. The graph has the total of 78 edges. After a dispute between the instructor (Mr. Hi, a karate instructor, node 1) and the club's administrator (John A., the club president, node 34), the club split into two factions, with 16 and 18 members, respectively. Since the members of each group are known, this example is often used as a benchmark for network clustering algorithms.

We first study the effect the split had on local and global clustering coefficients. As can be seen from Table 2.1, global clustering coefficient increases after the split. For Mr. Hi's group, the increase is quite significant. However, in the case of John's faction the increase in global clustering coefficient is rather small. This may suggest that Mr. Hi's faction is more cohesive than John's. This hypothesis appears to be especially reasonable if one analyzes the aspects that led to the divide: Mr. Hi wished to raise the price of karate lessons, whereas John insisted on stabilizing prices. According to [66],

> "The supporters of Mr. Hi saw him as a fatherly figure who was their spiritual and physical mentor, and who was only trying to meet his own physical needs after seeing to theirs. The supporters of John A. and the other officers saw Mr. Hi as a paid employee who was trying to coerce his way into a higher salary."

Based on this description, Mr. Hi's supporters seemed to have formed a more ide-

Table 2.1: Global and average local clustering coefficients of Zachary's karate club before and after the split.

Clustering coefficient	Before split	Mr. Hi's faction	John's faction
Global	0.255682	0.418994	0.259615
Average local	0.570638	0.719712	0.651539

ologically united group than their opposition. Comparing the values of clustering coefficients in Table 2.1 we may also hypothesize that the global clustering coefficient is more indicative of a group's overall cohesiveness than t.

We computed all the largest global and local α-clusters in the karate club network for $\alpha = 1, 0.9, 0.8, 0.7$, and 0.6. The corresponding results are presented in Table 2.2. For each considered value of α, this table reports the size of a largest α-cluster ('Size') and lists the members of each largest α-cluster ('Members'). We observe that for $\alpha = 1$, the largest α-clusters, both global and local, are given by two maximum cliques of size 5, $\{1, 2, 3, 4, 8\}$ and $\{1, 2, 3, 4, 14\}$, which have 4 out of 5 vertices in common. The union of these two cliques gives the largest local and global α-cluster for $\alpha = 0.9$, see Figure 2.2 for an illustration. For $\alpha = 1$ and 0.9, all the optimal α-clusters consisted of members of Mr. Hi's faction only. However, for $\alpha = 0.8$ and below, all the largest global α-clusters consist of a mix of members of both factions, see Figure 2.3 for an illustration. As for maximum local α-clusters, there are only two optimal α-clusters for $\alpha = 0.7$ in Figure 2.4 and one – for $\alpha = 0.6$ in Figure 2.5 that contain an "outlier" node 9, representing a member of John's faction. Interestingly, [66] points out that "Person number 9 was a weak supporter of John but joined Mr. Hi's club after the split." These results confirm our earlier observation that local α-clusters provide a much better description of cohesive subgroups than global

α-clusters.

Next, we compute the largest local α-clusters that contain node 34 (John, the leader of one of the two factions). The corresponding results are reported in Table 2.3. For $\alpha = 1, 0.9, 0.8$, and 0.7, we obtain the same pair of alternative solutions given by cliques of size 4. For $\alpha = 0.6$, there is a unique maximum local α-cluster containing node 34. This solution consists of 6 nodes and is nothing else but the union of the two cliques obtained for higher values of α. Note that all the nodes listed in the optimal solutions in Table 2.3 are members of John's faction, which once again illustrates strong cohesiveness properties of local α-clusters.

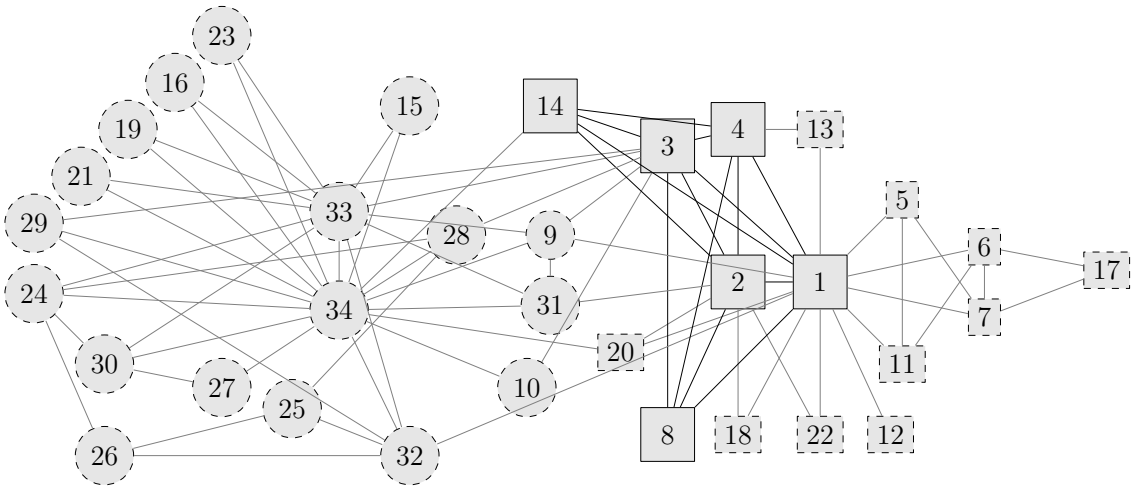


Figure 2.2: Zachary's karate club network with Mr. Hi's and John's faction members shown on the left and on the right, respectively. The nodes with a solid border show the members of the largest global and local α-cluster for $\alpha = 0.9$.

The CPU time used to compute each maximum local and global α-cluster has not exceeded 6 seconds, with local α-clusters computed slightly faster than the global ones. Taking into account that local α-clusters are also superior to the global ones in terms of their cohesiveness properties, we do not find global α-clusters to be of much

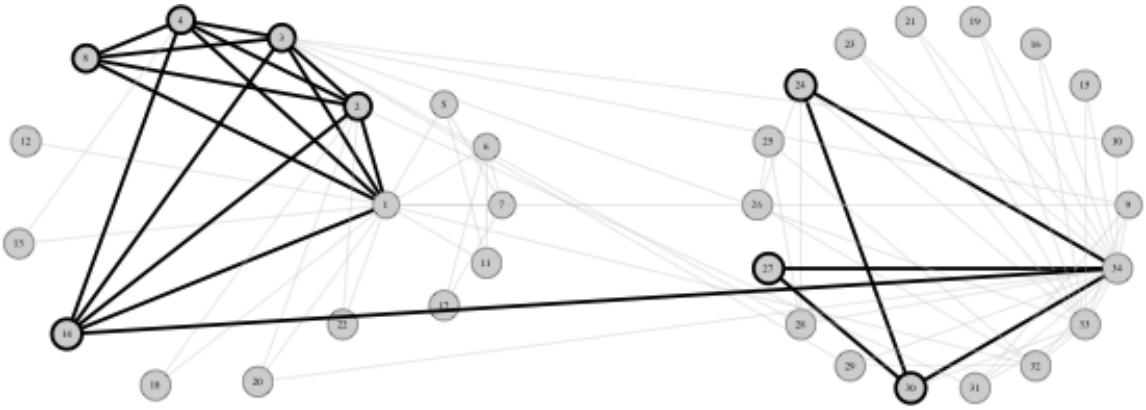


Figure 2.3: Zachary's karate club network with Mr. Hi's and John's faction members shown on the right and on the left, respectively. The nodes with a solid border show the members of the largest global α-cluster for $\alpha = 0.8$.

interest in studying cohesive subgroups and focus on computing local α-clusters in the remaining experiments.

2.3.2 Football Graph

Second, we analyze the football graph due to [31], which represents American college football games in year 2000. Nodes represent 115 teams that played in that season and 613 edges represent games played between them. These 115 teams are subdivided in 11 conferences. Usually, each team plays most of its games with the teams in the same conference. Table 2.4 presents clustering coefficient information for each conference. In addition, there were 5 teams that did not belong to any conference, referred to as independents.

We compute the largest local α-clusters in the football network for $\alpha = 1, 0.9, 0.8$, and 0.75. The corresponding results are presented in Table 2.5. Visually, Figure 2.6 shows the solution with respect to the whole graph when α is equal to 1. Figure 2.7 presents an alternative solution. for the same α value. As pointed out in Table 2.5, we find the same solution for α values 1 and 0.9. For each considered value of α,

Table 2.2: Description of all the largest α-clusters in Zachary's karate club network. The members of John's faction are shown in bold.

	Global		Local	
α	Size	Members	Size	Members
1	5	1,2,3,4,8	5	1,2,3,4,8
		1,2,3,4,14		1,2,3,4,14
0.9	6	1,2,3,4,8,14	6	1,2,3,4,8,14
0.8	10	1,2,3,4,8,14,**23**,**24**,**26**,**34**	6	1,2,3,4,8,14
		1,2,3,4,8,14,**24**,**26**,**30**,**32**		
		1,2,3,4,8,14,**24**,**27**,**28**,**30**		
		1,2,3,4,8,14,**15,24,26,34**		
		1,2,3,4,8,,**24,26,30,33,34**		
		1,2,3,4,8,14,,**24,25,26,33**		
		1,2,3,4,8,14,,**24,26,27,34**		
		1,2,3,4,8,,**24,27,30,33,34**		
		1,2,3,4,8,14,**19,24,26,34**		
		1,2,3,4,8,14,**24,26,30,34**		
		1,2,3,4,8,14,**24,27,30,34**		
		1,2,3,4,8,14,**16,24,26,34**		
		1,2,3,4,8,14,**21,24,26,34**		
		2,3,4,5,**9**,**15,25,28,29,31**		
0.7	13	1,2,3,4,6,7,8,14,17,**24,27,30,34**	6	1,2,3,4,13,14
				1,2,3,4,8,20
				1,2,3,4,14,22
				1,2,3,4,14,18
				1,2,3,4,8,13
				1,2,3,4,8,14
				1,2,3,4,8,**9**
				1,2,3,4,8,18
				1,2,3,4,8,22
				1,2,3,4,**9**,14
				1,2,3,4,14,20
0.6	15	1,2,3,4,5,6,7,8,14, **23,24,27,30,33,34**	7	1,2,3,4,8,**9**,14
		1,2,3,4,5,6,7,8,11,**17,23,24,30,33,34**		1,2,3,4,8,13,14
		1,2,3,4,6,7,8,14,**17,25,26,27,30,32,34**		1,2,3,4,8,14,18
		1,2,3,4,6,7,8,11,13,14,**17,24,27,30,34**		1,2,3,4,8,14,20
		and 184 more solutions		1,2,3,4,8,14,22

Table 2.3: Description of the largest local α-clusters containing node 34 in Zachary's karate club network.

α	Size	Members
1, 0.9, 0.8, 0.7	4	9, 31, 33, 34
		24, 30, 33, 34
0.6	6	9, 24, 30, 31, 33, 34

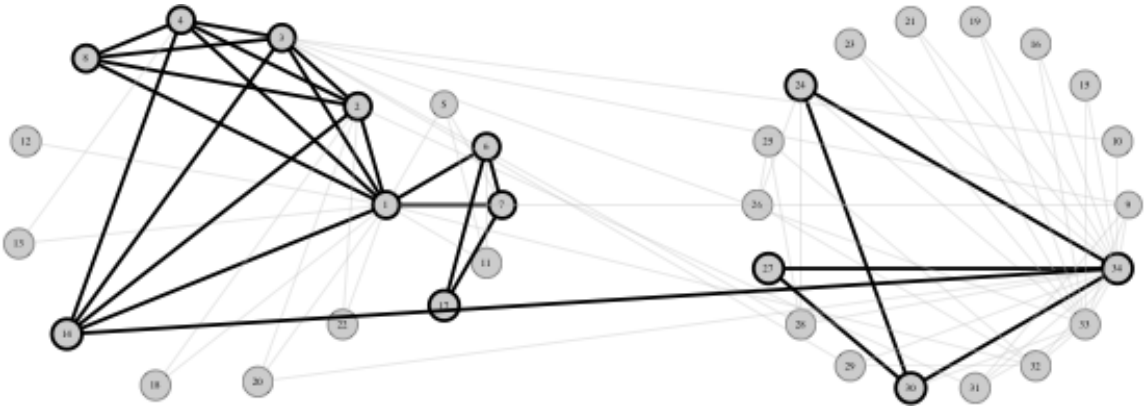


Figure 2.4: Zachary's karate club network with Mr. Hi's and John's faction members shown on the right and on the left, respectively. The nodes with a solid border show the members of the largest global α-cluster for $\alpha = 0.7$.

this table reports the size of a largest α-cluster ('Size') and lists the nodes of each optimal α-cluster ('Solutions'). We observe that for $\alpha = 1$ and 0.9, the largest α-clusters are given by two maximum cliques of size 9. The first maximum clique corresponds to Atlantic Coast Conference (ACC), and the second maximum clique corresponds to Western Athletic Conference (WAC). As we decrease α to 0.8, we obtain a unique optimal solution, which consists of 9 teams from WAC and three teams from other conferences, namely Conference USA (Houston), Mountain West (New Mexico State), and Sunbelt (Nevada Las Vegas).

For a slightly smaller value of $\alpha = 0.75$, we obtain 10 optimal solutions, each

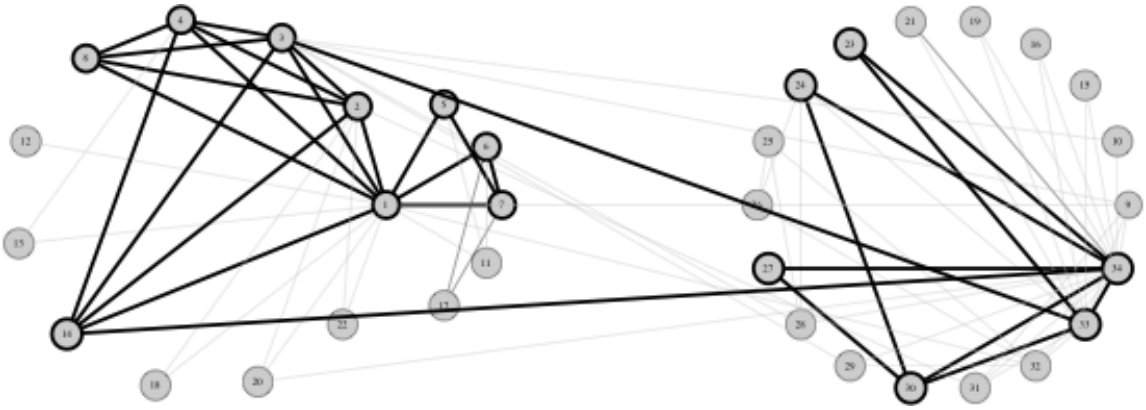


Figure 2.5: Zachary's karate club network with Mr. Hi's and John's faction members shown on the right and on the left, respectively. The nodes with a solid border show the members of the largest global α-cluster for $\alpha = 0.6$.

Table 2.4: College football conference sizes, global, average local and minimum local clustering coefficients.

		Clustering coefficient		
Conference	Size	Global	Average local	Minimum local
Atlantic Coast	9	1.00	1.00	1.00
Mid American	13	0.643	0.641	0.57
Big East	8	1.00	1.00	1.00
Conference USA	10	0.82	0.74	0.00
SEC	12	0.68	0.68	0.68
PAC10	10	0.86	0.86	0.86
Mountain West	8	1.00	1.00	1.00
Big 10	11	0.75	0.75	0.75
Big 12	12	0.68	0.68	0.68
Western Athletic	10	0.93	0.86	0.00
Sunbelt	7	0.75	0.83	0.33

consisting of 25 teams. The solutions roughly correspond to 3 conferences in their size and consist of 3 large clusters corresponding to different conferences and a couple additional teams outside. Most of the solutions reported in Table 2.5 for $\alpha = 0.75$ consist of 8 teams from ACC, 8 teams from WAC, 8 teams from Mountain West

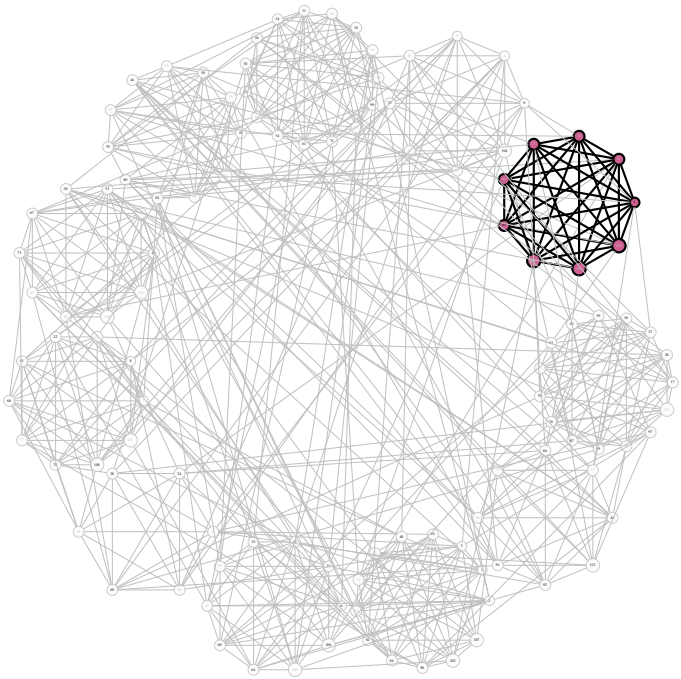


Figure 2.6: Football network with local α-cluster when α is 1 and 0.9.

Conference, and 1 more team that plays two teams from one of these three conferences (e.g., Vanderbilt, Oklahoma, and Arizona, respectively, in the first three reported solutions). Figure 2.8 shows an example solution with respect to the whole graph.

Again, we observe that local α-clusters with high values of α consistently describe tightly knit groups of actors. However, as α is decreased from 0.8 to 0.75, we see a dramatic increase in the size of α-clusters, which for $\alpha = 0.75$ consist of several tightly knit clusters. This can be explained by the clearly-defined modular structure of the football network, where teams from the same conference are clustered together in a supervised fashion due to the conference schedule requirements. The same is not expected to be the case in social networks where actors are free to decide on their

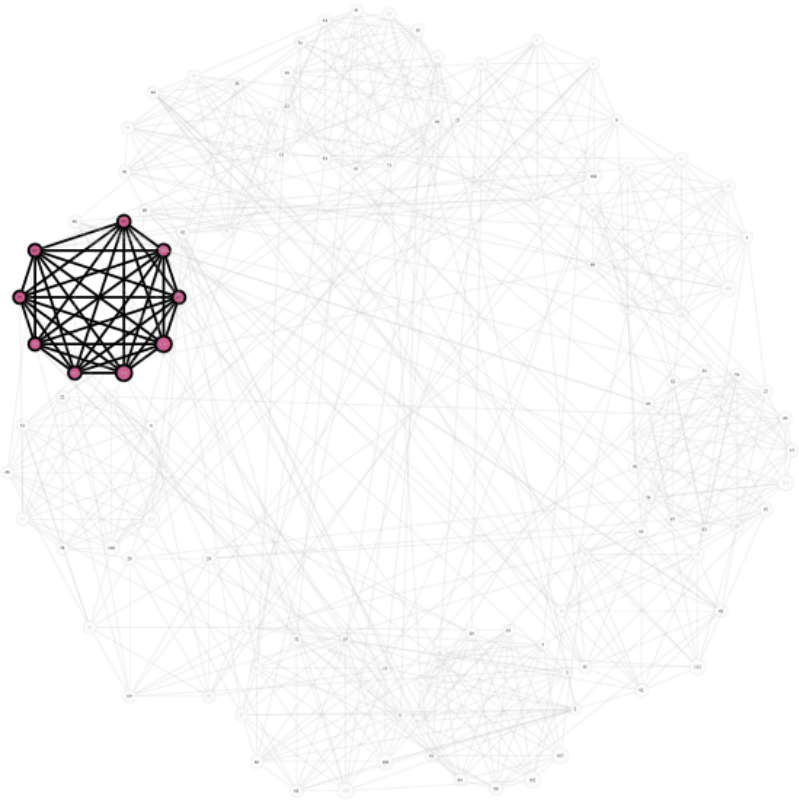


Figure 2.7: Football network with local α-cluster when α is 1 and 0.9, alternative solution.

local interactions.

2.3.3 Santa Fe Institute (SFI) Collaboration Network

The third considered network is the largest connected component of SFI collaboration network [31]. There are 118 scientists and an edge is drawn if they coauthored one or more articles between 1999 and 2000. There are 200 collaborations between the scientists.

