POLYHEDRAL COMBINATORICS, COMPLEXITY & ALGORITHMS FOR k-CLUBS IN GRAPHS

By

FOAD MAHDAVI PAJOUH

Bachelor of Science in Industrial Engineering
Sharif University of Technology
Tehran, Iran
2004

Master of Science in Industrial Engineering
Tarbiat Modares University
Tehran, Iran
2006

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2012

POLYHEDRAL COMBINATORICS, COMPLEXITY &
ALGORITHMS FOR k-CLUBS IN GRAPHS

Dissertation Approved:

Dr. Balabhaskar Balasundaram

Dissertation Advisor

Dr. Ricki Ingalls

Dr. Manjunath Kamath

Dr. Ramesh Sharda

Dr. Sheryl Tucker

Dean of the Graduate College

Dedicated to my parents

ACKNOWLEDGMENTS

I would like to wholeheartedly express my sincere gratitude to my advisor, Dr. Balabhaskar Balasundaram, for his continuous support of my Ph.D. study, for his patience, enthusiasm, motivation, and valuable knowledge. It has been a great honor for me to be his first Ph.D. student and I could not have imagined having a better advisor for my Ph.D. research and role model for my future academic career. He has not only been a great mentor during the course of my Ph.D. and professional development but also a real friend and colleague. I hope that I can in turn pass on the research values and the dreams that he has given to me, to my future students.

I wish to express my warm and sincere thanks to my wonderful committee members, Dr. Ricki Ingalls, Dr. Manjunath Kamath and Dr. Ramesh Sharda for their support, guidance and helpful suggestions throughout my Ph.D. research and applications for academic positions. Their guidance and support served me well and I owe them my heartfelt appreciation.

I am also deeply grateful to Dr. Illya V. Hicks, Dr. Sergiy Butenko, Dr. Oleg Prokopyev, Dr. Vladimir Boginski, for advising me and supporting me in my research and during my search for a faculty position. I consider myself truly lucky to be able to receive guidance and support from these great and knowledgable individuals.

The School of Industrial Engineering and Management (IE&M) at Oklahoma State University (OSU) continuously supported my Ph.D. study and provided me with wonderful opportunities for professional development. I owe my sincere gratitude to Dr. William J. Kolarik, Head of IE&M department, for supporting me to pursue my Ph.D. in the Operations Research area, and also providing me with several opportunities to

teach in IE&M department.

My warm thanks are due to efficient and friendly administrative staff at IE&M department. I want to specially thank Mindy Bumgarner, Melissa Miller and Lyndsey Wenninger for their help and support during my Ph.D. study at OSU.

I wish to also thank Esmaeel Moradi, Zhuqi Miao and Juan Ma for being wonderful friends and great colleagues in our research group at OSU.

This Ph.D. dissertation research was partially sponsored by U.S. Department of Energy grant DE-SC0002051. This financial support is greatly appreciated. Some of the computational experiments discussed in this dissertation were performed at the OSU High Performance Computing Center. I am grateful to Dr. Dana Brunson for her support in conducting these experiments.

Words cannot express the love I have for my parents. I wish to express my heartfelt gratitude to my parents, Rouhollah Mahdavi Pajouh and Manijeh Keyhanshekouh, for their never-ending love, support and encouragement. I am forever indebted to my parents and wish I could show them just how much I love and appreciate them. Last but not the least, I want to deeply thank my lovely wife, Pia Guyman, for her faithful support, encouragement and patience. Her presence and companionship has turned my journey through this Ph.D. program into a pleasure. For all that, and for giving me life and her heart, she has my everlasting love.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Recent advances in high-throughput data collection in different fields such as internet analytics, social network analysis, bioinformatics, finance, and telecommunication among others have led to an increased demand for effective models and tools for data mining. Data mining is the process of summarizing, visualizing and processing large datasets in order to extract useful knowledge from the data by using advanced mathematical techniques [1]. In a variety of data mining applications, the elements of a system and the relationships among them are modeled as a graph which is often visualized as vertices (points, nodes) connected by edges (lines, arcs). Graph algorithms and optimization techniques are used to uncover specific patterns in such graph models of data [2, 3].