We compute the largest local α-clusters in the SFI collaboration network for $\alpha = 1, 0.9, 0.8, 0.7$, and 0.6. The corresponding results are presented in Table 2.6. We observe that for $\alpha = 1$, the largest α-clusters are given by five maximum cliques

Table 2.5: Description of the largest local α-clusters in the football network.

α	Size	Solutions
1, 0.9	9	2,26,34,38,46,90,104,106,110
		47,50,54,68,74,84,89,111,115
0.8	12	47,49,50,54,68,70,74,84,89,105,111,115
0.75	25	{1,2,5,10,17,24,26,38,42,46,47,50,54,
		63,74,84,89,90,94,104,105,106,110,111,115}
		{1,2,5,10,17,24,26,38,42,46,47,50,54,
		74,84,85,89,90,94,104,105,106,110,111,115}
		{1,2,5,10,17,23,24,26,38,42,46,47,50,54,74,
		84,89,90,94,104,105,106,110,111,115}
		{1,5,10,17,24,26,34,38,42,46,47,50,54,74,
		84,89,90,94,104,105,106,109,110,111,115}
		{1,2,5,10,17,24,26,38,42,46,47,50,54,74,84,
		89,90,94,104,105,106,109,110,111,115}
		{1,5,10,17,24,26,34,38,42,46,47,50,54,74,84
		85,89,90,94,104,105,106,110,111,115}
		{1,5,9,10,17,24,26,34,38,42,46,47,50,54,74,
		84,89,90,94,104,105,106,110,111,115}
		{1,5,10,17,23,24,26,34,38,42,46,47,50,54,74,
		84,89,90,94,104,105,106,110,111,115}
		{1,5,10,17,24,26,34,38,42,46,47,50,54,63,74,
		84,89,90,94,104,105,106,110,111,115}
		{1,2,5,9,10,17,24,26,38,42,46,47,50,54,74,
		84,89,90,94,104,105,106,110,111,115}

of size 5. The first two of them correspond to researchers in statistical physics, the third one is a cohesive subset of RNA structure group, and the last two correspond to mathematical ecologists. For $\alpha = 0.8$, we find a unique optimal solution which corresponds to scientists working on RNA structure. For a smaller value of $\alpha = 0.6$, we obtain two similar optimal solutions (differing by just one member), both of which correspond to RNA structure scientists. In each case, we observe that α-clusters correspond to groups of researchers with similar interests.

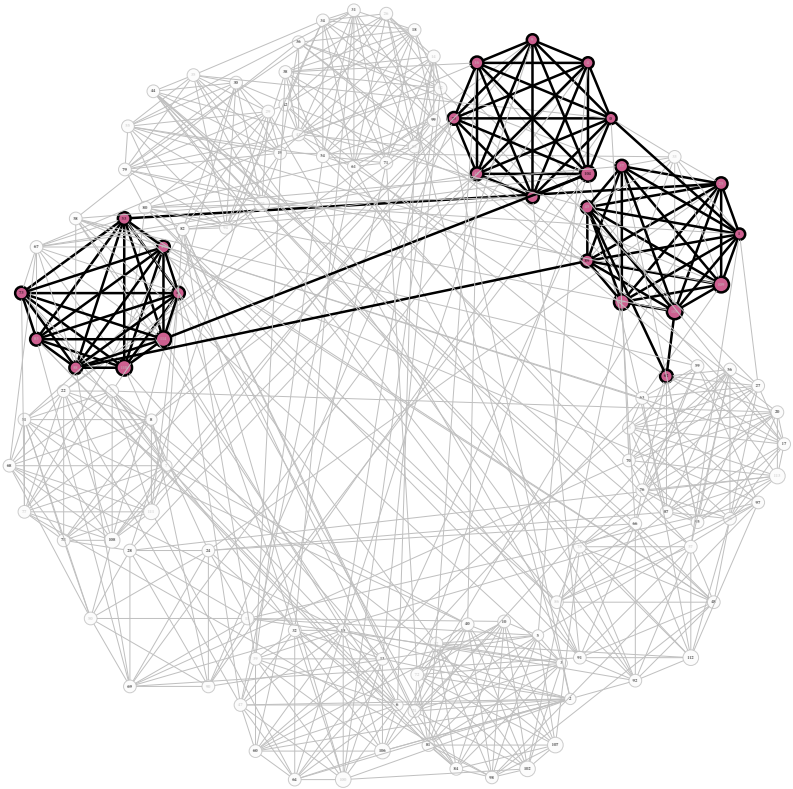


Figure 2.8: Football network with local α-cluster when α is 0.75.

2.3.4 Dolphins Network

Another social network we analyze is dolphins network constructed by [43]. In the corresponding study, interactions of 62 dolphins were traced, with the edges describing the pairs of dolphins with higher than expected level of interaction between them. The total of 159 such pairs were recorded. In the literature, it is repeatedly mentioned that there are two main groups consisting of 20 and 42 members, referred to as group 1 and group 2, respectively. We computed the largest local α-clusters in the dolphins network for $\alpha = 1, 0.9, 0.8, 0.7$, and 0.6. The corresponding results are presented in Table 2.7. For $\alpha = 1$, the largest α-clusters are given by three maximum cliques of size 5, one of which corresponds to group 1, and the other two – to group

Table 2.6: Description of the largest local α-clusters in Newman's SFI collaboration network.

α	Size	Solution
1	5	6,17,18,28,74
		18,44,49,52,74
		27,37,64,93,117
		42,100,102,107,111
		55,68,94,100,105
0.8	6	3,27,37,64,93,117
0.6	8	3,27,37,48,56,64,93,117
		3,27,37,48,57,64,93,117

2. For $\alpha = 0.9$, there is a unique optimal solution of size 6, which is the union of the two maximum cliques corresponding to group 2. We note that each of the optimal α-clusters reported in Table 2.7 consists of nodes representing dolphins from one of the two groups only. This, once again, indicates that local α-clusters tend to describe strong cohesive subgroups in real-life social networks.

2.3.5 Terrorist Network Compiled by Krebs

Finally, we consider the terrorist interaction network compiled by [38] using the information available about the tragic events of September 11, 2001. This network consists of 62 nodes representing terrorists connected to the attacks (see Table 2.8 for the list of names), and the edges correspond to pairs of persons that were known to interact the past. In total there were 153 interactions observed.

We compute the largest α-clusters for $\alpha = 1, 0.9, 0.8, 0.7$ and 0.6. The results are reported in Table 2.9. There are three maximum cliques of size 6 for $\alpha = 1$, each of them shows the close relationships between hijackers. All members of the first two groups were among the hijackers involved in the World Trade Center attacks, and all nodes in the third group were members of the Hamburg terror cell [38]. For $\alpha = 0.9$,

Table 2.7: Description of the largest local α-clusters in dolphins network. An asterisk (*) indicates the members of group 1; unmarked nodes represent group 2.

α	Size	Solution
1	5	7*,10*,14*,18*,58*
		19,25,30,46,52
		19,22,30,46,52
0.9	6	19,22,25,30,46,52
0.8	6	6*,7*,10*,14*,18*,58*
		7*,10*,14*,18*,42*,58*
		7*,10*,14*,18*,55*,58*
		7*,10*,14*,42*,55*,58*
		15,17,34,38,39,44
		16,19,25,30,46,52
		19,22,25,30,46,52
0.7	7	7*,10*,14*,18*,42*,55*,58*
		16,19,22,25,30,46,52
0.6	8	2*,6*,7*,10*,14*,42*,55*,58*
		2*,7*,10*,14*,,33*,42*,55*,58*
		6*,7*,10*,14*,18*,42*,55*,58*
		15,17,21,22,34,38,39,44
		15,17,21,34,35,38,39,41
		15,17,21,34,35,38,39,44
		15,17,21,34,35,38,39,51
		15,17,21,34,35,38,41,51
		15,17,21,34,35,38,44,51
		15,17,21,34,38,39,41,44
		15,17,21,34,38,39,41,51
		15,17,21,34,38,39,44,51
		15,17,21,34,38,41,44,51
		15,17,22,34,38,39,44,53
		15,17,34,37,38,39,41,51

node 23 is added to one of the cliques, which also corresponds to a terrorist that took part in the WTC North attack. Likewise, for $\alpha = 0.8$ node 10 is added to the largest 0.9-cluster, an alleged organizer and financier of the 9/11 attacks. For $\alpha = 0.7$ there are two optimal solutions of size 9 differing by just one node; namely, node 21

Table 2.8: Description of the nodes in the terrorist network.

Node	Name	Node	Name
1	Wail Alshehri	32	Agus Budiman
2	Satam Suqami	33	Ahmed K. I. Samir Al-Ani
3	Nabil alMarabh	34	Majed Moqed
4	Raed Hijazi	35	Rayed Mohammed Abdullah
5	Waleed Alshehri	36	Faisal Al Salmi
6	Ahmed Alghamdi	37	Bandar Alhazmi
7	Mohand Alshehri	38	Abdelghani Mzoudi
8	Saeed Alghamdi	39	Abu Qatada
9	Fayez Ahmed	40	Abu Walid
10	Mustafa Ahmed al-Hisawi	41	Abu Zubeida
11	Abdul Aziz Al-Omari	42	Ahmed Ressam
12	Hamza Alghamdi	43	David Courtaillier
13	Ahmed Alnami	44	Djamal Beghal
14	Ahmed Alhaznawi	45	Essid Sami Ben Khemais
15	Mamoun Darkazanli	46	Essoussi Laaroussi
16	Mohamed Abdi	47	Fahid al Shakri
17	Marwan Al-Shehhi	48	Haydar Abu Doha
18	Zakariya Essabar	49	Imad Eddin Barakat Yarkas
19	Salem Alhazmi	50	Jean-Marc Grandvisir
20	Nawaf Alhazmi	51	Jerome Courtaillier
21	Said Bahaji	52	Kamel Daoudi
22	Ziad Jarrah	53	Lased Ben Heni
23	Mohamed Atta	54	Madjid Sahoune
24	Abdussattar Shaikh	55	Mamduh Mahmud Salim
25	Mounir El Motassadeq	56	Mehdi Khammoun
26	Khalid Al-Mihdhar	57	Mohamed Bensakhria
27	Zacarias Moussaoui	58	Mohammed Belfas
28	Ramzi Bin al-Shibh	59	Nizar Trabelsi
29	Lofti Raissi	60	Samir Kishk
30	Hani Hanjour	61	Seifallah ben Hassine
31	Osama Awadallah	62	Tarek Maaroufi

from the first solution is replaced by node 32 in the second solution. Most of the nodes included in the solutions were linked to the Hamburg terror cell. For $\alpha = 0.6$, we have a unique optimal solution of size 13 (see Figure 2.9 for an illustration). Interestingly, 11 of the 13 nodes correspond to actual hijackers involved in the WTC

Table 2.9: Description of the largest local α-clusters in Krebs's terrorist network. The hijackers that participated in the WTC North, WTC South, Pentagon, and Pennsylvania attacks are indicated by the upper bar ($^{-}$) lower bar (_), asterisk (*), and hat sign ($\hat{}$), respectively. In addition, the members of the Hamburg terror cell are marked with a dagger (†).

α	Size	Solution
1	6	$\bar{1}$,$\bar{2}$,$\bar{5}$,$\underline{9}$,$\bar{11}$,$\underline{17}^{\dagger}$
		$\bar{1}$,$\bar{2}$,$\bar{5}$,$\underline{9}$,$\bar{11}$,$\bar{23}^{\dagger}$
		$\underline{17}^{\dagger}$,$18^{\dagger}$,$21^{\dagger}$,$\hat{22}^{\dagger}$,$\bar{23}^{\dagger}$,$28^{\dagger}$
0.9	7	$\bar{1}$,$\bar{2}$,$\bar{5}$,$\underline{9}$,$\bar{11}$,$\underline{17}^{\dagger}$,$\bar{23}^{\dagger}$
0.8	8	$\bar{1}$,$\bar{2}$,$\bar{5}$,$\underline{9}$,10,$\bar{11}$,$\underline{17}^{\dagger}$,$\bar{23}^{\dagger}$
0.7	9	$\underline{17}^{\dagger}$,$18^{\dagger}$,$21^{\dagger}$,$\hat{22}^{\dagger}$,$\bar{23}^{\dagger}$,$28^{\dagger}$,29,30^{*},35
		$\underline{17}^{\dagger}$,$18^{\dagger}$,$\hat{22}^{\dagger}$,$\bar{23}^{\dagger}$,$28^{\dagger}$,29,30^{*},32,35
0.6	13	$\bar{1}$,$\bar{2}$,$\bar{5}$,$\underline{9}$,$\bar{11}$,$\underline{17}^{\dagger}$,19^{*},20^{*},24,26^{*},30^{*},31,34^{*}

South, WTC North, and Pentagon attacks. The two remaining nodes (24 and 31) were linked to two of the hijackers involved in the Pentagon attack. In particular, node 24 represents an FBI informant, whose "contacts with the hijackers, had they been capitalized on, would have given the San Diego FBI field office perhaps the Intelligence Community's best chance to unravel the September 11 plot" according to Joint Inquiry into Intelligence Community Activities before and after the Terrorist Attacks of September 11, 2001.

2.3.6 Other Social Network Instances

To asses the scalability of the proposed method, we conducted numerical experiments for the maximum local α-cluster problem on other benchmark social network instances obtained from Trick's graph coloring page (http://mat.gsia.cmu.edu/COLOR03/) and Pajek dataset (http://vlado.fmf.uni-lj.si/pub/networks/data/), as well as Barabási-Albert [11] and Erdös-Rényi [23] random graphs.

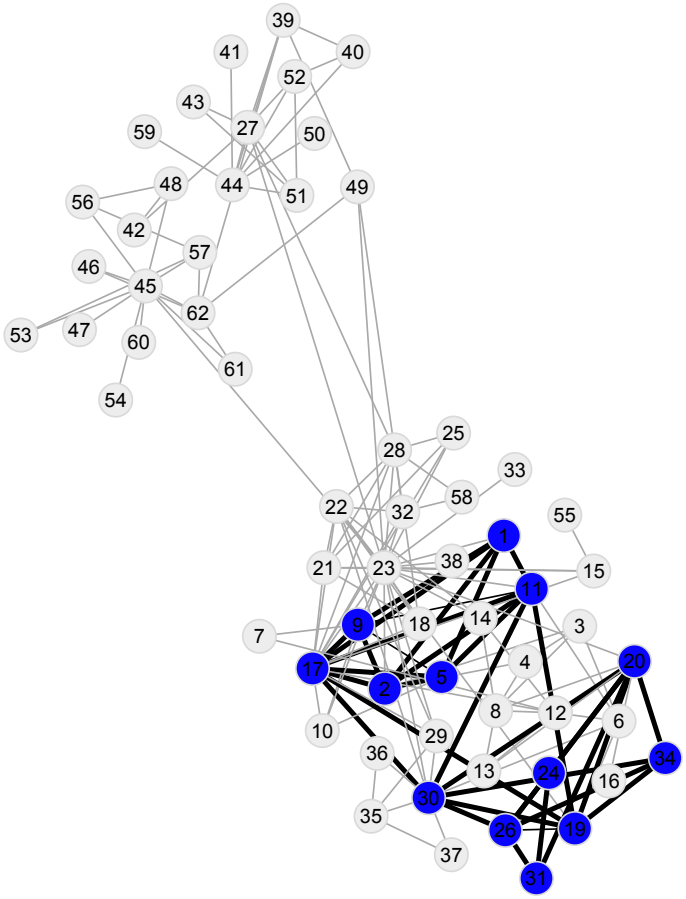


Figure 2.9: The largest local α-cluster for $\alpha = 0.6$ in the terrorist network. Nodes 1, 2, 11 were involved in the WTC North attack, 9,17 participated in the WTC South attack, 19,20,26,30,34 were involved in the Pentagon attack.

We use connected graphs for all computational experiments. If the original graph is disconnected, then the largest connected component is considered. Moreover, we make the network undirected if the original one is directed. The test instances contain 22 various social networks which include

- **monkey5**: The graph of interactions among a troop of monkeys, observed in the wild by Linda Wolfe as they sported by a river in Ocala, Florida. Two nodes are connected by an edge if a corresponding pair of monkeys were jointly

Table 2.10: Optimal solution and their number for real-life network instances used in the experiments.

			Optimal solutions, $\alpha = \ldots$					Number of solutions, $\alpha = \ldots$				
Graph	$\|V\|$	$\|E\|$	0.6	0.7	0.8	0.9	1	0.6	0.7	0.8	0.9	1
monkeys5	19	60	11	9	8	8	6	3	11	3	1	4
taro	22	39	6	4	3	3	3	4	10	10	10	10
strike	24	38	4	4	4	4	4	4	1	1	1	1
dining	26	42	4	3	3	3	3	2	5	5	5	5
high-tech	33	91	11	8	7	6	6	8	14	4	3	1
korea1	33	68	9	7	7	6	5	1	9	3	1	2
korea2	35	84	9	7	6	5	5	6	9	9	2	2
mexican	35	117	13	7	6	5	5	1	12	5	2	2
sawmill	36	62	3	3	3	3	3	18	18	18	18	18
tailorT1	39	158	13	10	9	8	6	26	16	1	1	7
tailorT2	39	223	21	15	12	8	7	1	4	2	46	4
flying	48	170	10	8	8	6	6	11	5	1	6	6
attiro	59	128	5	4	4	4	4	3	2	2	2	2
prison	67	142	6	5	5	5	5	3	5	5	1	1
huck	69	297	28	19	12	11	11	14	14	26	1	1
sanjuansur	74	144	5	4	4	4	4	5	3	3	3	3
jean	77	254	28	25	13	12	10	1	1	9	1	2
david	87	406	24	18	15	12	11	405	26	2	19	1
anna	138	493	29	19	15	12	11	≥ 984	10	340	8	1
lindenstrasse	232	303	4	3	3	3	3	5	12	12	12	12
dolphins	62	159	8	7	6	6	5	15	2	7	1	3
santa fe	118	200	8	6	6	5	5	2	14	1	5	5

present at the river at least 5 times during the observation period.

- **taro**: The graph represents the relation of gift-giving (taro exchange) among 22 households in a Papuan village.

- **dining**: Dining-table partners in a dormitory at a New York State Training School.

- **flying**: The graph represents 48 cadet pilots participated in a sociometric test at an US Army Air Forces flying school by 1943, administered by Leslie D. Zeleny. Two cadets are connected by an edge if one of them name the other as the person with whom he would like to fly.

- **mexican**: The network contains the core of the political elite in Mexico: the

Table 2.11: Computational time for the real-life instances used.

			Computational time (sec), $\alpha = \ldots$				
Graph	$\|V\|$	$\|E\|$	0.6	0.7	0.8	0.9	1
monkeys5	19	60	1.24	1.34	1.68	1.59	1.92
taro	22	39	1.12	0.77	0.90	0.82	1.09
strike	24	38	0.71	0.52	0.56	0.84	0.70
dining	26	42	0.14	0.16	0.16	0.18	0.17
high-tech	33	91	4.50	5.30	5.84	6.20	4.98
korea1	33	68	2.24	2.12	1.88	1.97	1.87
korea2	35	84	5.15	5.85	8.25	9.09	5.55
mexican	35	117	8.58	9.13	8.93	8.66	8.46
sawmill	36	62	2.47	2.49	2.59	2.52	2.41
tailorT1	39	158	17.40	17.34	18.30	18.50	17.49
tailorT2	39	223	20.17	29.49	35.17	33.01	21.19
flying	48	170	87.71	45.03	37.75	34.49	17.23
attiro	59	128	47.40	52.95	46.68	49.73	30.51
prison	67	142	24.21	22.13	28.35	17.14	21.73
huck	69	297	13.32	19.48	41.42	31.23	28.01
sanjuansur	74	144	95.68	59.76	67.62	57.48	52.87
jean	77	254	19.59	16.07	53.49	48.04	34.43
david	87	406	150.67	112.40	107.24	98.83	74.91
anna	138	493	952.76	7911.46	560.98	1094.70	495.33
lindenstrasse	232	303	21.03	23.67	24.01	23.70	23.64
dolphins	62	159	99.22	68.65	78.17	73.76	43.46
santa fe	118	200	59.46	67.73	61.72	59.52	382.06

presidents and their closest collaborators. In this network, edges represent significant political, kinship, friendship, or business ties.

- **prison**: The graph represents 67 prison inmates and is compiled by John Gagnon in 1950s. Two prisoners are connected by an edge if one of them name the other as his closest friend.

- **korea1, korea2**: The graphs are based on family planning discussions in Korea among 39 women. An edge between two women represents a family planning discussion among them.