1.1 Graph models of data

In several real world systems, the data can be modeled by a graph in which an edge implies similarity (or dissimilarity) between the entities represented by the endpoints of the edge. Some examples are social networks, internet graphs, call graphs, stock-market graphs and biological networks. *Social networks* are graph models representing sociological information such as acquaintance among people. Vertices usually represent people and an edge indicates a "tie" between two people. A tie could mean that they know each other, they visited the same place or any other sociological connection. *Scientific collaboration networks* with vertices representing authors (in a particular field or with publications in a particular journal) and edges indicating co-authorship

fall under this category [4, 5, 6, 7, 8]. An *internet graph* has vertices representing IP addresses and edges in such graphs are determined based on information from routing protocols or using traceroute probes [9]. In web mining and internet analytics, the web can be modeled by a graph in which each webpage is represented by a vertex and two vertices are adjacent if the corresponding webpages are linked together [10]. In *call graphs*, vertices represent telephone numbers and an edge represents a call placed from one vertex to another in a specified time interval [11]. *Stock-market graphs* have vertices representing stocks and two stocks are connected by an edge if their prices are positively correlated over some threshold value over a period of time in history [12, 13, 14]. Biological networks such as *protein interaction networks (PIN)*, *gene co-expression networks (GCN)* and *metabolic networks* are used to model biological information. A protein interaction network is represented by a graph with the proteins as vertices, and an edge exists between two vertices if the proteins are known to interact based on two-hybrid assays and other biological experiments [15, 16, 17]. In gene co-expression networks, vertices represent genes and an edge exists between two vertices if the corresponding genes are co-expressed with correlation higher than a specified threshold in microarray experiments [18]. A metabolic network represents metabolites (molecules of glucose, amino acids, and macro-molecules like polysaccharides) and their conversion through enzyme-catalyzed biochemical reactions. In these networks, metabolites are represented by vertices and a directed edge from metabolite A to B indicates that A is converted to B by some reaction [19].

In some practical situations, one has to deal with uncertainties associated with the system which may result in probabilistic availability/failure of the system components. In such situations, the graph model can be random in which every node/edge has a probability of partial or complete failure. Some examples include power grid component failures, airline hub failures due to weather, or freeway closures due to emergencies. Additionally, some graph models of data such as protein interaction

networks, gene co-expression networks and metabolic networks are constructed based on experimental studies that are subject to incomplete information, as well as experimental errors. For instance, the authors of [19] analyzed the metabolic network of 43 organisms and recognized two main sources of error as the erroneous annotation of biochemical reactions, and missing reactions. In such graphs, the level of confidence about the existence of each component is modeled by a probability distribution.

1.2 Cluster models in graphs

Detecting clusters in graph models of data is a powerful data mining tool. One could formalize the notion of a graph-theoretic cluster by requiring one or more of the following structural properties: (a) Each vertex has a large number of neighbors inside the cluster, (b) The pairwise distances between vertices in the cluster (or in the subgraph induced by the cluster) is small, (c) The cluster has a large number (fraction) of edges with both endpoints inside the cluster, (d) The minimum number of vertices or edges whose removal results in a disconnected cluster is large.

A clique in a graph is a set of pairwise adjacent vertices. A clique is said to be *maximum*, if it is a clique of the largest cardinality in the graph and is said to be *maximal* by inclusion, if it is not strictly contained in a larger clique. In the graph shown by Figure 1.1, set $\{2, 3, 4, 5\}$ forms a maximum clique and set $\{1, 2, 5\}$ is a maximal clique which is not maximum.

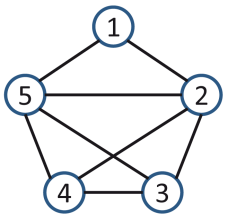

Figure 1.1: A graph in which set $\{2, 3, 4, 5\}$ forms a maximum clique and set $\{1, 2, 5\}$ is a maximal clique which is not maximum

Given a simple, undirected and finite graph, finding a maximum clique, clustering the graph into minimum number of cliques that cover or partition the vertex set, and enumerating inclusionwise maximal cliques are fundamental combinatorial optimization problems associated with cliques. This basic model and the associated problems have been well studied in graph theory, polyhedral combinatorics, and complexity theory leading to many deep results in these areas.

Clique is an *ideal* graph model for a cluster in which every vertex has maximum degree, the pairwise distances within the cluster is minimum, the number of possible edges within the cluster and the edge/vertex connectivity is maximum. Cliques, and many graph theoretic *clique relaxations* were originally proposed in *social network analysis* (SNA) to model *cohesive subgroups* in social networks [20]. The need for relaxations of the clique model arises in practice when dealing with massive data sets which are inevitably prone to errors that manifest in the graph model as false or missing edges. The clique definition which requires complete pairwise adjacency in the cluster becomes overly restrictive in such situations. Graph-theoretic clique relaxations address this need by relaxing the aforementioned structural properties in a controlled manner via user-specified parameters. This relaxation can be based on vertex degree (k-plex), distance (k-clique, k-club) or edge density (γ-quasi-clique). The focus of this dissertation is on a distance-based relaxation of cliques known as k-clubs which model low diameter clusters in graphs.

1.3 Notations and definitions

In this dissertation, an arbitrary, simple, finite, undirected graph of order n and size m is denoted by $G = (V, E)$ where $V = \{1, \ldots, n\}$ and $(i, j) \in E$ when vertices i and j are adjacent with $|E| = m$. For a vertex $i \in V, N_G(i)$ denotes the set of vertices adjacent to i in G, called the *neighborhood* of i and $N_G[i] = \{i\} \cup N_G(i)$ denotes the *closed neighborhood* of i. The set of *non-neighbors* of i is given by $V \setminus N_G[i]$.

Denote by $deg_G(i)$, the degree of vertex i in G given by $|N_G(i)|$. The maximum and minimum degrees in a graph are denoted respectively by $\Delta(G)$ and $\delta(G)$. The *complement graph* of G is denoted by $\overline{G} = (V, \overline{E})$. Given a simple, undirected graph $G = (V, E)$, the subgraph induced by $S \subseteq V$ is denoted by $G[S] = (S, E \bigcap (S \times S))$ (see [21] for basic definitions from graph theory). The *vertex connectivity* $\kappa(G)$ and *edge connectivity* $\lambda(G)$ of a graph are the minimum number of vertices and edges, respectively, whose removal results in a disconnected or trivial graph. It is known that when G is nontrivial, $\kappa(G) \leq \lambda(G) \leq \delta(G)$ [22].

Definition 1.1 *A subset $S \subseteq V$ is called a clique if $G[S]$ is complete and it is called an independent set if $G[S]$ is edgeless.*

The maximum clique problem is to find a clique of maximum cardinality, referred to as the clique number of G, denoted by $\omega(G)$. The maximum cardinality of an independent set of G is called the *independence number* of the graph G and is denoted by $\alpha(G)$. The associated problem of finding a largest independent set is the *maximum independent set problem.* Clearly, I is an independent set in G *if and only if* I is a clique in $\overline{G}$ and consequently $\alpha(G) = \omega(\overline{G})$. In this dissertation, *maximality* and *minimality* of sets are always defined based on inclusion and exclusion, respectively. The maximum clique and independent set problems on arbitrary graphs are NP-hard [23] and are hard to approximate within $n^{1-\epsilon}$ for any $\epsilon > 0$ [24].

Definition 1.2 *A proper coloring of a graph is one in which every vertex is colored such that no two vertices of the same color are adjacent.*

A graph is said to be *t-colorable* if it admits a proper coloring with t colors. Vertices of the same color are referred to as a *color class* and they induce an independent set. The *chromatic number* of the graph, denoted by $\chi(G)$ is the minimum number of colors required to properly color G. For any graph G, $\omega(G) \leq \chi(G)$, as different colors are required to color the vertices of a clique.

Definition 1.3 *A vertex cover is a subset of vertices such that every edge in the graph is incident at some vertex in the vertex cover.*

Clearly, C is a vertex cover of G if and only if $V \setminus C$ is an independent set in G. *The minimum vertex cover problem* seeks to find a vertex cover of minimum cardinality.

Definition 1.4 *A dominating set is a subset of vertices such that every vertex in the graph is either in this set or has a neighbor in this set.*

The minimum cardinality of a dominating set is called the *domination number*, denoted by $\gamma(G)$. It should be noted that every maximal independent set is also a minimal dominating set.

For two vertices $i, j \in V$, $d_G(i, j)$ denotes the length of the shortest path (in number of edges) between i and j in G. By convention, when no path exists between two vertices, the shortest distance between them is infinity. Given a graph $G = (V, E)$, the k-th power of G is denoted by $G^k = (V, E^k)$ where $E^k = \{(u, v) : d_G(u, v) \leq k\}$. For any given $i \in V$, define distance k-neighborhood of i in G as $N_G^k(i) = \{j \in V : 1 \leq d_G(i, j) \leq k\} = N_{G^k}(i)$, that is the neighborhood of i in the power graph G^k. For any given set $S \subseteq V$, define distance k-neighborhood of S in G as $N_G^k[S] = S \cup (\cup_{i \in S} N_G^k(i))$. The diameter of $G = (V, E)$ is given by $diam(G) = \max_{i,j \in V} d_G(i, j)$. The diameter of G is said to be infinite if there does not exist a path between a pair of vertices. If the graph G under consideration is obvious, the subscript G in the neighborhood, degree and distance notations is sometimes dropped for simplicity.

Definition 1.5 *A k-clique is a subset $S \subseteq V$ for which $d_G(u, v) \leq k$ for all $u, v \in S$.*

Definition 1.6 *A k-club is a subset $S \subseteq V$ for which $d_{G[S]}(u, v) \leq k$ for all $u, v \in S$. Equivalently, S is a k-club if $diam(G[S]) \leq k$.*

Clearly, every k-club is also a k-clique since $d_{G[S]}(u, v) \leq k \Rightarrow d_G(u, v) \leq k$. However, for $k \geq 2$ the converse is not always true. In fact for $k \geq 2$, a k-clique S can

contain vertices $u, v \in S$ such that $d_G(u, v) \leq k$ but $d_{G[S]}(u, v) > k$. In the graph G shown in Figure 1.2 [25], set $S = \{1, 2, 3, 4, 5\}$ forms a 2-clique which is not a 2-club. It should be noted that $d_G(1, 5) = 2$ and $d_{G[S]}(1, 5) = 3$.