- **attire, sanjualsur**: The graphs represent visiting ties among families in Attiro and San Juan Sur villages in Turrialba, Costa Rica, 1948. Each edge represents "frequent visits" from one family to another.

- **strike, sawmill, hi-tech, tailorT1, tailorT2**: Employee communication (strike, sawmill), friendships (hi-tech) and interactions (tailorT1, tailorT2) in various firms.

- **anna, david, huck, jean**: Graphs, where each node represents a book character and two nodes are connected by an edge if the corresponding characters encounter each other in the book (http://mat.gsia.cmu.edu/COLOR03/). The graphs for four classic works are considered: Tolstoy's Anna Karenina, Dicken's David Copperfield, Twain's Huckleberry Finn, and Hugo's Les Misérables.

- **lindenstrasse**: The data corresponds to a graph of the characters and their relations in the long-running German soap opera called 'Lindenstrasse'.

- **dolphins**: Dolphins network constructed by [43]. In the corresponding study, interactions of 62 dolphins were traced, with the edges describing the pairs of dolphins with higher than expected level of interaction between them.

- **santa fe**: The largest connected component of Santa Fe Institute (SFI) collaboration network [31]. There are 118 scientists and an edge is drawn if they coauthored one or more articles between 1999 and 2000.

The results are summarized in Table 2.10 and the corresponding computational times are displayed in Table 2.11. If there are multiple solutions for a given network and parameter α, we report the time required to obtain the first solution. The alternative solutions take much less time to compute since optimality is easy to verify and does not need to be proved any more (and proving optimality typically takes much longer than finding an optimal solution).

We can observe that for some of the larger instances, such as 'anna', the running time increases to several thousands of seconds. However, the relatively small number

Table 2.12: Optimal solutions and their number for the random network instances used.

			Optimal solutions, $\alpha = \ldots$					Number of solutions, $\alpha = \ldots$				
Graph	$\|V\|$	$\|E\|$	0.6	0.7	0.8	0.9	1	0.6	0.7	0.8	0.9	1
BA_50_2	50	97	4	3	3	3	3	23	20	20	20	20
BA_50_3	50	144	8	6	5	4	4	13	6	6	6	6
BA_50_4	50	190	11	8	7	6	5	8	5	5	2	3
BA_50_5	50	235	15	11	8	7	6	9	1	27	5	6
BA_100_2	100	197	4	3	3	3	3	46	24	24	24	24
BA_100_3	100	294	8	6	5	4	4	12	11	9	9	9
BA_100_4	100	390	12	8	7	6	5	5	10	7	4	4
BA_100_5	100	485	15	11	8	7	6	137	2	21	4	4
ER_50_100	50	100	4	3	3	3	3	5	14	14	14	14
ER_50_150	50	150	4	3	3	3	3	25	31	31	31	31
ER_50_200	50	200	7	5	5	4	4	10	1	1	4	4
ER_50_250	50	250	8	6	5	4	4	5	6	9	15	15
ER_100_200	100	200	3	3	3	3	3	6	6	6	6	6
ER_100_300	100	300	4	3	3	3	3	7	29	29	29	29
ER_100_400	100	400	6	4	4	4	4	2	1	1	1	1
ER_100_500	100	500	6	4	4	4	4	8	5	5	5	5

Table 2.13: Computational time for random instances.

			Computational time (sec), $\alpha = \ldots$				
Graph	$\|V\|$	$\|E\|$	0.6	0.7	0.8	0.9	1
BA_50_2	50	97	1.27	1.06	1.03	1.05	1.11
BA_50_3	50	144	9.99	9.48	11.94	11.24	6.57
BA_50_4	50	190	26.32	26.06	13.43	11.74	11.39
BA_50_5	50	235	93.10	55.01	32.60	36.05	38.05
BA_100_2	100	197	3.27	3.28	3.40	4.21	4.15
BA_100_3	100	294	111.50	96.11	216.26	78.68	112.66
BA_100_4	100	390	1540.01	1289.11	548.11	767.2	309.22
BA_100_5	100	485	14736.16	4371.98	2741.46	1985.02	871.53
ER_50_100	50	100	3.06	3.70	3.46	3.28	4.01
ER_50_150	50	150	10.08	13.68	11.84	5.46	6.93
ER_50_200	50	200	98.87	45.10	37.88	39.25	16.18
ER_50_250	50	250	401.78	225.93	197.84	120.68	48.73
ER_100_200	100	200	32.12	33.15	30.45	28.38	33.87
ER_100_300	100	300	118.78	97.16	86.90	90.34	98.85
ER_100_400	100	400	4212.58	1874.37	999.23	568.40	305.87
ER_100_500	100	500	6978.11	2751.40	1753.21	1242.32	805.18

of vertices and edge density allow us to solve these instances to optimality within a reasonable amount of time.

To explore the dependence of the running time on the instance size in a more

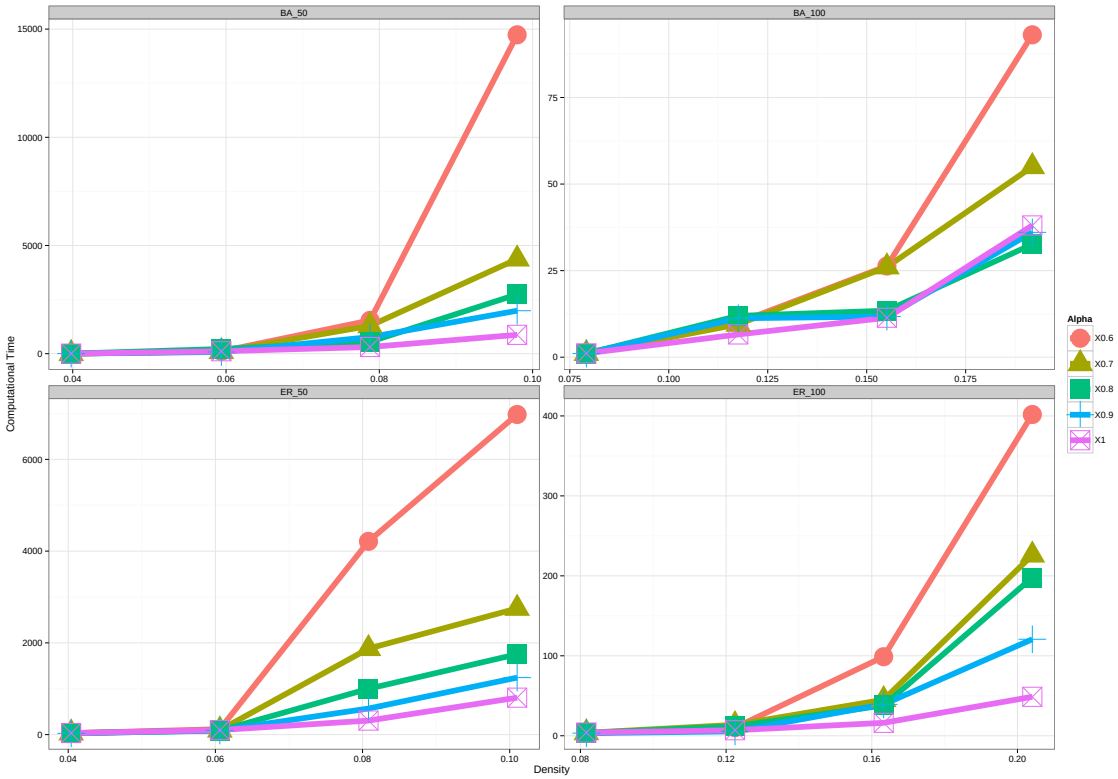


Figure 2.10: Graphical illustration of the dependence between the running time and edge density for the random graphs described in Tables 2.10-2.13.

systematic fashion, we conduct experiments with Barabási-Albert [11] and Erdös-Rényi [23] random graphs. Barabási-Albert model produces power-law random graphs, where the probability that a node has a degree k is proportional to $k^{-\beta}$ with $\beta \approx 3$. Power law structure has been observed in many real life complex systems, including social networks [46]. Erdös-Rényi model is given by a uniform random graph $G(n, m)$, where n is the number of vertices and m is the number of edges. The model assumes that any graph having n vertices and m edges has the same probability to occur.

The corresponding results are presented in Tables 2.12-2.13 and Figure 2.10. In these tables, 'BA_n_s' represents a Barabási-Albert random graph on n vertices

($n = 50$ and 100) for several values of the parameter s (s=2,3,4, and 5), where s is the number of edges added per node in a graph generating process, and 'ER_n_m' stands for a uniform random graph with n vertices and m edges. Figure 2.10 shows a significant increase in the running time with the increase in edge density for different values of α.

2.4 Local α-Clustering Algorithm

Dropping the connectivity requirement from the definition of an α-cluster may result in solutions corresponding to disjoint unions of α-clusters. These disjoint α-clusters can be used as seeds for clustering methods that partition the network into clusters. In this section, we discuss such a clustering algorithm and evaluate its performance on the four real-life networks used in the previous section.

Algorithm 1 Local α-clustering algorithm

1: Given a connected graph $G = (V, E)$ and α.
2: Find a maximum local α-cluster of G without the connectivity requirements, denote the corresponding induced subgraph by $G' = (V', E')$.
3: Let $V'_1, \ldots V'_k$ be the sets of nodes corresponding to the different connected components of G', where k is the number of connected components.
4: **while** $V \backslash (\cup_{j=1}^{k} V'_j) \neq \emptyset$ **do**
5: Find a node $i \in I$ and a cluster V'_j with the largest size of $N_G(i) \cap V'_j$.
6: $V'_j = V'_j \cup \{i\}$
7: **end while**
8: **return** clustering $V'_1, \ldots, V'_k$

The proposed approach is outlined in Algorithm 1. We proceed by computing a maximum local α-cluster with relaxed connectivity requirements, which typically results in several α-clusters $V'_1, \ldots, V'_k$ corresponding to different connected components in the induced subgraph. Each such α-cluster $V'_j, j = 1, \ldots, k$ forms an initial

seed. At each iteration of the algorithm, we pick an unassigned node i that has the largest number of neighbors in one of the clusters V_j' and assign i to V_j'. We apply the procedure recursively until all nodes are assigned to a cluster. In the experiments below we used $\alpha = 1$.

We first apply the proposed local α-clustering algorithm to the Zachary's karate club network. We demonstrate the results in Figure 2.11. In the figure, black edges

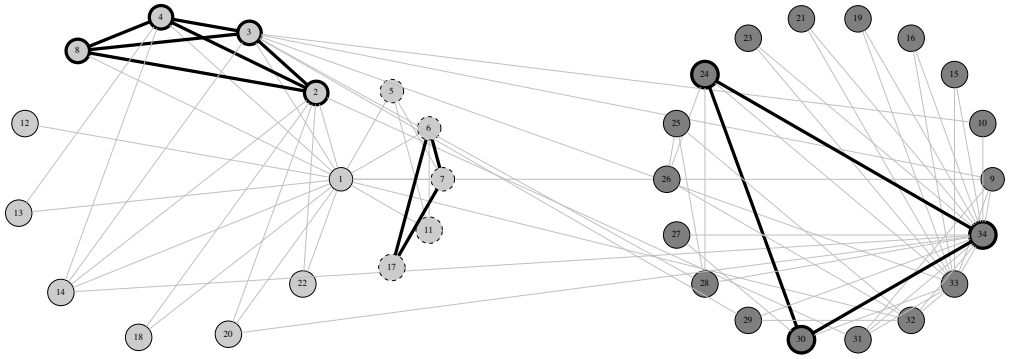


Figure 2.11: Local α-clustering for the karate example.

belong to the subgraph induced by the maximum relaxed α-cluster solution, and gray edges are outside of the subgraph. Our clustering algorithm finds three clusters, one of which coincides with the John's faction (dark-gray nodes). The other two clusters correspond to Mr. Hi's faction (light-gray nodes), with one of the clusters consisting of a group of 5 students that only interact with themselves and Mr. Hi (the corresponding nodes are shown as dashed circles) and the other cluster containing the remaining light-gray nodes (shown as solid-border circles). The clustering we obtained can be viewed as an enhancement of the results reported in the literature, since in addition to correctly identifying the two factions it singles out a clearly defined subgroup within one of the factions.

The second considered example deals with the football graph. The results are illustrated in Figure 2.12. In this figure, colors of edges have the same interpretation

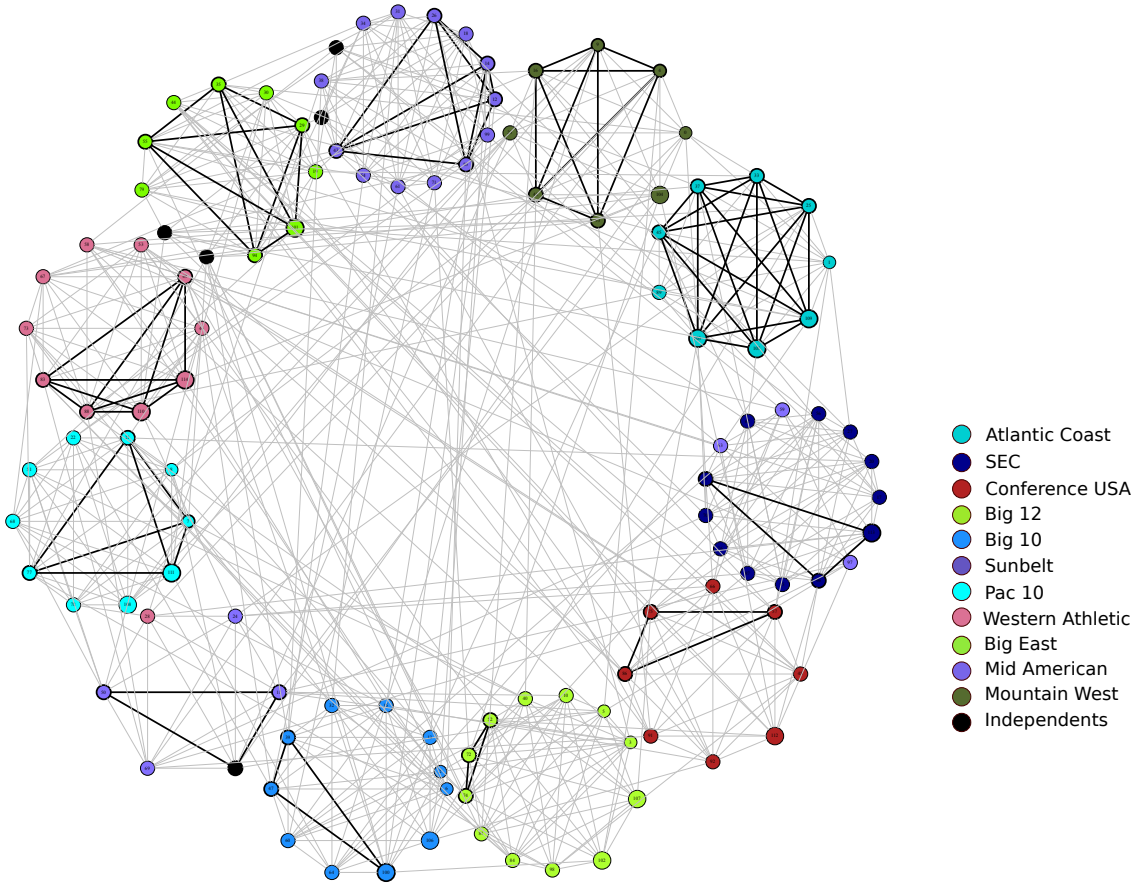


Figure 2.12: Local α-clustering for the football graph.

as in the karate example above. Our clustering algorithm finds 11 clusters, nearly coinciding with the actual conferences. The algorithm expectedly places the five independent teams by assigning them to the "closest" conferences. In addition, it misplaces four teams from Sunbelt Conference, which are assigned to the Western Athletic Conference cluster and SEC. This can be explained by observing that the

number of games the misplaced teams played against the other Sunbelt Conference teams and Western Athletic Conference/SEC teams was similar. The results we obtained are comparable with the results of [31].

Next, we apply our algorithm to the largest component of SFI collaboration network. We show the results in Figure 2.13.

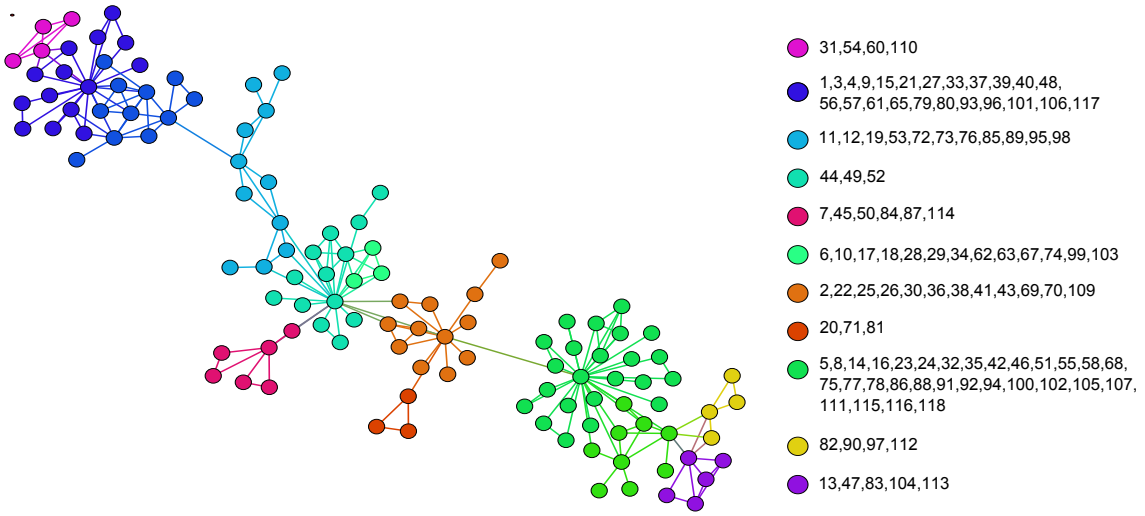


Figure 2.13: Local α-clustering for Santa Fe Institute largest component.

In this figure, vertices with different colors correspond to different clusters that algorithm identifies. At the coarse-level, our algorithm finds four clusters that group authors by their research interests. At a finer-level, our algorithm identifies subgroups of close collaborators within each research area.

The results of clustering the dolphins graph with our local α-clustering algorithm are shown in Figure 2.14. At the loose-level, our algorithm finds two main closely connected clusters. Only two dolphins, "Oscar" and "PL", are misplaced compared to the well-known clustering of the dolphins network [29]. At a more confined level,

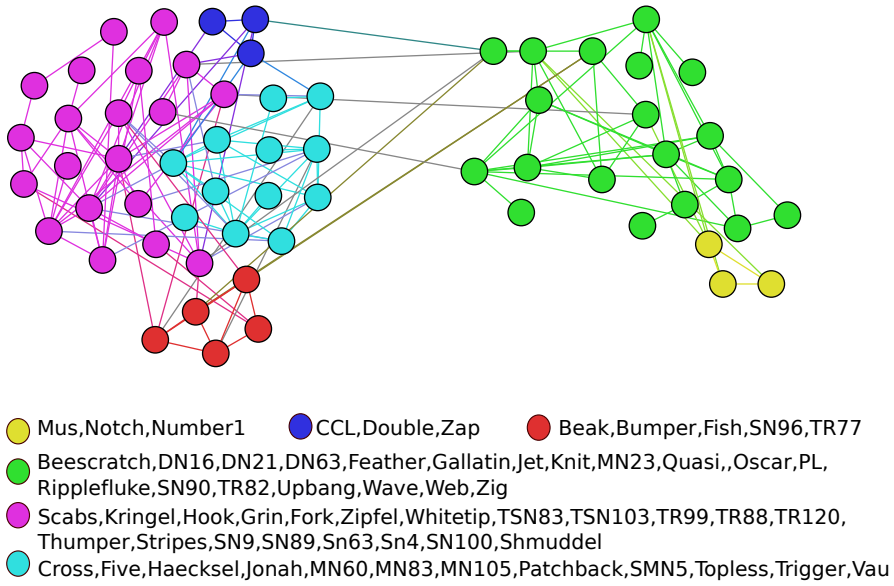


Figure 2.14: Local α-clustering for dolphins graph.

our algorithm recognizes more tightly-knit subgroups of dolphins within the larger clusters.

Finally, we apply our clustering algorithm to the Krebs's terrorist network, with the resulting clusters shown in Figure 2.15. The algorithm finds six clusters, three of which consist of the associates of the hijackers (with just one exception, hijacker node 22, included in one of the clusters), and the remaining three of which correspond to the actual hijackers. The three groups roughly correspond to the WTC, Pentagon, and Pennsylvania attacks, respectively.