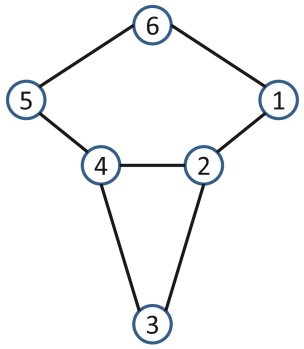

Figure 1.2: A graph G in which set $S = \{1, 2, 3, 4, 5\}$ is a 2-clique but not a 2-club

Both k-cliques and k-clubs describe a clique when $k = 1$ and are relaxations when $k \geq 2$. Further, every k-club is also a $(k + 1)$-club by definition. The *k-clique number* of G denoted by $\widetilde{\omega}_k(G)$ is the cardinality of a maximum k-clique in G and the maximum k-clique problem is to find a k-clique of the maximum cardinality in G. The *k-club number* of G denoted by $\overline{\omega}_k(G)$ and the maximum k-club problem are similarly defined. For a given graph G and a positive integer k, the following inequality is true:

$$\omega(G) \leq \overline{\omega}_k(G) \leq \widetilde{\omega}_k(G).$$

Concepts of particular relevance to these models are distance-k independence [26] and distance-k domination.

Definition 1.7 *Given a graph $G = (V, E)$, $I \subseteq V$ is a distance-k independent set (k-independent set for short) if for every distinct pair $i, j \in I$, $d_G(i, j) \geq k + 1$.*

Clearly, distance-1 independence is equivalent to pairwise nonadjacency, that is a classical independent set. It should be noted that the definition becomes more restrictive as k increases. A distance-$(k+1)$ independent set is also distance-k independent,

but the converse is not always true. Further, I is a k-independent set in G if and only if I is an independent set in the power graph G^k. Also note that if I is a k-clique in G, I need not be a k-independent set in the complement graph $\bar{G}$ for $k \geq 2$. For instance, $\{1, 2, 4, 5, 6\}$ is a 2-clique/2-club in the graph in Figure 1.2, but it is not a 2-independent set in the complement. In fact, it is still a 2-clique/2-club in the complement graph. Additionally, a k-clique or a k-club can intersect a k-independent set in at most one vertex.

Definition 1.8 *A set $S \subseteq V$ is called a distance-k dominating set (k-dominating set for short) if for any $v \in V \setminus S$, there exists a $u \in S$ such that $d_G(u, v) \leq k$.*

A distance-1 dominating set is a classical dominating set. A distance-k dominating set is also distance-$(k+1)$ dominating, but the converse is not necessarily true. Additionally, I is a k-dominating set in G if and only if I is a dominating set in the power graph G^k.

Note that k-cliques are equivalent to cliques through a simple observation concerning power graphs. Set $S \subseteq V$ is a k-clique in G if and only if S is a clique in G^k. For instance, set $\{1, 2, 4, 5, 6\}$ which is a 2-clique in graph G shown in Figure 1.2, forms a clique in G^2. Accordingly, given a graph G and a positive integer k, $\widetilde{\omega}_k(G) = \omega(G^k)$. Hence, the maximum k-clique problem for any k is equivalent to the maximum clique problem on the corresponding power graph. Because of this close correspondence between k-cliques in original graph and cliques in the power graph, the k-clique model has not been extensively studied.

1.4 Applications of k-clubs

The k-clubs, in particular 2-clubs, have been a practical and popular choice for clusters in various applications. This is possibly encouraged by the fact that graph models of data from biology, internet analytics and social sciences are known to exhibit power-

law degree distribution (also referred to as "scale-free graphs"). In such graphs, n_t which denotes the number of vertices of degree t has been empirically observed to obey a power law $n_t \propto t^{-\beta}$ where $\beta > 1$ is a constant. Such graphs tend to have a large number of vertices and an extremely small number of edges, and it has been argued that underlying their evolution is a *preferential attachment scheme* where new edges tend to appear at vertices that already have high degree, resulting in the power-law degree distribution and a large connected component of low diameter [27, 28, 29].