2.5 Conclusion

To conclude, this section introduces novel clique relaxation models (i.e., α-clusters) and methods for finding cohesive subgroups in networks. We use local and global clustering coefficient metrics since these metrics are efficiently used to characterize real world networks such as social and biological networks that have small-world

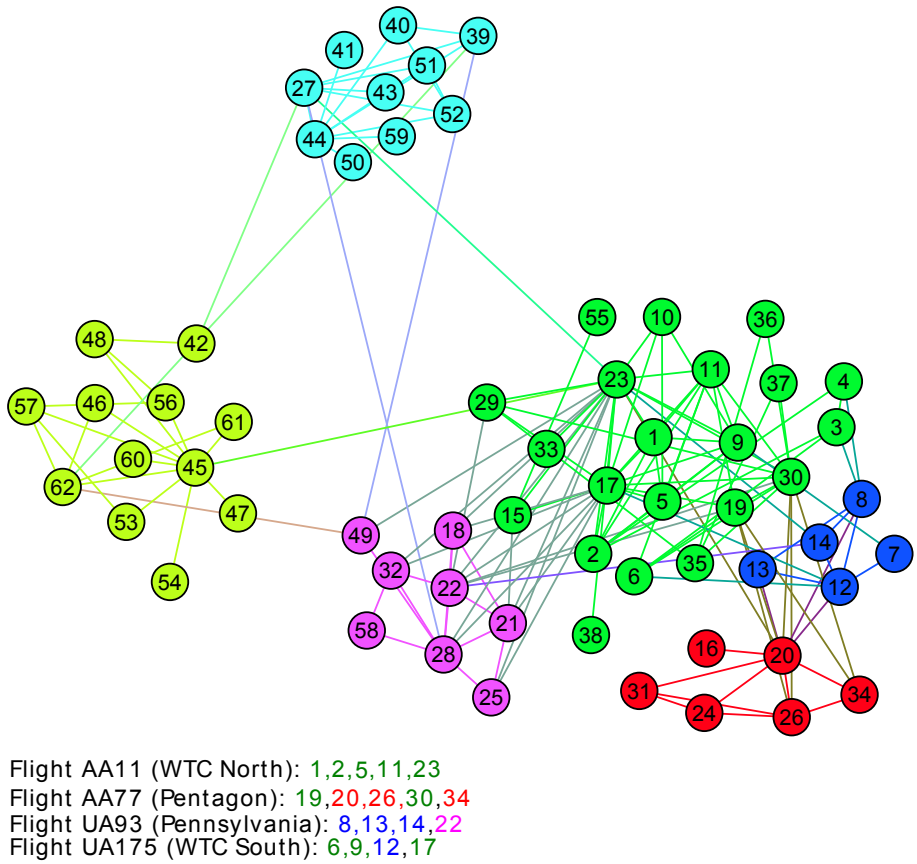


Figure 2.15: Local α-clustering for terrorist network compiled by Krebs.

properties. We also introduce a novel clustering algorithm based on α-cluster solution which effectively identifies the clusters without the common assumption of fixing the number of clusters a priori. We evaluate the correctness of this algorithm on the well-known real world social graphs, and obtain very comparable results to the ones reported in the literature. Additionally, to the best of our knowledge, we perform the first reported clustering analysis of the Kreb's terrorist network revealing the cohesive subgroups within the network.

3. INDEPENDENT UNION OF CLIQUES

In this section, we focus on a special case of α-cluster, namely *Independent Union of Cliques* (IUC), in which $\alpha = 1$ and no connectivity constraints are imposed. We observe that solutions of two classical problems in combinatorial optimization, maximum independent set and maximum clique, are feasible solutions for the IUC problem. Moreover, the solutions of these classical problems are lower bounds to the size of IUC solution. We show that the IUC problem is NP-Hard and we study the complexity results of various special graph types. We also develop two algorithms to find the exact solution based on *integer programming* and *combinatorial branch and bound*, respectively. Furthermore, we develop several approximate algorithms based on various heuristic techniques.

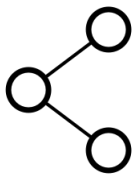
The remainder of this section is organized as follows. In the next section, we introduce the necessary definitions and study some basic structural properties of *IUC*. Section 3.2 reports complexity results on several graphs like planar graphs, unit disk graphs, and claw-free graphs. Section 3.3 presents both exact as well as approximate algorithms, and their experimental results on benchmark graph instances. This section is based on the working paper by Ertem et al. [26].

3.1 Definitions and Properties

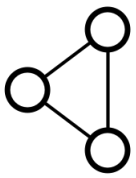
This section presents basic graph-theoretic definitions and notations used throughout the section. Let $G = (V, E)$ be a simple graph with set V of n vertices (nodes) and set E of edges (links), $E \subseteq \{\{i, j\} : i, j \in V\}$. If $\{i, j\} \in E$, vertices i and j are *adjacent* to each other. We can represent G using its adjacency matrix, whose elements $a_{ij}, i, j = 1, \ldots, n$ indicate whether there is an edge between nodes i and j. The *degree* $d_G(i)$ of vertex i is the number of neighbors of i in G.

A set of vertices $I \subseteq V$ is called *independent set* if there is no edge between any two vertices i, j, i.e. $(i, j) \notin E \; \forall i, j \in I$. A *clique* C is a set of vertices such that every pair of distinct vertices are adjacent in G. These two structures are closely related, indeed C is a clique if and only if C is an independent set in the complementary graph $\overline{G} = (V, \overline{E})$ where $\overline{E} = \{(i, j) | i, j \in V, i \neq j, (i, j) \notin E\}$. $\Delta(G[S])$ denotes the maximum degree value of a node in the induced subgraph.

The maximum cardinality of a clique in G is called *clique number* and is denoted by $\omega(G)$. The maximum independent set size of a given graph G is called the *independence number* of G, and denoted $\alpha(G)$. We define the size of the independent union of cliques size of a graph G as $\zeta(G)$.



(a) Open Triangle



(b) Closed triangle

Figure 3.1: Examples of an open triangle and a closed triangle.

An *open triangle* is an induced path of three vertices, where a direct edge is missing between a pair of nodes in the induced graph, as it is shown in Figure 3.1a. On the other hand, a closed triangle is a subgraph where all three pairs are adjacent in the induced graph, see Figure 3.1b.

Assume that an induced subgraph $G[C]$ is an *IUC*. Since it consists of connected components forming a clique, there are no open triangles. Given a subgraph without open triangles, each connected component of the subgraph must be a complete graph. Thus, it is an *IUC*. This fact leads us to the following property.

Property 1. *A set C is an* IUC *if and only if there is no open triangle in the induced subgraph $G[C]$.*

Proof. Let the set C be an *IUC*, so that it consists of union of independent cliques. That is, the vertices are directly connected within the clique, but there is no connection between the cliques. Therefore, there is no open triangle in the subgraph $G[C]$.

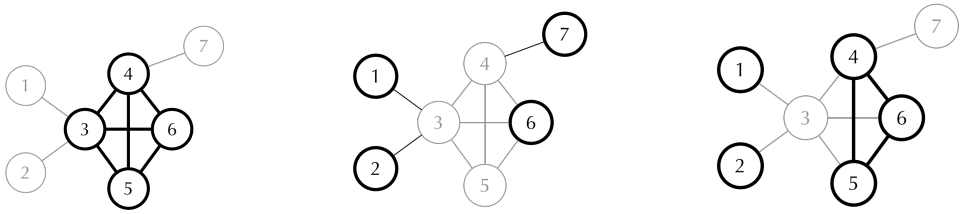
Conversely, let there be no open triangles in the induced graph $G[C]$, that is either neighbors of a node are all pairwise adjacent, resulting in a clique, or there are nodes with no neighbors, resulting in members of independent set. From the definition of IUC, nodes inducing a clique or members of independent set, imply a solution for IUC. Therefore, a set C is an *IUC* if and only if there is no open triangles in the induced subgraph $G[C]$. □

In Figure 3.2, we show maximum clique, maximum independent set and maximum *IUC* solutions for a given simple undirected graph $G = (V, E)$. In Figure 3.2a, the maximum clique in the graph G is shown. Note that this solution, consisting of four nodes, is a feasible IUC solution. In Figure 3.2b, the maximum independent set solution is shown, which also consists of four nodes and is a feasible solution for IUC. Finally, in Figure 3.2c, the maximum IUC solution is presented. Note that this is the optimal solution for IUC, which consists of five nodes, including nodes from both the maximum clique solution and the maximum independent set solution.

The following two properties of IUCs are easy to see.

Property 2. *Any independent set and any clique in G is a feasible solution to the IUC problem.*

Property 3. *Independent union of cliques is a structure that is hereditary on induced*



(a) Max. clique (b) Max. ind. set (c) Max. *IUC*

Figure 3.2: Maximum clique, maximum independent set and maximum independent union of cliques solutions of a given graph G.

subgraphs, i.e., any subset of a solution for independent union of cliques is also a solution for independent union of cliques.

Theorem 1 (Yannakakis, 1978)**.** *The problem of finding the largest-order induced subgraph not violating property π that is nontrivial, interesting and hereditary on induced subgraphs is NP-hard.*

The *maximum independent union of cliques problem* is a special case of *minimum node deletion problem*, similar to the maximum clique and maximum independent set problems. Minimum node deletion problem is aiming to find the minimum number of deletions that still satisfies the desired property. The maximum independent union of cliques problem seeks to find the largest order induced subgraph satisfying the independent union of cliques. Independent union of cliques has *hereditary* property, since the deletion of any subset of vertices in $G[S]$ still results in a graph whose set of vertices is an independent union of cliques. Independent union of cliques is *nontrivial*, such that a single vertex graph is a feasible independent union of cliques. It is an interesting problem because there are arbitrarily large graphs which satisfy the independent union of cliques property. This brings us to the Yannakakis complexity result, in Theorem 1 [64], and conclude that maximum independent union

of cliques problem is NP-hard.

Lemma 1 (Degree)**.** *If S is an independent union of cliques solution of G, and $\omega(G)$ is given as the size of maximum clique, $\Delta(G[S]) \leq \omega(G) - 1$*

Proof. Maximum degree of a node in an independent union of cliques solution would be $\omega(G) - 1$ if the maximum clique is part of the solution. Maximum degree of a node in the solution would be less than that if a strict subset of maximum clique is in the IUC solution. □

Proposition 1. *If the max IUC solution C, consists of a single clique only, then it is i) a maximum clique and ii) a maximum dominating clique.*

Proof. Let us assume that C is not a maximum clique. Then, adding another set of nodes to convert this clique in to a maximum clique will provide a larger size of IUC solution. Hence, this contradicts that C is the solution of max IUC. Given the max IUC solution, C, is a single clique, we show that it is a maximum dominating clique. Let us assume that it is not maximum dominating clique. This implies that, there exists at least one node n, not directly connected to any node in the maximum clique. Adding this node to the current IUC solution C, also forms a feasible IUC solution. Hence, this contradicts that C is the solution for the max IUC. □

Similarly, we can write a proposition for the independent set case.

Proposition 2. *If the maximum IUC solution consists of an independent set only, then it is a maximum independent set.*

Lemma 2. *If the maximum independent union of cliques solution is equal to k disjoint union of cliques, then complement graph of the solution is a complete k-partite graph.*

3.2 Complexity on Various Graphs

In this section, we analyze the computational complexity of IUC for several graph classes like planar graphs, unit disk graphs and claw-free graphs.

3.2.1 Planar Graphs

Problem: Planar 3-SAT with at most 3 occurrence per variable (3-PSAT$_{\bar{3}}$).

Input: A set of variables U and a collection of clauses $E = \{E_1, ..., E_m\}$ such that each clause contains at most three literals, each variable occurs at most three times, and the undirected graph $G = (M, N)$ is planar, where $M = U \cup E$ and $N = \{(u_i, E_j) | u_i \in E_j \text{ or } \bar{u}_i \in E_j\}$.

Output: A truth assignment for U satisfying E.

3-PSAT$_{\bar{3}}$ is shown to be NP-hard in [40]. We will show the problem to find $\zeta(G)$ is NP-hard even when G is a planar graph or a unit disk graph by reduction from 3-PSAT$_{\bar{3}}$. First, we need two lemmata about $\zeta(G)$ in path graphs and cycle graphs.

Lemma 3. *Given $G = P_n$ a path graph, $\zeta(G) = \lceil \frac{2}{3}n \rceil$.*

Proof. By definition of *IUC*, for any three consecutive vertices v_i, v_{i+1}, v_{i+2}, $i = 1, 2, ..., n-2$, at most two of them could be in an *IUC*, so $\zeta(G) \leq \lceil \frac{2}{3}n \rceil$. Let $S = \{v_i | i \not\equiv 2 (\text{mod } 3)\}$, then obviously $G[S]$ is an *IUC* and $|S| = \lceil \frac{2}{3}n \rceil$, so $\zeta(G) = \lceil \frac{2}{3}n \rceil$. □

Lemma 4. *Given $G = C_n$ a cycle graph and $n = 3m$ where m is an integer, then $\zeta(G) = 2m$ and the maximum* IUC *consists of either S_0, S_1 or S_2 where $S_k = \{v_i | i \equiv k \text{ or } k+1 (mod\ 3)\}$, $k = 0, 1, 2$.*

Proof. By definition of *IUC*, for any three consecutive vertices v_i, v_{i+1}, v_{i+2} (Let $v_0 = v_n$), at most two of them could be in an *IUC*, so $\zeta(G) \leq 2m$. Conversely, obviously

each of $G[S_0]$, $G[S_1]$ and $G[S_2]$ is an *IUC* and $|S_0| = |S_1| = |S_2| = 2m$, so $\zeta(G) = 2m$.

Meanwhile, suppose S is a maximum *IUC*. If there exist two consecutive vertices $v_i, v_{i+1} \notin S$, S is an *IUC* of P_{n-2}, so $|S| \leq \lceil \frac{2}{3}(n-2) \rceil = 2m-1$, which is a contradiction. If for some $v_i \in S$, $v_{i-1}, v_{i+1} \notin S$, S is an *IUC* of P_{n-3} together with an isolated vertex v_i, so $|S| \leq \lceil \frac{2}{3}(n-3) \rceil + 1 = 2m-1$, which is also a contradiction. Thus, it is easy to show S is either S_0, S_1 or S_2. □

Theorem 2. *The problem of finding $\zeta(G)$ for a planar graph G is NP-hard.*

Proof. We use the reduction from 3-PSAT$_{\bar{3}}$. Suppose the number of variable is q and number of clauses is m. In the reduction each variable u_i will be represented by a cycle graph $C^i_{r_i}$, where $r_i \equiv 0 (\text{mod } 3)$. By lemma 4, $\zeta(C^i_{r_i}) = \frac{2}{3} r_i$ and maximum *IUC* is made up by either S^i_0, S^i_1 or S^i_2. Call S^i_0 *null nodes*, S^i_1 *true nodes* and S^i_2 *false nodes*. Each clause E_j including three literals is represented in the reduction by a configuration of three vertices shown in Figure 3.3 (left). Let the three vertices be v^j_x,v^j_y and v^j_z, corresponding to literals x_j, y_j and z_j, respectively. v^j_x, for example, connects to exactly two consecutive vertices v^i_k and v^i_{k+1}, where i is such that $x_j \in \{u_i, \bar{u_i}\}$. Here if $x_j = u_i$, choose a suitable k such that $k \equiv 1 (\text{mod } 3)$; if $x_j = \bar{u_i}$, choose a suitable k such that $k \equiv 2 (\text{mod } 3)$. Similarly, each clause E_j including two literals is represented by a configuration of two vertices shown in Figure 3.3 (right). As the instance of 3-PSAT$_{\bar{3}}$ is planar, the resulting G is planar, and obviously the reduction is polynomial. Let

$$p = \sum_{i=1}^{q} \frac{2}{3} r_i + m,$$

we claim that E is satisfiable if and only if $\zeta(G) = p$.

First we show $\zeta(G) \leq p$. Let Z be a maximum *IUC*. If in configuration of some E_j, more than one of $v^j_x, v^j_y, v^j_z \in Z$, w.l.o.g., assume $v^j_x, v^j_y \in Z$, where v^j_x connects to $v^{i_1}_{k_1}$ and $v^{i_1}_{k_1+1}$, v^j_y connects to $v^{i_2}_{k_2}$ and $v^{i_2}_{k_2+1}$. By definition of *IUC*, $v^{i_1}_{k_1}, v^{i_1}_{k_1+1}, v^{i_2}_{k_2}, v^{i_2}_{k_2+1} \notin Z$.

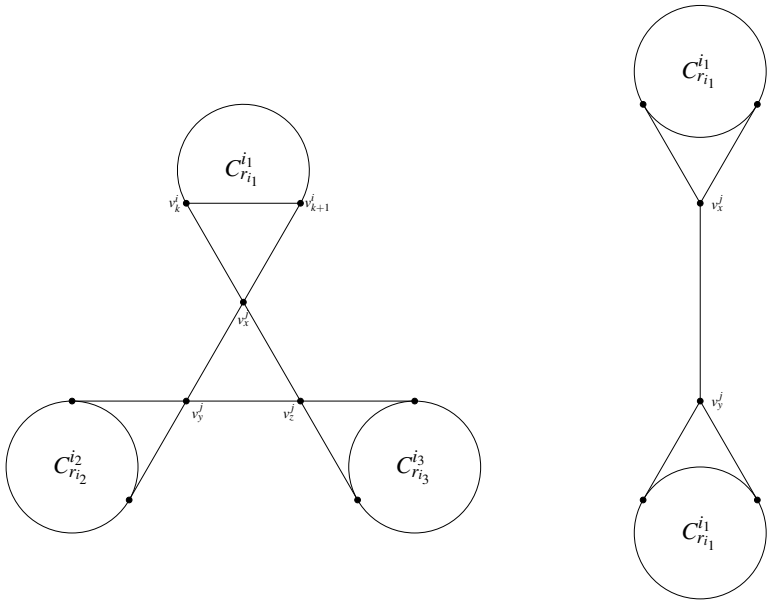


Figure 3.3: Clause configuration of three literals (left) and two literals (right).

On the other side, we consider $C^i_{r_i}$ for each variable u_i. Let

$$J_i = \{j | x_j \in \{u_i, \bar{u_i}\}, x_j \in E_j, v^j_x \in Z\}.$$

As u_i occurs at most three times in E, we have $0 \leq |J_i| \leq 3$.

- If $|J_i| = 0$, $|Z \cap V(C^i_{r_i})| \leq \frac{2}{3} r_i$ by lemma 4.
- If $|J_i| = 1$, as above, there exist $v^i_{k_1}, v^i_{k_1+1} \notin Z$, $G[Z \cap V(C^i_{r_i})]$ is an IUC in path graph $G[V(C^i_{r_i}) \setminus \{v^i_{k_1}, v^i_{k_1+1}\}] = P_{r_i-2}$, thus by lemma 3, $|Z \cap V(C^i_{r_i})| \leq \frac{2}{3} r_i - 1$.
- If $|J_i| = 2$, there exist $v^i_{k_1}$, $v^i_{k_1+1}$, $v^i_{k_2}$, $v^i_{k_2+1} \notin Z$, so $G[V(C^i_{r_i}) \cap Z]$ is made up of two path graphs $P_{k_2-k_1-2}$ and $P_{r_i+k_1-k_2-2}$. By lemma 3,

$$|Z \cap V(C^i_{r_i})| \leq \lceil \frac{2}{3}(k_2 - k_1 - 2) \rceil + \lceil \frac{2}{3}(r_i + k_1 - k_2 - 2) \rceil = \frac{2}{3} r_i - 2.$$

The last equation is derived by enumeration of $k_2 - k_1 (\text{mod } 3)$.

- If $|J_i| = 3$, similarly, $G[V(C^i_{r_i} \cap Z]$ is made up of three path graphs $P_{k_2-k_1-2}$, $P_{k_3-k_2-2}$ and $P_{r_i+k_1-k_3-2}$. Note $k_1, k_2, k_3 \equiv 1$ or 2 (mod 3) by construction of G, by enumerating all eight possible combination of k_1, k_2 and k_3, we derive $|Z \cap V(C^i_{r_i})| \leq \frac{2}{3} r_i - 3$.