The probabilistic combinatorics of large, sparse graphs such as power-law graphs are not appropriately described by the classical Erdös-Rényi uniform random graph model $G(n, p)$ [30, 31]. An extension of the Erdös-Rényi model for large, sparse graphs are random graphs of given expected degree sequence [32]. Given a sequence $w = (w_1, w_2, \ldots, w_n)$, a random graph $G(w)$ is defined where the probability of an edge between $i, j \in V$ is $p_{ij} = w_i w_j / \sum_{k \in V} w_k$. This model includes the classical $G(n, p)$ model which is obtained by choosing $w = (np, np, \ldots, np)$, and it represents a power-law random graph model when w is a sequence that obeys a power law. Employing such models, it has been recently shown that power-law random graphs with exponent $\beta \in (2, 3)$ (empirically observed values of β often lie in this range) have connected components of diameter $O(\log n)$ and a distinct core of diameter $O(\log \log n)$ with high probability. This result has been noted empirically in other disciplines as "small world phenomenon" [33, 34] and "six degrees of separation" [35].

Due to the low diameter of G in which we seek to find clusters, 2-clubs have been more popular in practice than k-clubs with larger k. Furthermore, 2-clubs can be viewed as a subset of vertices in which every pair of vertices either have a direct edge or have a common neighbor in the subset. This intuitive "two-hop" interpretation has encouraged the choice of 2-clubs in many applications where such a two-hop transitivity is expected. In other words, it is expected that if two vertices have a common neighbor, then the two vertices are likely to be related even if a direct edge

does not exist. This is employed in data mining applications in internet analytics and text mining. For instance, the 2-club model is used to cluster web sites to facilitate faster search and retrieval of topically related information from the internet in [10]. Authors of [10] note that the 2-club is a "key construct that productively formalizes" the notion of local collections of topically related web sites. Similar approaches are used in a text mining application in hyper-linked documents in [36]. Authors of [36] conclude that the two-hop transitivity captured by the 2-club concept is particularly insightful in this application, even compared to other k-clubs for k different from 2.

The concept of k-clubs can be used to detect protein complexes from protein interaction networks of organisms [37, 38]. It was observed in [37] that the maximum k-club identified in the protein interaction networks of *Helicobacter Pylori* and *Saccharomyces Cerevisiae* contained a $(k-2)$-club kernel and vertices outside this kernel were adjacent to some vertex inside the kernel. This was interesting in the light of observations that structures where interactions of proteins occur through a central protein are likely to be found in similar biological processes [39]. Figure 1.3 shows a maximum 2-club cluster and a large 3-club cluster in the protein interaction network of *Helicobacter Pylori*. As shown in this figure, these 2-club and 3-club clusters respectively contain a 0-club (single vertex) and a 1-club (single edge) kernels through which the interactions of proteins occur.

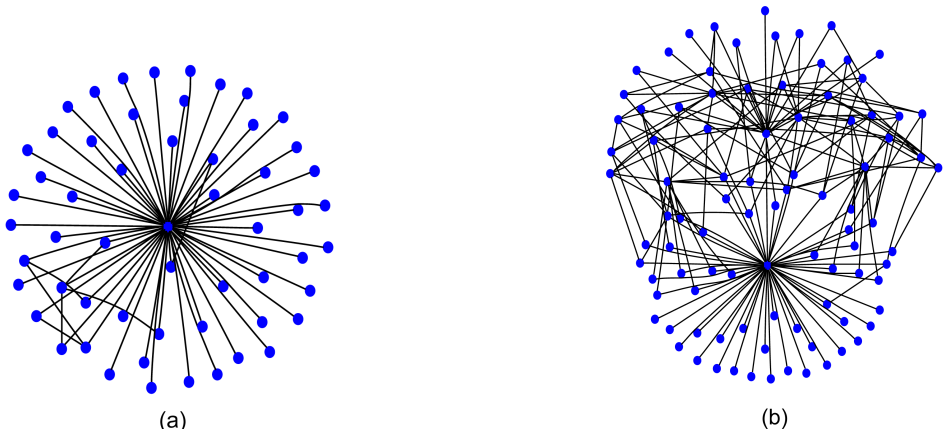

Figure 1.3: (a) A maximum 2-club cluster and (b) a large 3-club cluster in the protein interaction network of *Helicobacter Pylori*

CHAPTER 2

BACKGROUND*

This chapter provides some background on the k-club model, related optimization problems, and presents a review of available literature on this distance-based clique relaxation. The distance-based clique relaxations were introduced in social network analysis soon after cliques were introduced to model cohesive subgroups [40]. These models relaxed the adjacency requirement of cliques to a short path requirement. The k-clique model was originally introduced in social network analysis by Luce [41] to model cohesive subgroups. The k-club model was introduced by Alba [25] and further developed by Mokken [42], to address the drawback of k-cliques where members outside the cohesive subgroup can be used on the short paths required between members inside the group. Recall the formal definitions from Section 1.3. This chapter reviews the current literature on k-clubs, with a special focus on computational complexity, algorithms, and polyhedral results.

2.1 Complexity and approximation

The maximum k-clique and k-club problems are NP-hard for any *fixed* positive integer k [37, 43] and they remain NP-hard even in graphs of fixed diameter with $diam(G) > k$ [37]. Hence, the maximum k-club problem is known to be NP-hard for every fixed k even on graphs of diameter $k+1$. For given positive integers k and l ($k \neq l$), the

*Parts of this chapter will appear in B. Balasundaram and F. Mahdavi Pajouh: Graph-theoretic clique relaxations and applications. P. Pardalos, D-Z. Du and R. Graham (Eds) Handbook of Combinatorial Optimization, 2nd Edition ©Springer.

problem of recognizing whether $\overline{\omega}_k(G) = \overline{\omega}_l(G)$ is also NP-hard. This gap-recognition complexity result was further used to show that for an integer $k \geq 2$ and in a graph G with $\overline{\omega}_k(G) > \Delta(G) + 1$, unless P=NP, there does not exist a polynomial time algorithm for finding a k-club of size strictly larger than $\Delta(G) + 1$ [44]. Recalling that $\Delta(G)$ denotes the maximum vertex degree in G, it is clear that a vertex of maximum degree along with all its neighbors forms a k-club in G for any $k \geq 2$.

Unless P=NP, the maximum k-club problem was shown to be inapproximable within a factor of $n^{\frac{1}{3}-\epsilon}$ for any $\epsilon > 0$ [45], which has been strengthened to $n^{\frac{1}{2}-\epsilon}$ recently [46]. Approximation algorithms of factor $n^{\frac{1}{2}}$ and $n^{\frac{1}{3}}$ for even and odd k respectively, have also been proposed recently [46].

It is well known that cliques parameterized by solution size (identifying cliques of a particular size) is not fixed-parameter tractable, and in fact is a basic $W[1]$-hard problem [47]. An encouraging result for k-clubs is that it is fixed-parameter tractable when parameterized by solution size [48, 49]. The k-club problem is one of the few parameterized problems for which nonexistence of a polynomial-size many-to-one kernel and existence of a polynomial-size Turing kernel are known [48, 49]. The k-club problem can be solved on trees and interval graphs in $O(nk^2)$ and $O(n^2)$, respectively and it is polynomial time solvable on graphs with bounded tree- or cliquewidth [48]. The 2-club problem can be solved on bipartite graphs in $O(n^5)$ [48].

2.2 Polyhedral combinatorics

The maximum k-club problem admits the following integer programming formulation [43, 37].

$$\begin{aligned} \bar{\omega}_k(G) &= \max \sum_{i \in V} x_i \\ \textit{subject to:} & \\ x_i + x_j &\leq 1 + \sum_{l: P_{ij}^l \in \mathbb{P}_{ij}} y_{ij}^l \quad \forall \ (i,j) \notin E \\ x_p &\geq y_{ij}^l \quad \forall\ p \in V(P_{ij}^l),\ P_{ij}^l \in \mathbb{P}_{ij},\ (i,j) \notin E \\ x_i &\in \{0,1\} \quad \forall\ i \in V \\ y_{ij}^l &\in \{0,1\} \quad \forall\ P_{ij}^l \in \mathbb{P}_{ij},\ (i,j) \notin E \end{aligned}$$

where $\mathbb{P}_{ij}$ is an indexed collection of all paths of length at most k between vertices i, j in G and P_{ij}^l is the path with index l between vertices i, j. The formulation essentially ensures that if two vertices are in a k-club, then all the vertices in at least one path between them with length less than or equal to k are also included in the k-club. Note that the size of $\mathbb{P}_{ij}$ could be very large making this formulation difficult to handle. A more compact formulation available for the maximum 2-club problem is stated next.

$$\begin{aligned} \bar{\omega}_2(G) &= \max \sum_{i \in V} x_i \\ \textit{subject to:} & \\ x_i + x_j - \sum_{k \in N(i) \cap N(j)} x_k &\leq 1 \quad \forall \ (i,j) \notin E \\ x_i &\in \{0,1\} \quad \forall i \in V \end{aligned}$$

Given a graph $G = (V, E)$, the convex hull of the incidence vectors of k-clubs in G is called the k-club polytope of G denoted by $Q_k(G)$. Some preliminary results on these polytopes, especially the 2-club polytope are available from [37, 50].

Theorem 1 [37] presents the basic results on $Q_2(G)$ and the first family of facets developed.

Proposition 1 ([51]) *(Distance-k independent set inequalities) Given a graph $G = (V, E)$, let $I \subseteq V$ be a maximal k-independent set. Then the following inequality is valid for $Q_k(G)$.*

$$\sum_{i \in I} x_i \leq 1$$

Theorem 1 ([37]) *Consider the 2-club polytope $Q_2(G)$ of a graph $G = (V, E)$.*

a. $dim(Q_2(G)) = n$.

b. $x_i \geq 0$ induces a facet of $Q_2(G)$ for every $i \in V$.

c. For $i \in V$, $x_i \leq 1$ induces a facet of $Q_2(G)$ if and only if $d_G(i, j) \leq 2 \; \forall \; j \in V$.

d. The inequality $\sum_{i \in I} x_i \leq 1$ induces a facet of $Q_2(G)$ if and only if I is a maximal distance-2 independent set in G.