Therefore,

$$|Z \cap V(C^i_{r_i})| \leq \frac{2}{3} r_i - |J_i|$$

in general. Let A be the total number of vertices in Z from the set of configurations of clauses that more than one vertex is included in Z and B be the total number of vertices from the set of configurations of clauses that at most one vertex is included in Z, then $A \leq \sum_{i=1}^{q} J_i$, $B \leq m$ and $B = m$ if and only if for a configuration of each clause there is exactly one vertex included in Z. Thus,

$$\begin{aligned} |Z| &= \sum_{i=1}^{q} |Z \cap V(C^i_{r_i})| + A + B \\ &\leq \sum_{i=1}^{q} \frac{2}{3}(r_i - |J_i|) + A + B \\ &\leq \sum_{i=1}^{q} \frac{2}{3} r_i + B \leq p. \end{aligned}$$

and $|Z| = p$ if and only if for each $C^i_{r_i}$ either one of S^i_0, S^i_1 or S^i_2 is included in Z and for each configuration of E_j exactly one of v^j_x, v^j_y and v^j_z is included in Z. So $\zeta(G) \leq p$.

Now assume that E is satisfied by a truth assignment τ. Construct Z as follows. For $i = 1, 2, ..., q$, if $\tau(u_i) =$ True, include in Z all true nodes in $C^i_{r_i}$; otherwise, include in Z all false nodes in $C^i_{r_i}$. For each $E_j = x_j \vee y_j \vee z_j$, assume, for example, $\tau(x_j) =$ True, then the two vertices v^i_k, v^i_{k+1} connected to v^j_x are already included in

Z while $v^i_{k-1}, v^i_{k+2} \notin Z$, so v^j_x can also be included in Z so that Z is an *IUC*. Thus $|Z| = \sum_{i=1}^{q} \frac{2}{3} r_i + m = p$. So $\zeta(G) = p$.

To prove the converse, if $\zeta(G) = p$, let Z be an *IUC* that $|Z| = p$. Then for each $C^i_{r_i}$ either one of S^i_0, S^i_1 or S^i_2 is included in Z and for each configuration of E_j exact one of v^j_x, v^j_y and v^j_z is included in Z. Construct a truth assignment τ as follows. For each u_i, if corresponding $S^i_1 \subseteq Z$, $\tau(u_i) =$ True; otherwise $\tau(u_i) =$ False (In fact, $\tau(u_i)$ is arbitrary if S^i_0 is in Z). Note for each clause E_j and the corresponding configuration, w.l.o.g, assume $v^j_x \in Z$, consider the two vertices v^i_k, v^i_{k+1} connected to v^j_x. At least one of v^i_k, v^i_{k+1} belongs to Z no matter which one of S^i_0, S^i_1 or S^i_2 is included, so $v^i_k, v^i_{k+1} \in Z$ in order to keep Z an *IUC*. If $x^j = u_i$, $k \equiv 1 (\text{mod } 3)$ and S^i_1 is included; if $x^j = \bar{u_i}$, $k \equiv 2 (\text{mod } 3)$ and S^i_2 is included. Therefore, $\tau(x_j) =$ True. So every E_j is satisfied and τ is really a truth assignment, E is satisfiable. This completes the proof. □

Remark 1. *To prove Theorem 2, it is enough to make $r_i = 18$ for each $i = 1, ..., q$.*

3.2.2 Unit Disk Graphs

Corollary 1. *The problem of finding $\zeta(G)$ on a unit disk graph G is NP-hard.*

Proof. The reduction in Theorem 2 could be easily transformed to UDG context as every $C^i_{r_i}$ can obviously be represented by a set of unit disk graphs and configuration of every clause can be represented by unit disk graphs shown in Figure 3.4. It is easy to verify that the reduction is polynomial. So the statement is true. □

3.2.3 Claw-Free Graphs

Corollary 2. *The problem to find $\zeta(G)$ when G is a claw-free graph is NP-hard.*

Proof. We still use the reduction from an instance of 3-PSAT$_{\bar{3}}$ to a graph G. Suppose the number of variables is q while number of clauses is m (in fact, we can use 3-SAT$_{\bar{3}}$,

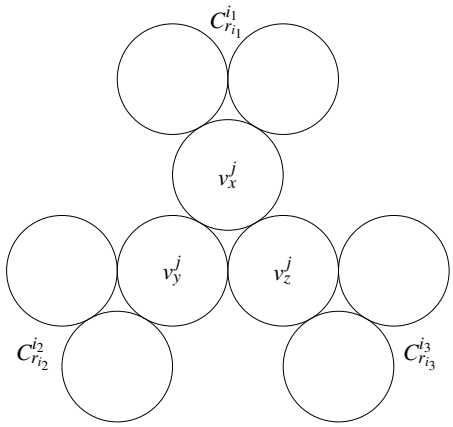


Figure 3.4: Clause configuration represented by unit disk graphs.

which does not need the instance to be planar). In the reduction each variable u_i will be represented by a gadget G_i shown in Figure 3.5. G_i contains an inner cycle and an outer cycle. Note that an *IUC* has hereditary property (Property 3), and by Lemma 4 we know the maximum *IUC* restricted to either inner or outer cycle is made of vertices labeled 1 and 2, or 2 and 3, or 1 and 3, thus, it is easy to know the only maximum *IUCs* in G_i consists of vertices labeled 1 and 2, or 2 and 3, or 1 and 3. Call them S^i_0, S^i_1 and S^i_2, respectively, and call S^i_0 *null nodes*, S^i_1 *true nodes* and S^i_2 *false nodes*. Each clause E_j including three literals is represented by a configuration shown in Figure 3.6. Here v^j_x connects to exactly four vertices v^i_a,v^i_b,v^i_c and v^i_b that form a clique, where i is such that $x_j \in \{u_i, \bar{u}_i\}$. Here, if $x_j = u_i$, let $v^i_a, v^i_b, v^i_c, v^i_d \in S^i_1$; if $x_j = \bar{u}_i$, let $v^i_a, v^i_b, v^i_c, v^i_d \in S^i_2$. Similarly, each clause E_j including two literals is represented by a configuration of two vertices. Obviously, this reduction is polynomial and by checking every vertex, it is easy to show G is claw-free.

Let $p = 8q + m$, we claim that E is satisfiable if and only if $\zeta(G) = p$.

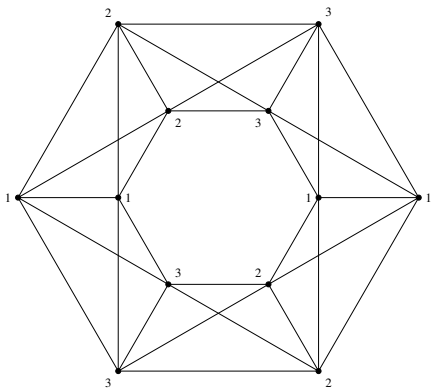


Figure 3.5: A variable gadget.

In fact, by the same arguments as in Theorem 2, we get $\zeta(G) \leq p$ and $\zeta(G) = p$ if and only if for each G_i either one of S_0^i, S_1^i or S_2^i is included in the maximum IUC and for each configuration of E_j exactly one of v_x^j, v_y^j and v_z^j is included in that IUC.

If E is satisfied by a truth assignment τ, construct the maximum IUC for $G[Z]$ as follows. For $i = 1, 2, ..., q$, if $\tau(u_i) =$ True, include in Z all true nodes in G_i; otherwise, include in Z all false nodes in G_i. For each $E_j = x_j \vee y_j \vee z_j$, assume, for example, $\tau(x_j) =$ True, then the four adjacent vertices $v_a^i, v_b^i, v_c^i, v_d^i$ are already included in Z, so v_x^j can also be included in Z, making Z an IUC. Thus, $|Z| = \sum_{i=1}^{q} 8 + m = p$. So, $\zeta(G) = p$.

Conversely, if $\zeta(G) = p$ and Z is a maximum IUC, construct a truth assignment τ as follows. For each u_i, if corresponding $S_1^i \subseteq Z$, $\tau(u_i) =$ True; otherwise $\tau(u_i) =$ False. By the same argument as in Theorem 2, every E_j is satisfied and τ is really a truth assignment, and thus E is satisfiable. This completes the proof. □

Proposition 3. *The problem of finding $\zeta(G)$ when G is a bipartite graph is also NP-hard.*

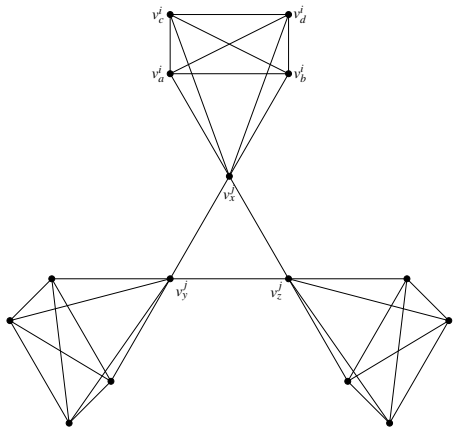


Figure 3.6: A clause configuration.

Proof. As a clique in a bipartite graph is either a single vertex or a edge, an IUC in a bipartite graph is exactly a 1-dependent set, that is a set of independent vertices and edges. Thus, the problem of finding $\zeta(G)$ is the problem of finding the maximum 1-dependent set in a bipartite graph, which is shown to be NP-hard in [20]. □

3.3 Methodology

The maximum IUC problem consists of finding the largest set of nodes that satisfies the independent union of cliques properties. This problem can be formulated as a binary integer linear program.

3.3.1 Integer Programming Formulation

Decision variables: Let us define x_i, $i \in V$ as a binary decision variable to indicate whether node i is in the independent union of cliques set C, that is,

$$x_i = \begin{cases} 1, & \text{if node } i \text{ is in the independent union of clique set } C; \\ 0, & \text{otherwise.} \end{cases}$$

Our objective is to maximize the number of vertices in the independent union of clique set C, hence the objective is

$$\text{maximize} \sum_{i \in V} x_i. \tag{3.1}$$

Independent union of cliques constraints: We formulate the so-called *angle constraints* as follows:

$$x_i + x_j + x_k \leq 2 \quad \forall i, j, k \in V \text{ such that } a_{ij} + a_{jk} + a_{ik} = 2 \tag{3.2}$$

Note that this constraint is used for all open triangles in the graph.

Then a binary integer programming formulation can be written as:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in V} x_i \\ \text{subject to} \quad & x_i + x_j + x_k \leq 2 \quad \forall i, j, k \in V \text{ such that } a_{ij} + a_{jk} + a_{ik} = 2 \\ & x_i \in \{0, 1\}, \forall i \in V. \end{aligned}$$

Another alternative formulation is if we replace x_i with $y_i = 1 - x_i$, then our mathematical model can be represented as:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in V} y_i \\ \text{subject to} \quad & y_i + y_j + y_k \geq 1 \quad \forall i, j, k \in V \text{ such that } a_{ij} + a_{jk} + a_{ik} = 2 \\ & y_i \in \{0, 1\}, \forall i \in V. \end{aligned}$$

This formulation is very similar to the set covering problem, where the sets to be covered are given by all triples of vertices inducing paths of length 2 in the graph.

The proposed integer programming formulations allow us to use standard solvers

to find the maximum IUC solutions on the moderate-size graphs. To work on the larger graph instances, we introduce a new algorithm based on a combinatorial branch and bound approach.

3.3.2 Branch and Bound Approach

With the realistic bounds provided by the static variable ordering in Russian Doll Search (RDS) approach [59], the algorithmic framework provided by [56] serves as a simple and effective tool to develop search algorithms to find exact solutions for problems with hereditary structures. Since the maximum IUC problem has hereditary properties, we develop an exact algorithm by applying this framework.

Our algorithm proceeds as follows. First, we impose an order on vertices based on their degree in ascending order. We start with an empty set solution, $NewSet$. Then, we go through each vertex v_i in the ordered list and compute a maximum IUC in the candidate set given by $\{v_i, \ldots, v_n\}$. When we go through all the vertices in the ordered list, we find the maximum IUC in G. The procedure is outlined in Algorithm 2. Here by $w(S)$ for a subset of vertices S we mean the cardinality of the set S. The verification procedure in Algorithm 3 verifies whether adding a new vertex to a given solution results in an IUC or not.

Two pruning points in finding the maximum IUC solution speed up the algorithm. First, we can prune when we know that the total size of the candidate set and the new set is smaller than the size of the current solution (i.e., we are guaranteed that even if we include all the nodes from the candidate set and the new set, the potential solution would be less than the current best solution.) Second, given that $\mu(i)$ is the maximum IUC of the induced graph G with nodes v_n to v_i in $V_{ordered}$, we can prune our search when we know that the best solution in G is less in cardinality than the current best solution.

Algorithm 2 *IUC* Algorithm.

```
1: Given G = (V, E)
2: V_ordered =Order(V)
3: NewSet = ∅, max := 0
4: for i := n down to 1 do
5:     CandidateSet := {v_i, v_{i+1}, ..., v_n}
6:     max := FIND_IUC(CandidateSet, NewSet, G, max)
7: end for
8: return max

9: function FIND_IUC(CandidateSet, NewSet, G, max)
10:    if CandidateSet = ∅ then
11:        if w(NewSet) > max then
12:            max:=w(NewSet)
13:        end if
14:        return
15:    end if
16:    while CandidateSet ≠ ∅ do
17:        if w(CandidateSet) + w(NewSet) < max then
18:            return
19:        end if
20:        i := min{j : v_j ∈ CandidateSet}
21:        if μ(i) + w(NewSet) < max then
22:            return
23:        end if
24:        CandidateSet := CandidateSet \ {v_i}
25:        NewSet' := NewSet ∪ {v_i}
26:        CandidateSet' := ∅
27:        for v ∈ CandidateSet do
28:            if Is_IUC(NewSet', v, G) then
29:                CandidateSet' := CandidateSet' ∪ {v}
30:            end if
31:        end for
32:        Find_IUC(CandidateSet', NewSet', G, max)
33:    end while
34:    return max
35: end function
```

Algorithm 3 *IUC* verification procedure.

```
1: function Is_IUC(CandidateSet, new_node, G)
2:     Cliques = FindCliques(CandidateSet, G)
3:     cliqueNum = 0
4:     for C ∈ Cliques do
5:         if G[C ∪ {new_node}] is connected but not a clique then
6:             return False
7:         end if
8:         if C ∪ {new_node} is a clique then
9:             cliqueNum = cliqueNum + 1
10:        end if
11:    end for
12:    if cliqueNum > 1 then
13:        return False
14:    else
15:        return True
16:    end if
17: end function

18: function FindCliques(CandidateSet, G)
19:    CliqueList := ∅
20:    for v ∈ CandidateSet do
21:        for clique ∈ CliqueList do
22:            if v is a neighbor of any vertex in clique then
23:                clique := clique ∪ {v}
24:                break
25:            end if
26:        end for
27:        new_clique = {v}
28:        CliqueList := CliqueList ∪ new_clique
29:    end for
30: end function
```

3.3.3 Results of Computational Experiments

In this section, we present computational results of the two proposed approaches for the maximum IUC problem on benchmark graph instances. All numerical computations were performed on a *Dell Precision WorkStation T7500*® computer with

Table 3.1: Computational results of solving the maximum *IUC* problem using branch and bound algorithm and integer programming formulations on DIMACS instances.

Graph	$\|V\|$	$\|E\|$	d	$\omega(G)$	IUC size	Solution time (sec)	
						B&B	IP
johnson8-2-4	28	210	0.55	4	7	0.80	1.30
hamming6-2	64	1824	0.90	32	32	2.22	59.31
johnson8-4-4	70	1855	0.76	14	14	3.66	81.07
MANN_a9	45	918	0.92	16	16	13.05	5.97
hamming6-4	64	704	0.35	4	16	884.74	80.15
johnson16-2-4	120	5460	0.76	8	15	10247.80	1714.56
keller4	171	9435	0.65	11	15	15054.90	49038.00
brock200_3	200	12048	0.61	15	15	53433.80	518413.40
brock200_4	200	13089	0.66	17	17	43236.90	Memory
brock200_1	200	14834	0.75	21	21	619853.00	Memory
brock200_2	200	9876	0.50	12	15	715634.00	Memory
c-fat200_1	200	1534	0.08	12	TiLim	>1390873	Memory
c-fat200_2	200	3235	0.16	24	TiLim	> 2667828	Memory
p_hat300-1	300	10933	0.24	8	TiLim	>5000000	Memory
MANN_a27	378	70551	0.99	120	>117	>300000	Memory
hamming10-2	1024	518656	0.99	512	512	494260	Memory

eight 2.40 GHz Intel Xeon® processors and 12 GB RAM.

In Table 3.1, we show the run time and solution size comparisons for integer programming (IP) and branch and bound (B&B) methods for several DIMACS instances. We denote the number of nodes by $|V|$, the edge number by $|E|$, the edge density by d, and the clique number by $\omega(G)$. For the small-size instances both IP and B&B methods work well in terms of the solution time. For large-size instances, the IP approach requires more memory than B&B. Hence, B&B approach can find solutions on the larger graph instances, whereas IP approach failed to do so. Although better than IP, B&B approach hit time limits on certain benchmark instances. Hence, we also develop faster heuristic algorithms that find approximate

Table 3.2: Solutions of the maximum *IUC* problem on selected DIMACS instances.

Graph	IUC size	Solution
johnson8-2-4	7	{22} {23} {24} {25} {26} {27} {28}
hamming6-2	32	{2,3,5,8,9,12,14,15,17,20,22,23,26,27,29,32,33, 36,38,39,42,43,45,48,50,51,53,56,57,60,62,63}
johnson8-4-4	14	{5,6,13,17,21,31,33,38,40,50,54,58,65,66}
MANN_a9	16	{4,5,6,7,9,10,17,19,22,25,28,31,34,37,40,43}
hamming6-4	16	{49,64} {50,63} {51,62} {52,61} {53,60} {54,59} {55,58} {56,57}
johnson16-2-4	15	{2,3,119} {106} {107} {108} {109} {110} {111} {112} {113} {114} {115} {116} {117} {118} {120}
keller4	15	{112} {113} {116} {117} {128} {129} {132} {133} {158} {159} {162} {163} {168} {169} {171}
brock200_3	15	{12,29,36,38,58,84,97,98,104,118,130,144,158,173,178}
brock200_4	17	{12,19,28,29,38,54,65,71,79,93,117,127,139, 161,165,186,192}
brock200_1	21	{4,26,32,41,46,48,83,100,103,104,107, 120,122,132,137,138,144,175,180,191,199}
brock200_2	15	{22} {37,188} {42,100,110,180,185} {92} {111,116} {140} {145,170} {193}
hamming10-2	512	clique of cardinality 512

solutions.

In Table 3.2, we present the actual solutions for the considered graphs. Vertices belonging to the same clique are placed within the same braces ("{ }"). In instances like hamming6-2, johnson8-4-4, MANN_a9, brock200_1, brock200_3, brock200_4 and hamming10-2, IUC solution consists of the maximum clique solution. In instances like johnson8-2-4 and keller4, IUC solution consists of the maximum independent set solution and for the rest of the instances IUC solution consists of several cliques.

3.3.4 Heuristics

Since the maximum *IUC* problem is NP-hard even for graphs with seemingly simple structure, we propose heuristic based approaches to find approximate solutions. We develop two efficient greedy heuristic methods. In the first heuristic, we start with finding a set of maximal cliques in G by using the approximate algorithm provided in [17]. We include the largest maximal clique, C, with the smallest total degree in our current *IUC* solution. Here, we define total degree of a clique as the sum of degrees (i.e., degree in G) of each node in the clique. Then, we remove all nodes in C with their corresponding neighbors from G. Finally, we repeat this process until there is no node left in G. Algorithm 4 shows the details of the first heuristic approach.

Algorithm 4 Maximum *IUC* Heuristic Algorithm 1.

1: **Initialization:** Set $S = \emptyset$.
2: **if** G has no vertices **then**
3: **return** S.
4: **end if**
5: Find a set of maximal cliques C.
6: Find the largest clique $c \in C$ with smallest total degree and add c to S.
7: Remove all nodes in c and their neighbors from G and return to Step 2.
8: **return** S.

In the second heuristic method, we first find a greedy independent set I in G. To find such a set, we simply start with adding the minimum degree node in G to I; we remove that node with all its neighbors from G. We continue until G has no nodes left.

Then we find a set of maximal cliques C in G by using the algorithm provided by [17] as in our first heuristic. We update our solution if any $c \in C$, improves our

current solution. We go through these cliques in C in the order of largest to smallest. Algorithm 5 shows the details of the second heuristic approach.

Algorithm 5 Maximum *IUC* Heuristic Algorithm 2.