Notice that Theorem 1c is implied by Theorem 1d, which is analogous to Padberg's maximal independent inequalities for the clique polytope [52]. This line of research has been recently furthered in [50]. Theorem 2 provides necessary and sufficient conditions under which the common neighborhood constraint in the formulation is facet inducing upon strengthening. Theorem 3 introduces another facet defining inequality for $Q_2(G)$.

Theorem 2 ([50]) *Given a graph $G = (V, E)$, let $a, b \in V$ and $I \subseteq V \setminus \{a, b\}$ be such that $I \cup \{a\}$ and $I \cup \{b\}$ are distance-2 independent sets in G and $dist_G(a, b) = 2$. The following inequality is valid for $Q_2(G)$ and induces a facet if and only if I is a maximal set.*

$$\sum_{i \in I \cup \{a,b\}} x_i - \sum_{j \in N(a) \cap N(b)} x_j \leq 1$$

Theorem 3 ([50]) *Given a graph $G = (V, E)$, let $a, b, c \in V$ and $I \subseteq V \setminus \{a, b, c\}$ be such that set $\{a, b, c\}$ is independent and sets $I \cup \{a\}$, $I \cup \{b\}$ and $I \cup \{c\}$ are distance-2 independent in G. The following inequality is valid for $Q_2(G)$ and induces a facet if and only if I is a maximal set.*

$$\sum_{i \in I \cup \{a,b,c\}} x_i - \sum_{j \in V} \alpha_j x_j \leq 1$$

where

$$\alpha_j = \begin{cases} 2, & \text{if } j \in N(a) \cap N(b) \cap N(c), \\ 1, & \text{if } j \in [N(a) \cap N(b)] \cup [N(a) \cap N(c)] \cup [N(b) \cap N(c)] \setminus [N(a) \cap N(b) \cap N(c)], \\ 0, & \text{Otherwise.} \end{cases}$$

The k-club polytope for $k \geq 3$ has fewer associated results and there is ongoing research on polyhedral approaches to these cases. Alternate formulations and valid inequalities for the maximum 3-club problem were introduced along with computational experiments in [53, 54]. It should be noted that while the exponential formulation for the maximum k-club problem discussed here is unsuitable for explicit use with integer programming solvers for large k, a more compact formulation has been recently developed in [55] that permits such use. The approach taken here is to first formulate the problem as a nonlinear integer program followed by linearizing the products of binary variables resulting in a linear integer program which uses $O(kn^2)$ variables and constraints. The authors also introduce the notion of an r-robust k-club which is a subset of vertices S such that $G[S]$ has at least r internally vertex disjoint paths of length at most k (in $G[S]$) between every pair of vertices. This model allows the detection of low diameter clusters that also have higher vertex connectivity, and thereby more robust to vertex or edge deletions.

2.3 Algorithms

Combinatorial exact algorithms appear to be the preferred approach presently for solving the maximum k-club problem for arbitrary k. The only existing combinatorial algorithm from [43] is a classical branch-and-bound approach based on variable dichotomy. The approach taken in this work to obtain upper-bounds on $\overline{\omega}_k(G)$ is to solve the maximum k-clique problem at each node of the search tree. Note that, the maximum k-clique problem (which is also NP-hard) must be solved to optimality to produce a valid bound. Details on this algorithm can be found in [43]. A fixed parameter tractable algorithm is presented in [48] to find a k-club of size c if one exists in G in $O((c-2)^c \cdot c! \cdot c^3 n + nm)$ time.

The results on the 2-club polytope have also led to branch-and-cut approaches for solving the maximum 2-club problem incorporating maximal distance-2 independent set facets discussed in [37, 51]. A sequential cutting-plane method for solving this problem is also proposed in [50]. Heuristic algorithms for finding large k-cliques or k-clubs in a given graph have been proposed in [56, 50, 57, 58].

2.4 Research statement

The available theoretical and algorithmic results addressing k-club model and related optimization problems are limited. A fundamental challenge in the development of theory and algorithms for the k-club model is its nonhereditary nature. Unlike k-cliques, the k-club model is nonhereditary, meaning every subset of a k-club is not necessarily a k-club. While the nonhereditary nature of k-clubs has been noted in literature (see [42, 37]), the computational complexity of testing maximality of k-clubs has remained open. The known polyhedral results associated with k-clubs are limited to some trivial and very specific facet inducing inequalities. There is a need for discovering more facet defining inequalities for the k-club polytope and identify-

ing classes of graphs in which complete description of the k-club polytope is known. The available integer programming formulations for the maximum k-club problem suffer from drastic increase of the number of decision variables and constraints for larger values of k. Therefore, the preferred approach available for solving the maximum k-club problem for arbitrary k is a combinatorial branch-and-bound algorithm. The only available combinatorial branch-and-bound algorithm employs a bounding strategy that requires solving an NP-hard problem to optimality.