Initialization: Set current solution S, Potential solution $PS = \emptyset$.
$S =$ FIND-APP-INDEPENDENT-SET(G)
Find a set of maximal cliques C.
For each $c \in C$, let $N_c = \{\text{all neighbors of } c \in G\}$ and $PS = \{S \cup c\} \setminus N_c$
if $|PS| > |S|$ **then** update S = PS
end if
return S.

function FIND-APP-INDEPENDENT-SET(G)
    Initialization: $I = \emptyset$
    while G has vertices **do**
        Find node n, that has the minimum degree in G
        $I = I \cup \{n\}$
        Remove n and its neighbors from G
    end while
    return I
end function

In Table 3.3 and Table 3.4, we present the results of Heuristic 1 and Heuristic 2 along with the branch and bound results on several DIMACS graph instances and social networks. As expected, heuristic approaches run much faster than IP and B&B, however, they find approximate solutions as opposed to exact solutions. With our heuristic approaches, we are able to approximately solve several DIMACS instances where B&B algorithm hits the time limits, such as c-fat and p_hat graphs. We observe that for relatively low-density (e.g., less than 0.16 in our experiments) instances Heuristic 1 provides much closer solutions to the optimal solution than Heuristic 2, whereas in comparatively high-density instances Heuristic 2 provides much closer solutions to the optimal solution.

Table 3.3: Computational results of solving the maximum *IUC* problem using heuristics on DIMACS instances.

Graph	$\|V\|$	$\|E\|$	d	IUC with B&B		Heuristic 1		Heuristic 2	
				Size	Time	Size	Time	Size	Time
johnson8-2-4	28	210	0.55	7	0.80	4	0.005	**7***	0.007
MANN_a9	45	918	0.92	16	13.05	**16**	0.015	**16***	0.016
johnson16-2-4	120	5460	0.76	15	10247.80	8	0.097	**15***	0.132
keller4	171	9435	0.65	15	15054.90	10	0.171	**15***	0.253
hamming6-2	64	1824	0.90	32	2.22	**22**	0.027	**22**	0.036
johnson8-4-4	70	1855	0.76	14	3.66	**8**	0.029	**8**	0.042
brock200_3	200	12048	0.61	15	53433.80	**11**	0.245	**11**	0.320
brock200_4	200	13089	0.66	17	43236.90	**11**	0.238	**11**	0.326
brock200_1	200	14834	0.75	21	619853.00	**13**	0.262	**13**	0.334
hamming6-4	64	704	0.35	16	884.74	8	0.075	**12**	0.045
brock200_2	200	9876	0.50	15	715634.00	8	0.241	**9**	0.326
hamming10-2	1024	518656	0.99	512	494260	Memory		**301**	63.175
MANN_a27	378	70551	0.99		TiLim	**120**	4.479	**120**	4.529
c-fat500_1	500	4459	0.04		TiLim	**320**	10.615	276	1.851
c-fat500_2	500	9139	0.07		TiLim	**300**	3.758	247	1.205
c-fat200_1	200	1534	0.08		TiLim	**119**	0.552	104	0.169
c-fat200_2	200	3235	0.16		TiLim	**134**	0.300	110	0.170
p_hat300-1	300	10933	0.24		TiLim	24	0.816	**37**	0.950
p_hat300-2	300	21928	0.49		TiLim	13	0.551	**26**	0.713
p_hat1000-1	1000	122253	0.24		TiLim	26	20.752	**70**	22.314
p_hat700-1	700	60999	0.25		TiLim	32	6.687	**60**	7.311

As a third heuristic, we can use both Heuristic 1 and Heuristic 2 and report the the best solution of the two. Since both heuristics have relatively quick run time, this hybrid approach would give the best of both approaches with a slight increase in run time.

3.4 Conclusion

In this section, we introduce a novel mathematical model called *independent union of cliques* which is similar in nature to two classical problems in combinatorial optimization, the maximum clique and the maximum independent set problems. We

Table 3.4: Computational results of the maximum *IUC* problem using heuristics on social network instances.

Graph	$\|V\|$	$\|E\|$	IUC with IP		Heuristic 1		Heuristic 2	
			Size	Time	Size	Time	Size	Time
Karate	34	78	23	0.37	18	0.021	21	0.011
Dolphins	62	159	36	0.39	26	0.095	34	0.038
Kreb's	62	153	39	0.25	31	0.107	37	0.036
Les miserables	77	2148	61	0.11	53	0.350	54	0.100
SantaFe Collaboration	118	200	96	9.60	83	1.816	89	0.190
Football	115	613	47	7876.49	38	0.230	33	0.126

explore structural properties of this problem. We examine the complexity of finding the exact solution and use integer programming and branch-and-bound methods, as well as several heuristic approaches to solve the problem.

4. NETWORK ANALYSIS OF LARGE SCALE BRAIN NETWORKS

In this section, we explore a unique and novel experimental data set about animal brains provided by the Texas A&M Institute for Preclinical Studies (TIPS) in order to understand the potential changes in animal brains after a certain trauma. In particular, the available data includes fMRI measurements for several pig brains before and after a concussion-inducing blast. We represent these fMRI measurements as a network and perform graph theoretical analysis of these networks. Specifically, using graph structural concepts such as clustering coefficient, centrality and degree distribution; as well as clustering based on clique relaxations, we analyze the effect of concussion on animal brains.

The remainder of this section is organized as follows. First, we introduce the necessary graph structural definitions and background in Section 4.1. Second, we describe the experimental design and the workflow to convert raw fMRI brain data into a network in Section 4.2. We present the summary of our results in Section 4.3. Finally, the section concludes with a summary of findings and suggestions for future research in Section 4.4. This section is based on the working paper by Ertem et al. [24].

4.1 Definitions and Background

In this section, we provide the definitions and the background for the graph theoretic concepts that we utilize for the rest of this section. First, we describe the graph structural concepts. Second, we explain clustering as an analysis tool to identify a partition of the network into cohesive subgroups.

4.1.1 Graph Structural Concepts

In this section, we provide definitions of the basic graph-theoretic concepts used. We consider a simple graph $G = (V, E)$, where V is the set of $|V| = n$ nodes and E is the set of $|E| = m$ edges corresponding to pairs of nodes. The *edge density* $\rho(G)$ of G is defined as the ratio of the number of existing edges to the number of all possible edges, that is $\rho(G) = 2m/(n(n-1))$. Nodes u and v connected by an edge are called *adjacent* or *neighbors*, denoted by $\{u, v\} \in E$. The neighborhood $N_G(v)$ of v in G is the set of all neighbors of v, i.e., $N_G(v) = \{u \in V : \{u, v\} \in E\}$. The degree $d_G(v)$ of a node v is the number of neighbors that v has in G, $d_G(v) = |N_G(v)|$.

A *path* of length r between nodes u and v in G is defined by an alternating sequence of distinct nodes and edges $u \equiv v_0, e_0, v_1, e_1, \ldots, v_{r-1}, e_{r-1}, v_r \equiv v$ such that $e_i = \{v_i, v_{i+1}\} \in E$ for all $1 \leq i \leq r-1$. Two nodes are connected in G if there exists a path between them in G. A graph is connected if all its nodes are pairwise connected; it is disconnected otherwise. The distance $d_G(u, v)$ between a pair of connected nodes u and v in G is the shortest length of a path connecting them.

Clustering coefficient is a measure of how nodes in a graph are clustered together. We distinguish between the global and local clustering coefficients. The global clustering coefficient $\mathcal{C}$ of G can be thought of as the probability that two randomly chosen neighbors of an arbitrary node of degree at least 2 are adjacent to each other. In other words, it is the proportion of number of closed triplets to the number of connected triplets of vertices. It can be expressed mathematically as follows:

$$\mathcal{C} = \frac{\sum\limits_{i \in V} \sum\limits_{j,k \in N_G(i), j<k} a_{jk}}{\sum\limits_{i \in V} \binom{d_G(i)}{2}}. \tag{4.1}$$

Local clustering coefficient is a metric defined for each vertex [62, 63]. It measures

the degree to which neighbors of a node are close to a clique. More formally, the local clustering coefficient C_i of node i of degree $d_G(i) \geq 2$ in G is given by

$$C_i = \frac{\sum_{j,k \in N_G(i), j<k} a_{jk}}{\binom{d_G(i)}{2}}. \tag{4.2}$$

Betweenness centrality $c_b(v)$ of a node v is a measure of a node's centrality in a network and is given by the sum, over all pairs of nodes s, t distinct from v, of the fraction of the shortest paths between s and t that pass through v. More formally,

$$c_b(v) = \sum_{s,t \in V \setminus \{v\}} \sigma_{st}^v / \sigma_{st}, \tag{4.3}$$

where σ_{st} is the number of shortest paths between s and t in G, and σ_{st}^v is the number of those of them that pass through v. *Closeness centrality* $C(v)$ of a node v is another measure of centrality of a node in a connected network. It is defined using the pairwise distances between v and all other nodes in the network as follows:

$$C(v) = \sum_{u \in V \setminus \{v\}} 1/d_G(u, v). \tag{4.4}$$

In the considered experimental data, we have measurements before and after the blast, enabling us to identify changes in the structural properties of the networks summarizing the available data. We compare various parameters, such as the degree distribution, edge density, the local and global clustering coefficient, and other global cohesiveness properties, before and after the treatment.

4.1.2 Network Clustering

Clustering is the task of grouping a set of objects in a way that objects in the same group are more similar to each other according to some distance metric than to

those in other groups. Clustering is one of the most common techniques in network analysis, which is extensively applied to brain networks [19, 30, 21, 39, 13, 22]. Many clustering methods have been developed with different features, like overlapping or non-overlapping clusters, temporal vector or graph representations, with the number of clusters fixed a priori (supervised clustering), etc. In this work we focus on unsupervised clustering (i.e., the number of clusters is not given in advance) with non-overlapping clusters.

We aim to identify related regions of the brain by looking at voxels that are found to be in the same cluster but are in different regions of the brain. Identifying such regions can be important, since if a region is affected in a post-traumatic case, related regions (voxels belonging to same clusters) may also be affected.

We use k-community clustering algorithm [60] which utilizes clique relaxation models in order to partition a network into cohesive subgroups. Two of the clique relaxation models employed in the clustering algorithm are k-core and k-community. A k-core is a group of nodes such that each node in the group is adjacent to at least k other members of the group, whereas a k-community is a connected subgraph such that endpoints of every edge have at least k common neighbors within the subgraph. This structure has been shown to be more effective than k-core in pruning the sparse graphs. In the next section, we explain the experiment design and our methodology.

4.2 Experiment Design and Methodology

TIPS collects fMRI data to understand the effect of concussion on animal brains. Blood oxygen level-dependent (BOLD) fMRI signal is used, which measures brain activity by changes in the blood level. It is a non-invasive way to measure spontaneous brain activity through the low frequency fluctuations in BOLD signals. Experiment involves BOLD measurements for each subject at a resting state before and after

a concussion-inducing blast. A total of 47 animals are exposed to varying degrees of pressure levels. In Table 4.1, for selected subjects, we show their corresponding experimental groups and blast characteristics.

In our analysis, we focus on five specific subjects. We choose two subjects from the control group (i.e., no pressure is applied), and three subjects from the treatment group (i.e., a certain pressure is applied). Control subjects are identified with the following subject identification numbers: 6245 and 6152, and treatment group subjects are 6161, 6239 and 6202.

Table 4.1: Subjects 6161, 6239 and 6202 are exposed to blast-treatment, and 6245 and 6152 are not exposed to any pressure, named as control group.

Subject ID	Group	Incident Pressure (psi)	Duration (msec)	Incident Impulse (psi-msec)
6161	Treatment	68.9	4.73	90.8
6202	Treatment	70.6	5.61	102.0
6239	Treatment	70.7	4.63	75.8
6245	Control	0	0	0
6152	Control	0	0	0

Functional MRI data acquisition Each subject is scanned at 500 time points before and after the blast with Siemens 3 Tesla MRI with C-arm machine. Experiment is designed to collect data at a resting state of a healthy animal first before a concussion and then after a concussion. Each measurement includes data about 102400 voxels, which are the smallest brain units in fMRI data. For each voxel, measurements are taken for 500 consecutive seconds (i.e., data for a voxel is a vector of size 500). To the best of our knowledge, such fMRI measurements with 102400 voxels are some of the most fine-grained and detailed measurements of animal brains. Table 4.2 shows

number of non-isolated nodes and number of edges for all the subjects before and after treatment (for the control group first and second screening).

Table 4.2: Number of non-isolated nodes, number of edges for subjects 6161, 6239 and 6202 are exposed to blast-treatment, and 6245 and 6152 are not exposed to any pressure, total number of nodes is equal to 102400.

Subject ID	Number of non-isolated nodes		Number of edges	
	Before	After	Before	After
6152	25597	12663	1304506	1311234
6245	23568	9465	4190831	396043
6239	19670	28498	4612100	19446111
6161	26916	11966	15966271	4347718
6202	29475	12394	14881718	5744303

In Figure 4.1, we summarize the steps to convert fMRI data into a graph. In step 1 we collect raw fMRI data from several subjects. In step 2, we extract temporal BOLD measurements from the raw data, these include temporal vectors from each voxel. In step 3, for each voxel pair, we calculate temporal cross-correlation values. In step 4, we convert these correlation values into a binary adjacency matrix with predetermined threshold values ranging between [0.7, 0.95]. If the value for a pair is greater than the threshold, we put an edge between the nodes representing these voxels. In step 5, we use the graphs given by the obtained adjacency matrices to perform our graph-based analysis. In the following sections, we illustrate the results using the graphs obtained with a threshold value 0.85. We observe similar results for graphs with different threshold values. Voxels are represented as the nodes in a graph, and relations between voxels are determined by temporal cross correlations.

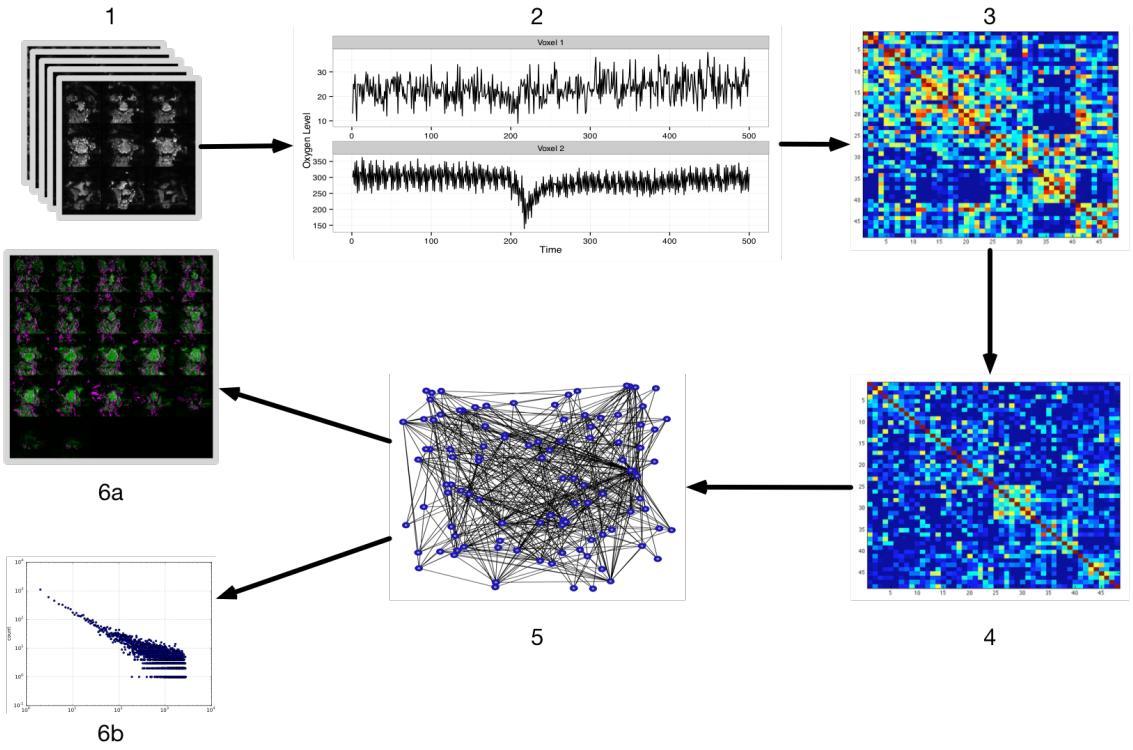


Figure 4.1: A flowchart for the construction and analysis of brain networks from the animal brain by fMRI. Step 1, collection of fMRI data from raw screening. Step 2, extraction of time course data for each voxel. Step 3, creation of temporal cross-correlation matrix. Step 4, creation of binary adjacency matrix with a given threshold. Step 5, graph representation of the threshold-based matrix. Step 6a, clustering analysis on the brain networks. Step 6b, graph theoretical analyses on the brain networks.

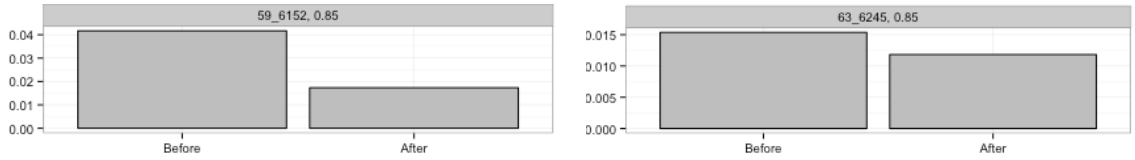
4.3 Summary of Results

4.3.1 Analysis of Structural Properties

In this section, we report the results of the application of network analysis techniques to the graphs representing the fMRI data as described above. We examine the edge density, the degree distribution, the centrality and the clustering coefficient in our control and treatment group. Specifically, we are interested in how a blast affects these parameters for the treatment group.

4.3.1.1 Edge Density

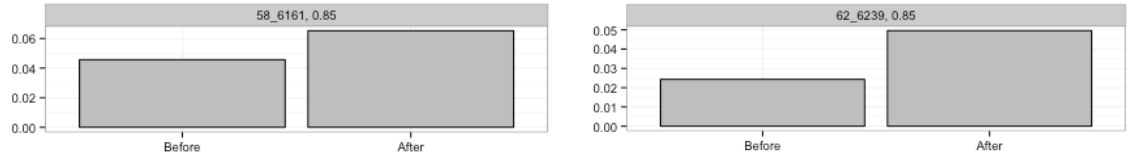
In Figure 4.2, we present the edge density for the graphs corresponding to the subjects in the control group. We observe a decrease in their edge density in the second period compared to the first, even though no blast is applied to the control subjects. We suspect this decrease might be due to the subjects getting used to the environment they are exposed to, or some other environmental factors (change in the air temperature, etc.).



(a) Edge density for 6152. (b) Edge density for 6245

Figure 4.2: Edge density is decreasing on the control group examples.

Conversely, for the treatment group, we observe an increase in the edge density, as we show in Figure 4.3. This might suggest that more and more voxels have correlated BOLD measurements after a certain concussion to restore the functional ability of the brain. However, more experiments are required to substantiate any conclusions.



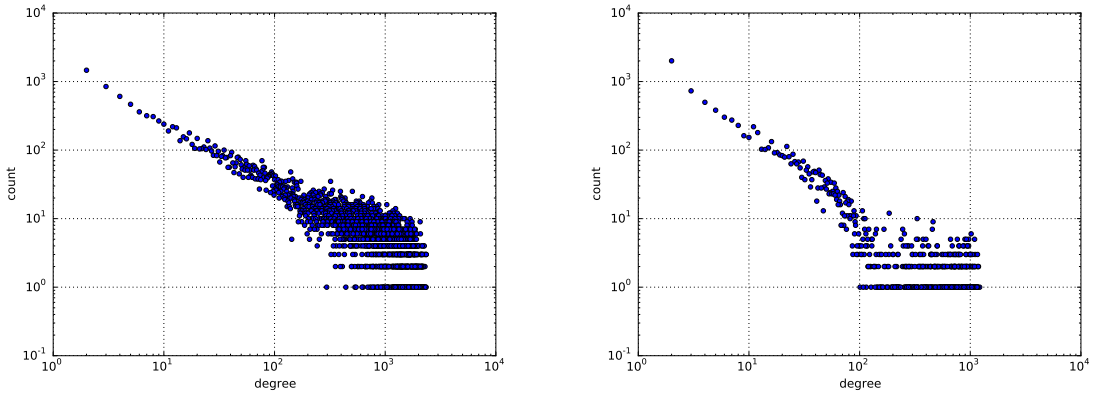
(a) Edge density for 6161. (b) Edge density for 6239

Figure 4.3: Edge density is increasing on the treatment group subjects, whereas it is decreasing on the control group subjects.

4.3.1.2 Degree Distribution

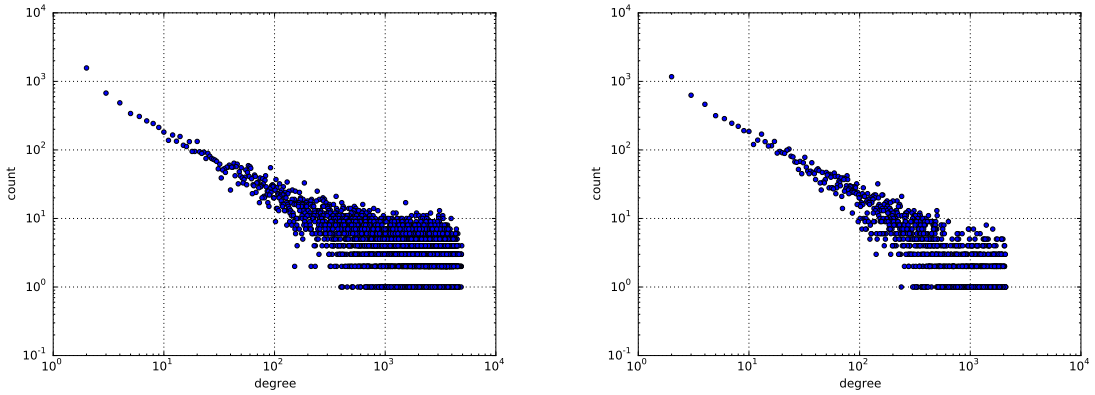
In Figures 4.4 and 4.5, we show the degree distribution comparisons (shown in log-log scale) of the corresponding graphs for control group subjects for their two different measurements. We observe a slightly fewer number of nodes with high degree values in the graphs that correspond to the second set of measurements than the graphs that correspond to the first set.

In contrast, in Figures 4.6 and 4.7, we show the degree distribution comparisons of the corresponding graphs for the treatment group subjects, measured before and after treatment. We observe a considerable increase in the number of nodes with high degree, suggesting that new connections are being made between voxels after concussion.



(a) First measurement (b) Second measurement

Figure 4.4: Degree distribution comparison for subject in the control group (6245)



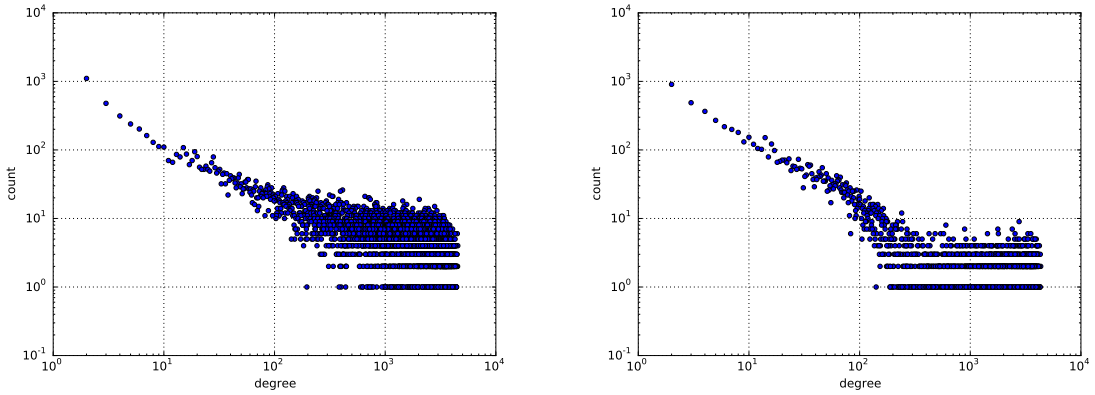
(a) First measurement (b) Second measurement

Figure 4.5: Degree distribution comparison for a control group subject (6152)

4.3.1.3 Clustering Coefficient

We calculate the average local clustering coefficient and global clustering coefficient for each subject before and after concussion in Table 4.3; see Figures 4.10 and 4.11 for illustration. We show the average local clustering coefficient comparisons of the control and treatment groups. We observe a slight increase in the average local clustering coefficient values for the treatment group subjects. At the same time, we observe a decrease in the values for the control group subjects. These results support previous observations that new connections might have been made between voxels after concussion to restore the brain functionality.

On the other hand, the global clustering coefficient increases in all cases. For the control subjects this means that we have a change in the network structure characterized by increase in the edge density and global clustering coefficient, but decrease in the average local clustering coefficient. We would like to explain this situation with two small examples. In Figures 4.8 and 4.9, the average local clustering coefficient decreases while the global clustering coefficient increases.



(a) Pre-concussion (b) Post-concussion

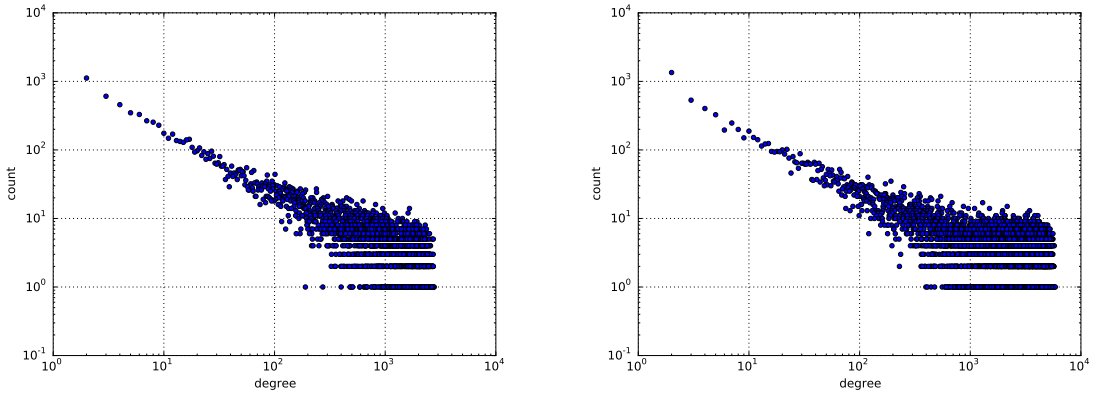
Figure 4.6: Degree distribution comparison for a treatment group subject (6161)

Table 4.3: Comparison of the average local and global clustering coefficients for the selected subjects.

		Average Local C. C.		Global C. C.	
Subject ID	Group	Pre-con.	Post-con.	Pre-con.	Post-con.
6161	Treatment	0.729	0.736	0.608	0.694
6239	Treatment	0.682	0.726	0.574	0.626
6245	Control	0.654	0.542	0.552	0.669
6152	Control	0.704	0.676	0.605	0.647

4.3.1.4 Maximum Clique Size

In Figure 4.12, we show the change in the maximum clique size for the corresponding graphs of control and treatment group subjects. We used branch and bound algorithm to find the maximum cliques for the corresponding graphs [10] .For control group subjects, we observe a considerable decrease in the size of the maximum clique between the two consecutive measurements. On the contrary, for the treatment group we observe an increase in the size of the maximum clique after the concussion, which also aligns with the hypothesis that after concussion the brain



(a) Pre-concussion (b) Post-concussion

Figure 4.7: Degree distribution comparison for a treatment group subject (6239)

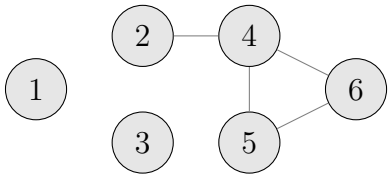


Figure 4.8: Small example with 6 nodes, global clustering coefficient is 3/5 and average local clustering coefficient is 7/9.

might have more voxels with correlated BOLD signals.

4.3.1.5 Centrality

In Table 4.4, we show the change in average betweenness centrality and closeness centrality for the corresponding graphs for treatment and control subjects. For control group subjects, we observe a considerable increase in average betweenness centrality. On the other hand, for the treatment group subjects, average betweenness centrality is decreasing or remaining almost the same. This conclusion also supports the hypothesis that after concussion the brain has a more cohesive network structure. Closeness centrality comparisons does not give any interesting insights

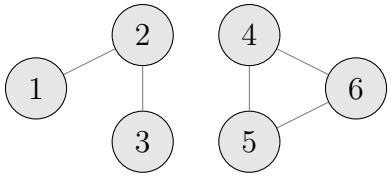


Figure 4.9: Small example-2 with 6 nodes, global clustering coefficient is 3/4 and average local clustering coefficient is 3/4.

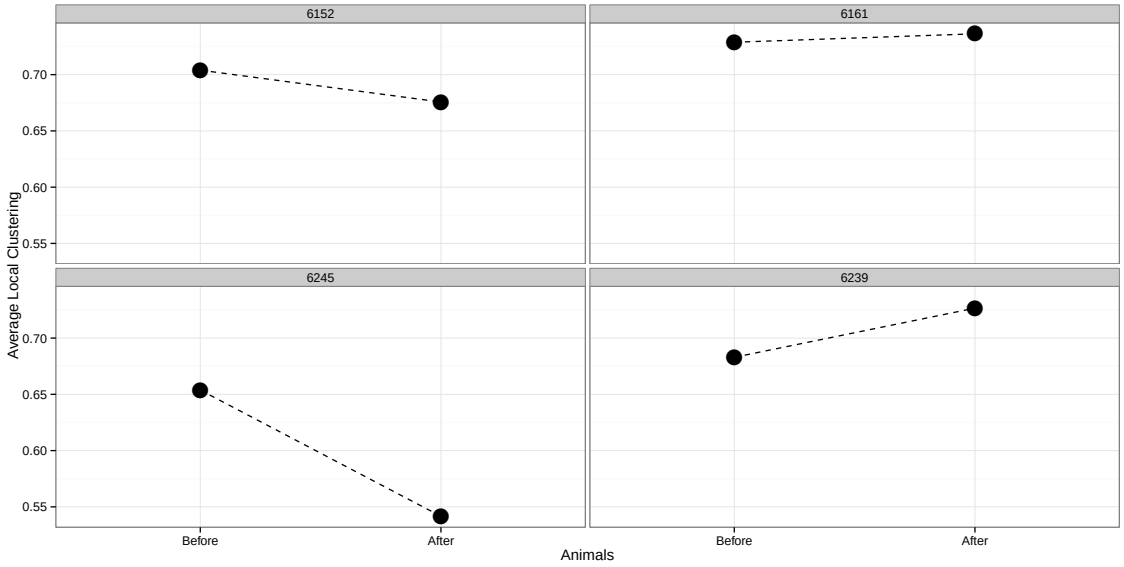


Figure 4.10: Average local clustering coefficients for the graphs before (pre) and after (post) concussion for both control and treatment group.

since both groups behave similarly in terms of this criterion.

4.3.2 Clustering Analysis

The main idea behind k-community clustering [60] is finding k-communities for large k and placing them in different clusters. One of the biggest advantages of this clustering algorithm is, there is no need to specify the number of clusters or any degree distribution a priori. We find the largest k' such that the k'-community of G is non-empty. We place all the k'-communities of G in distinct clusters. Then, we

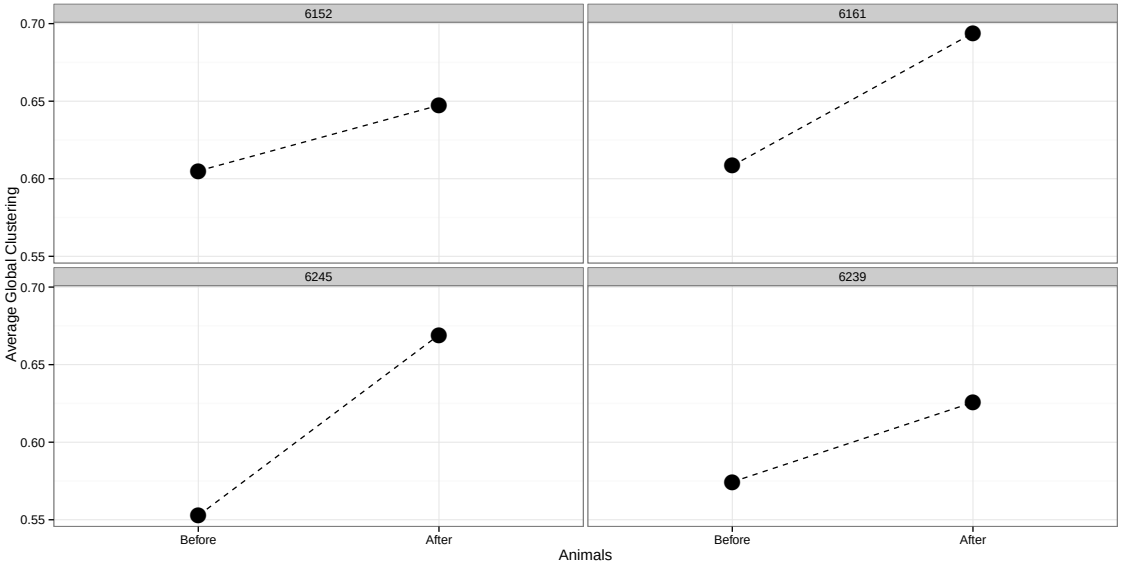


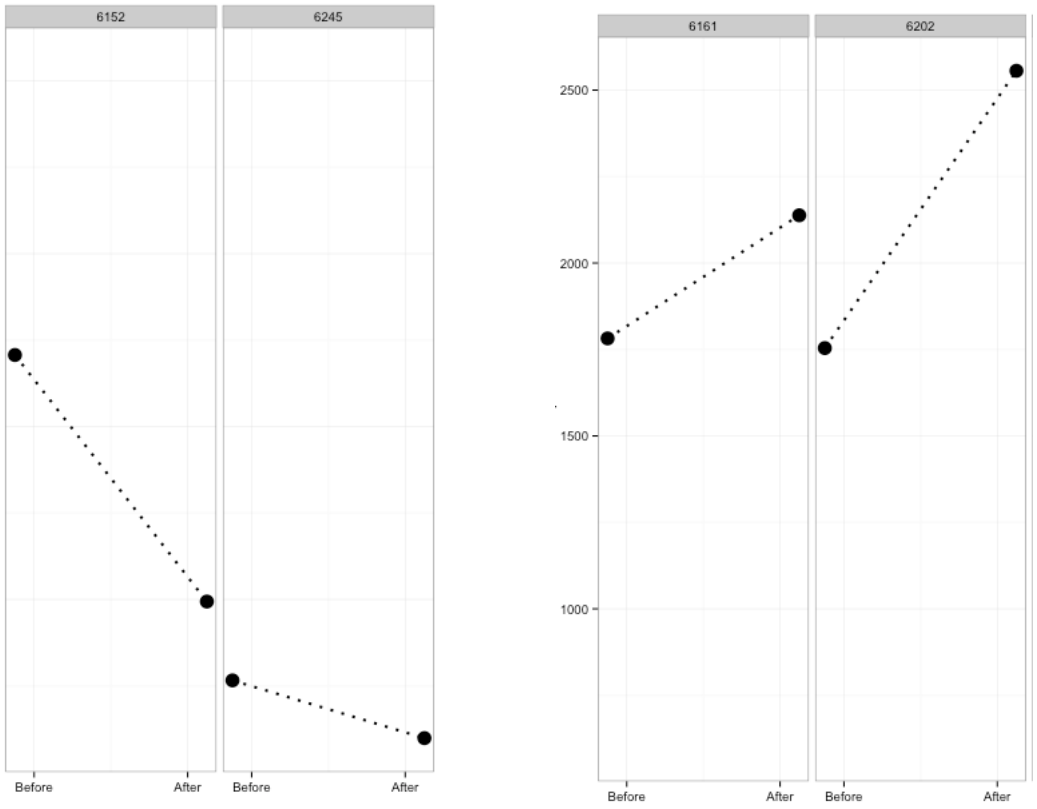
Figure 4.11: Global clustering coefficients for the graphs before (pre) and after (post) concussion for the control and the treatment group.

Table 4.4: Comparison of the average betweenness centrality and closeness centrality for the selected subjects.

		Average betweenness centrality		Closeness centrality	
Subject ID	Group	Pre-con.	Post-con.	Pre-con.	Post-con.
6202	Treatment	0.0011	0.0006	0.081	0.528
6161	Treatment	0.00069	0.0007	0.129	0.475
6239	Treatment	0.0022	0.0009	0.086	0.102
6245	Control	0.0018	0.0029	0.159	0.482
6152	Control	0.0011	0.0027	0.096	0.422

remove from G all the nodes that have been placed in a cluster. We repeat these steps until all nodes are assigned to a cluster.

In k-community clustering algorithm, each data element can be a member of only one cluster, and the number of clusters is flexible. The time complexity of the clustering algorithm is $O(M\Delta^3)$ where M is the number of edges in the graph, and



(a) Control group (b) Treatment group

Figure 4.12: Maximum clique size changes pre- and post-concussion for the control and the treatment group.

Δ is the maximum degree of a vertex in the graph.

Table 4.5: Comparison of number of clusters and the cluster sizes for the selected subjects.

ID	Pre/Post	NumClusters	ClusterSizes
6161	Pre	2	[26418, 16]
6161	Post	15	[8473, 725, 176, 315, 867, 165, 266, 75, 55, 71, 69, 79, 54, 55, 99]
6152		5	[24034, 105, 324, 222, 342]
6152	Control	11	[6534, 3333, 300, 373, 509, 165, 143, 232, 170, 414, 129]

Clusters of correlated activity in fMRI data can identify regions of interest and indicate interacting brain areas. We compare the number of clusters and cluster sizes in Table 4.5. The number of clusters for both the treatment group and the control group increase in their second measurement.

After finding the clusters, mapping the regions back to the actual brain and interpreting the correlations can yield interesting insights. Voxels in the same cluster are highly correlated. We map the clusters back into the original fMRI image (See Figures 4.13-4.16 for the images). Colored voxels indicate the voxels in the same cluster. The results of these figures are inconclusive. The k-community clustering algorithm does not aim to optimize any objective. As a future work we are planning to find another clustering algorithm that better fits the clustering of brain networks.

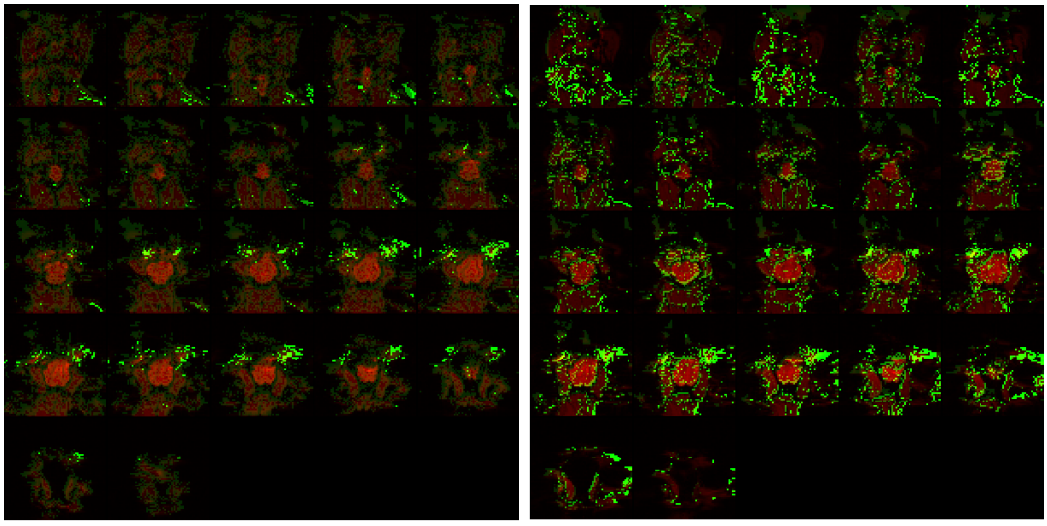


Figure 4.13: Identified clusters on the brain fMRI image for a subject in the control group (6152)

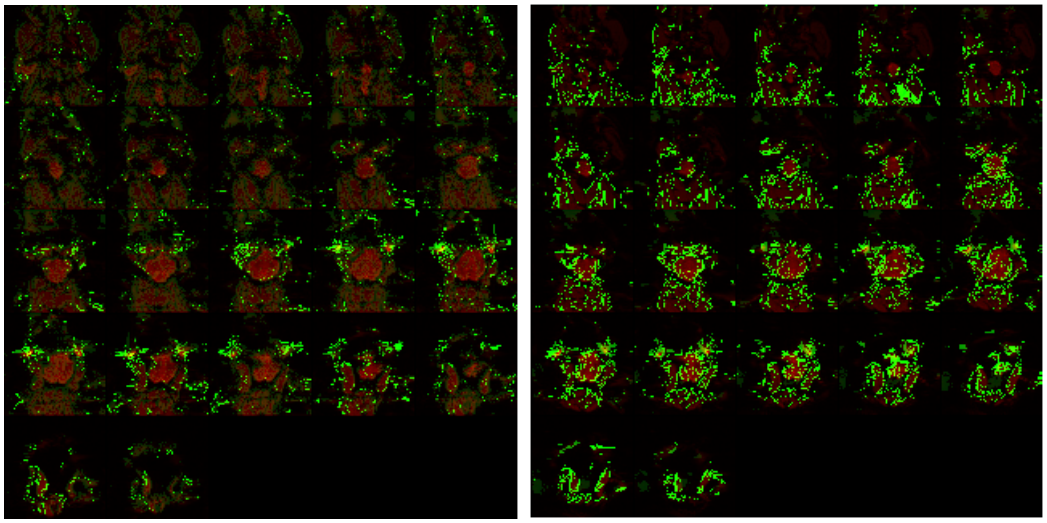


Figure 4.14: Identified clusters on the brain fMRI image for a subject in the control group (6245)

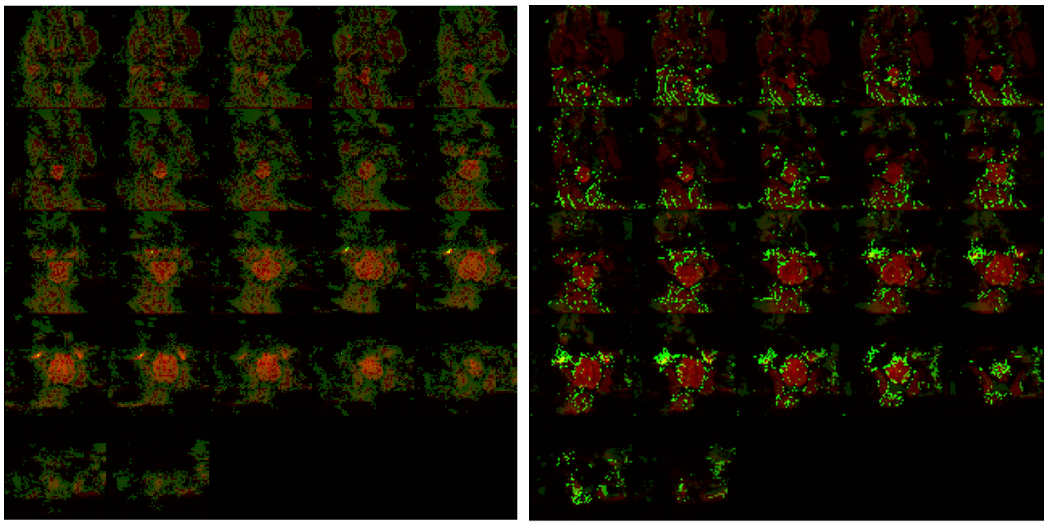


Figure 4.15: Identified clusters on the brain fMRI image for a subject in the treatment group (6161)

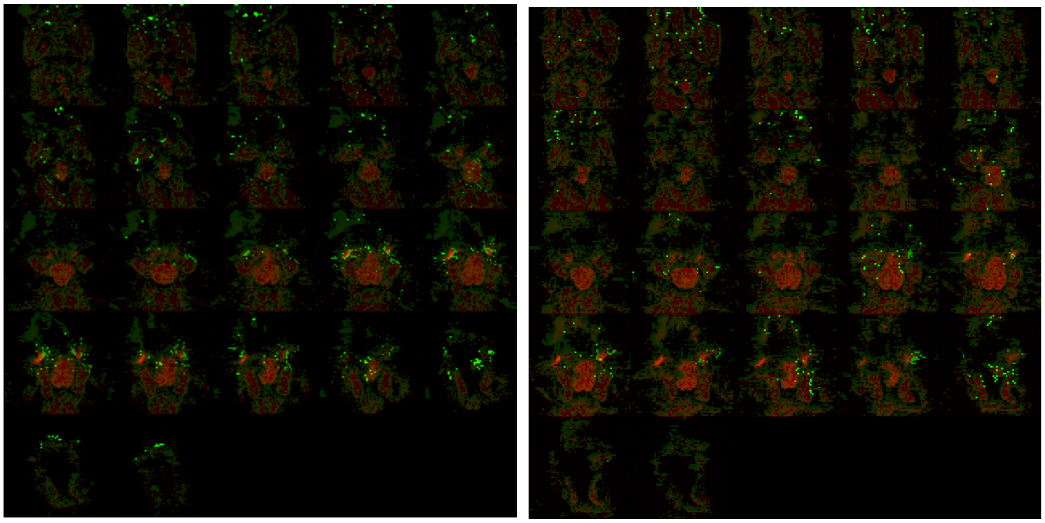


Figure 4.16: Identified clusters on the brain fMRI image for a subject in the treatment group (6239)

Table 4.6: Summary of graph theoretical measures before and after the blast in treatment and control groups

Graph theoretical measure	Control	Treatment
Edge density	$\Downarrow$	$\Uparrow$
Degree distribution	$\Downarrow$	$\Uparrow$
Average local clustering coefficient	$\Downarrow$	$\Uparrow$
Maximum clique size	$\Downarrow$	$\Uparrow$

In Table 4.6 we summarize the effect of blast in treatment and in control group. These results suggests that new connections may have been formed after the blast.

4.4 Conclusion

This section presents basic structural analysis on the brain networks. We use a very unique experimental data which involves fMRI measurements of animal subjects

at two phases. In the first phase all of the animals have been screened at a resting state. In the second phase, some animals are given a certain level of concussion and others represent the control group and animals are screened again. We examine the effect of concussion on animal brains by using graph mining tools such as edge density, degree distribution, centrality, clustering coefficient and maximum clique sizes. We observe that edge density values increase in the treatment group subjects after the concussion. Degree distributions become more heavy-tailed, that is, more nodes have higher degree values after concussion. Average betweenness centrality is decreasing after the concussion. The average local clustering coefficient values increase after the treatment suggesting the nodes become more clustered. These results suggest that new connections might have been made between voxels after concussion to restore the functional stability of the animal brain.

5. CONCLUSION AND FUTURE WORK

In this dissertation, we present three main contributions. In the first part, we introduce new clique relaxation models that are based on local and global clustering coefficients, respectively. Since clustering coefficients are commonly used to asses small-world properties of networks, imposing a high lower bound α on the clustering coefficient (local or global) within a cluster ensures that the corresponding subnetwork has strong small-world properties. We formulate optimization models that allow to compute largest local and global α-clusters in a network and use them to compute solutions for several real-world social networks from the literature. We observe that local α-clusters better identify real-life cohesive subgroups than their global counterparts. We also use local α-clusters to develop a network clustering approach, referred to as local α-clustering algorithm. The method first computes the largest local α-cluster with relaxed connectivity constraints and then uses each connected component of the solution as a seed cluster. Experiments on the well-known graphs from literature with the proposed algorithm show very promising results.

Our work about α-clusters can be extended in several interesting directions. For example, the proposed optimization approach could be enhanced by using scale reduction techniques and alternative formulations. Another direction, the inherent computational intractability of the considered problems, which generalize the notoriously hard maximum clique problem, suggests that developing effective heuristic algorithms could be practically useful, when one has to deal with networks consisting of millions of nodes and edges.

Another important practical issue that needs to be addressed is the choice of the values of α that would yield α-clusters of interest for a particular application.

Intuitively, this choice should depend on the (local) clustering coefficients of the network under investigation as well as other application-specific criteria. Perhaps one could develop semi-automated rules for choosing appropriate values of α. Furthermore, depending on the application, the proposed models could be enhanced by enforcing additional constraints to model desired properties of cohesive subgroups. Such modeling enhancements could also lead to computational advantages as they could reduce the set of feasible solutions. Finally, this work can be extended to directed [63], weighted [48], and two-mode networks [47].

In the second part of this dissertation, we introduce a novel mathematical model called *independent union of cliques* which is closely related to two classical problems in combinatorial optimization; maximum clique and maximum independent set problems. These two structures are closely related and many computational results may be applied equally well to either problem. We introduce a new mathematical model that in a way combines maximum clique and maximum independent set problems, namely *independent union of cliques* (IUC for short). We explore structural properties as well as complexity results for different graph types. We use exact algorithms like integer programming and branch-and-bound methods as well as the heuristic approaches to find the largest set of vertices that satisfies the novel mathematical model.

Our work related to IUC can be extended in several ways. It can be extended to investigate the expected IUC size of random graphs. In our preliminary analysis, we experimented with random graphs on 50 nodes with varying edge density. The results are summarized in Figure 5.1. We observe that IUC solution size is minimum when the edge density of a graph is close to 0.5. Moreover the maximum value is observed when the density is either close to 1 or 0, the graph with all the edges present and the graph with no edges.

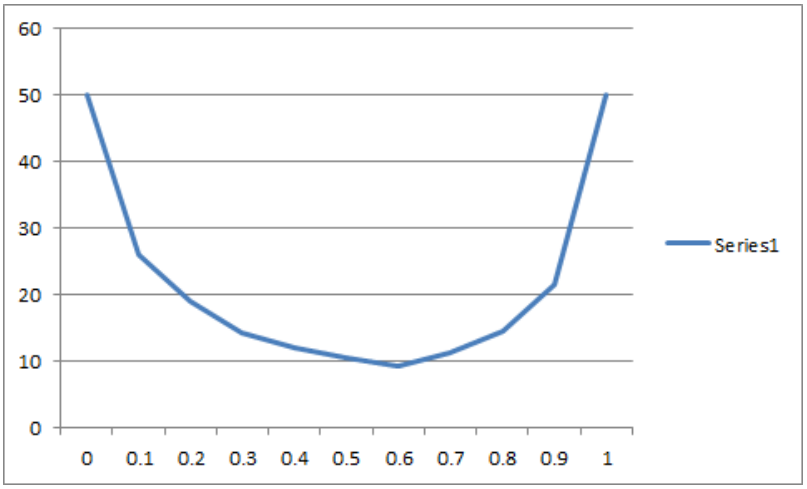


Figure 5.1: IUC solution sizes for random graphs with 50 nodes with varying density level. Solution sizes are average of 40 random graphs in each density value.

One extension of this observation can be characterizing the expected size of an IUC solution for Erdös-Rényi graphs as a function of p (i.e., the probability of randomly choosiness two nodes being neighbors of each other). In addition, extensive polyhedral studies of the maximum IUC problem can be performed. Also, more effective heuristic and metaheuristic methods could be developed for the maximum IUC problem.

Finally, this dissertation includes cohesive and robust clustering analysis of animal brain networks utilizing unique and novel experimental data. In collaboration with TIPS, we analyze multiple pairs of fMRI data about animal brains that are measured before and after a concussion. We utilize network analysis to first identify the similar regions in animal brains, and then compare how these regions as well as graph structural properties change before and after a concussion. To the best of our knowledge, this study is unique in the literature in that it explicitly examines the relation between concussion level and the functional unit interaction, with detailed and fine-grained fMRI measurements. We observe that graphs correspond to animal

brains after treatments have larger edge density, nodes with more neighbors and larger average local clustering coefficient. These pieces of observed evidence suggest that animal brains after concussion might have more voxels with correlated BOLD measurements to restore the brain functionality before concussion.

We would like to extend our work by applying α-clustering algorithm to the brain networks. To increase the scalability of α-cluster, we would like to introduce heuristic approaches and revise our clustering algorithm to handle large graphs.

REFERENCES

[1] J. Abello, S. Butenko, P. M. Pardalos, and M. G. C. Resende. Finding independent sets in a graph using continuous multivariable polynomial formulations. *Journal of Global Optimization*, 21:111–137, 2001.

[2] J. Abello, P. M. Pardalos, and M. G. C. Resende. On maximum clique problems in very large graphs. In *In External Memory Algorithms*, pages 119–130. American Mathematical Society, 1999.

[3] R. D. Alba. A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3:3–113, 1973.

[4] B. P. W. Ames and S. A. Vavasis. Convex optimization for the planted k-disjoint-clique problem. *Mathematical Programming*, 143(1-2):299–337, 2014.

[5] A. Anderson and M. S. Cohen. Decreased small-world functional network connectivity and clustering across resting state networks in schizophrenia: an fmri classification tutorial. *Frontiers in Human Neuroscience*, 7(520), 2013.

[6] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of np. *J. ACM*, 45(1):70–122, January 1998.

[7] E. Balas, V. Chvátal, and J. Nešetřil. On the maximum weight clique problem. *Mathematics of Operations Research*, 1987.

[8] E. Balas and C. S. Yu. Finding a maximum clique in an arbitrary graph. *SIAM J. Comput.*, 15(4):1054–1068, November 1986.

[9] E. Balas and C. S. Yu. On graphs with polynomially solvable maximum-weight clique problem. *Networks*, 19(2):247–253, 1989.

[10] B. Balasundaram, S. Butenko, and I. V. Hicks. Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*, 59(1):133–142, 2011.

[11] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[12] R. Baumgartner, C. Windischberger, and E. Moser. Quantification in functional magnetic resonance imaging: Fuzzy clustering vs. correlation analysis. *Magnetic Resonance Imaging*, 16(2):115 – 125, 1998.

[13] A. Baune, F. T. Sommer, M. Erb, D. Wildgruber, B. Kardatzki, and W. Palm, G.and Grodd. Dynamical cluster analysis of cortical fmri activation. *NeuroImage*, pages 477–489, 1999.

[14] B. C. Bernhardt, S. Hong, A. Bernasconi, and N. Bernasconi. Imaging structural and functional brain networks in temporal lobe epilepsy. *Frontiers in Human Neuroscience*, 7(624), 2013.

[15] R. V. Bevern, H. Moser, and R. Niedermeier. Approximation and tidying—a problem kernel for s-plex cluster vertex deletion. *Algorithmica*, 62(3-4):930–950, 2012.

[16] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization*, pages 1–74. Kluwer Academic Publishers, 1999.

[17] R. Boppana and M. M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT Numerical Mathematics*, 32(2):180–196, 1992.

[18] K. Caeyenberghs, A. Leemans, I. Leunissen, K. Michiels, and S. P. Swinnen. Topological correlations of structural and functional networks in patients with traumatic brain injury. *Frontiers in Human Neuroscience, 7, 726.*, 2013.

[19] D. Cordes, V. Haughton, J. C. Carew, K. Arfanakis, and K. Maravilla. Hierarchical clustering to measure connectivity in fmri resting-state data. *Magnetic Resonance Imaging*, 20(4):305–317, 2002.

[20] A. Dessmark, K. Jansen, and A. Lingas. *The maximum k-dependent and f-dependent set problem*. Springer, 1993.

[21] E. Dimitriadou, M. Barth, C. Windischberger, K. Hornik, and E. Moser. A quantitative comparison of functional {MRI} cluster analysis. *Artificial Intelligence in Medicine*, 31(1):57 – 71, 2004.

[22] S. Dodel, J.M. Herrmann, and T. Geisel. Functional connectivity by cross-correlation clustering. *Neurocomputing*, 44–46:1065 – 1070, 2002.

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

[24] Z. Ertem, M. Lenox, and S. Butenko. Graph theoretical analysis on brain networks. Working paper, 2015.

[25] Z. Ertem, A. Veremyev, and S. Butenko. Detecting large cohesive subgroups with high clustering coefficients. Submitted to Social Networks, 2014.

[26] Z. Ertem, Y. Wang, and S. Butenko. Characterizing and detecting independent union of cliques. Working paper, 2014.

[27] E. D. Fagerholm, P. J. Hellyer, G. Scott, R. Leech, and D. J. Sharp. Disconnection of network hubs and cognitive impairment after traumatic brain injury. *Brain*, page awv075, 2015.

[28] U. Feige and J. Kilian. Zero knowledge and the chromatic number. *J. Comput. System Sci.*, 57:187–199, 1998.

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

[30] S. Ghebreab and A. W. M. Smeulders. Identifying distributed and overlapping clusters of hemodynamic synchrony in fmri data sets. *Pattern Analysis and Applications*, 14(2):175–192, 2011.

[31] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[32] F. Glover. Surrogate constraints. *Operations Research*, 16:741–749, 1968.

[33] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, second corrected edition edition, 1993.

[34] Y. He and A. Evans. Graph theoretical modeling of brain connectivity. *Current Opinion in Neurology*, 23(4):341–350, 2010.

[35] Matthew O Jackson et al. *Social and economic networks*, volume 3. Princeton University Press Princeton, 2008.

[36] K. Jansen, P. Scheffer, and G. Woeginger. The disjoint cliques problem. *Operations Research*, 31:45–66, 1997.

[37] R. M. Karp. Reducibility Among Combinatorial Problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[38] V. E. Krebs. Uncloaking terrorist networks. *First Monday*, 7, 2002.

[39] O. Lange, A. Meyer-Baese, M. Hurdal, and S. Foo. A comparison between neural and fuzzy cluster analysis techniques for functional {MRI}. *Biomedical Signal Processing and Control*, 1(3):243 – 252, 2006.

[40] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.

[41] R. Luce. Connectivity and generalized cliques in sociometric group structure. *Psychometrika*, 15(2):169–190, June 1950.

[42] R. D. Luce and A. D. Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116, 1949.

[43] D. Lusseau. The emergent properties of a dolphin social network. *eprint arXiv:cond-mat/0307439*, July 2003.

[44] R. Mokken. Cliques, clubs and clans. *Quality & Quantity: International Journal of Methodology*, 13(2):161–173, April 1979.

[45] J. Nastos and Y. Gao. Familial groups in social networks. *Social Networks*, 35(3):439 – 450, 2013.

[46] M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[47] T. Opsahl. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Social Networks*, 35(2):159–167, 2013.

[48] T. Opsahl and P. Panzarasa. Clustering in weighted networks. *Social Networks*, 31(2):155 – 163, 2009.

[49] P. M. Pardalos and J. Xue. The maximum clique problem. *Journal of Global Optimization*, 4:301–328, 1994.

[50] J. Pattillo, N. Youssef, and S. Butenko. Clique relaxation models in social network analysis. In My T. Thai and Panos M. Pardalos, editors, *Handbook of Optimization in Complex Networks*, Springer Optimization and Its Applications, pages 143–162. Springer New York, 2012.

[51] A. Sanjeev, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *In Proc. 33rd Ann. IEEE Symp. on found. of comp. sci*, pages 14–23, 1992.

[52] J. Scott. *Social Network Analysis: A Handbook*. Sage Publications, London, 2 edition, 2000.

[53] S. B. Seidman and B. L. Foster. Graph-theoretic generalization of the clique concept. *Journal of Mathematical Sociology*, 6:139–154, 1978.

[54] H. D. Sherali and W. P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Springer, 1st ed. softcover of orig. ed. 1998 edition, December 1998.

[55] F Skidmore, D Korenkevych, Y Liu, G He, E Bullmore, and Panos M Pardalos. Connectivity brain networks based on wavelet correlation analysis in parkinson fmri data. *Neuroscience letters*, 499(1):47–51, 2011.

[56] S. Trukhanov, C. Balasubramaniam, B. Balasundaram, and S. Butenko. Algorithms for detecting optimal hereditary structures in graphs, with application to clique relaxations. *Computational Optimization and Applications*, 56(1):113–130, 2013.

[57] A. Veremyev and V. Boginski. Identifying large robust network clusters via new compact formulations of maximum k-club problems. *European Journal of Operational Research*, 218(2):316–326, 2012.

[58] A. Veremyev, O. A. Prokopyev, and E. L. Pasiliao. Critical nodes for communication efficiency and related problems in graphs. Working paper, 2014.

[59] G. Verfaillie, M. Lemaître, and T. Schiex. Russian doll search for solving constraint optimization problems. In *AAAI/IAAI, Vol. 1*, pages 181–187, 1996.

[60] A. Verma and S. Butenko. Network clustering via clique relaxations: A community based approach. *10th DIMACS Implementation Challenge*, 2011.

[61] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, New York, 1994.

[62] D. J. Watts. *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton University Press, Princeton, NJ, 1999.

[63] D. J. Watts and S. H. Strogatz. Collective dynamics of'small-world'networks. *Nature*, 393(6684):409–10, 1998.

[64] M. Yannakakis. Node-and edge-deletion np-complete problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 253–264, New York, NY, USA, 1978. ACM.

[65] H. Yu, A. Paccanaro, V. Trifonov, and G. Mark. Predicting interactions in protein networks by completing defective cliques. *Bioinformatics*, 22(7):823–829, April 2006.

[66] W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):pp. 452–473, 1977.

[67] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Theory Comput*, volume 3, pages 103–128, 2007.