Additionally, the existing literature on the k-club model restrict their attention to deterministic graphs. There is a need to study k-clubs and related optimization problems on random graphs in which nodes/arcs have probabilities for complete/partial failure.

These facts motivate a theoretical and algorithmic study of the k-club model on deterministic and random graphs in order to enrich the literature on this distance-based clique relaxation model in the following four major areas: complexity, combinatorial algorithms, polyhedral studies, and the problem in a probabilistic setting. The specific research objectives (RO) addressed in this dissertation are as follows.

RO1. Investigate the computational complexity of testing the inclusionwise maximality of k-clubs on arbitrary and restricted graph classes.

RO2. Investigate new lower- and upper-bounding strategies to develop new combinatorial branch-and-bound algorithms for solving the maximum k-club problem and study their computational performance on benchmark instances.

RO3. Investigate the 2-club polytope in order to discover new facet inducing inequalities, and study their separation complexity. Further, explore the derivation of complete polyhedral description for special graph classes.

RO4. Study the maximum 2-club problem under uncertainty, specially in graphs subject to probabilistic edge failures, in order to develop a stochastic optimization

formulation and decomposition algorithms for the problem.

2.5 Outline of the dissertation

The remainder of this document is structured as follows. The computational complexity of k-clubs maximality testing, which has been open for almost a decade, is addressed in Chapter 3. The NP-completeness of k-clubs maximality testing is proved in this chapter and a class of graphs on which this problem is polynomial-time solvable is also presented. Chapter 4 presents new bounding strategies for the k-club number of a graph and provides a general framework for a combinatorial branch-and-bound algorithm for solving the maximum k-club problem. This chapter also includes the experimental results of solving the maximum k-club problem on a test-bed of benchmark graphs by utilizing the proposed branch-and-bound algorithm. The 2-club polytope is studied in Chapter 5 and a new family of facet inducing inequalities, which strictly includes all known nontrivial facets for this polytope, is introduced in this chapter. The separation complexity of the newly discovered facet inducing inequalities and the complete description of the 2-club polytope of trees are also addressed in this chapter. The focus of Chapter 6 is on the maximum 2-club problem under uncertainty. Here, we are interested in detecting large "risk-averse" 2-clubs in a network subject to probabilistic edge failures. Conditional Value-at-Risk (CVaR) is used as a quantitative measure of risk aversion and a new decomposition algorithm for solving CVaR constrained maximum 2-club problem is proposed. This chapter also includes preliminary numerical results to compare the computational performance of the developed algorithm with a recent algorithm from the literature. Chapter 7 presents the concluding remarks summarizing the contributions and identifies some directions for future research in this area.

CHAPTER 3

COMPLEXITY OF k-CLUB MAXIMALITY TESTING*

As discussed in Section 2.1, the maximum k-club problem is known to be NP-hard for any fixed positive integer k, while the computational complexity of the k-club maximality testing has remained open. Answering this question has important implications for developing theory and algorithms for k-clubs. The complexity of k-club maximality testing is addressed in this chapter and it is shown that this problem is also NP-hard. The NP-completeness of k-clubs maximality testing is a direct consequence of the property being nonhereditary. This observation will be discussed in detail in the next section before presenting the main complexity result. Additionally, a class of graphs on which maximality of a k-club is polynomially verifiable, is also introduced in this chapter.

3.1 The nonhereditary nature of k-clubs

A graph property Π is said to be *hereditary on induced subgraphs*, if every vertex induced subgraph of G satisfies Π whenever G satisfies Π. The fundamental challenge in the study of k-clubs is their lack of heredity. We explain this with examples and consider some basic consequences.

For any integer $k \geq 1$, the k-clique model is hereditary on induced subgraphs, since every subset of a k-clique is also a k-clique. This property however does not

*Parts of this chapter are reprinted with permission from F. Mahdavi Pajouh and B. Balasundaram: On inclusionwise maximal and maximum cardinality k-clubs in graphs. Discrete Optimization, 2012, DOI: http://dx.doi.org/10.1016/j.disopt.2012.02.002 ©Elsevier.

hold for the k-club model for $k \geq 2$. For example, in the graph shown in Figure 1.2, the set $\{1, 2, 4, 5, 6\}$ is a 2-club which means it is also a 2-clique. Every subset of this set is a 2-clique but subset $\{1, 2, 4, 5\}$ for instance is not a 2-club.

Inclusionwise maximality of any polynomially verifiable hereditary property such as k-clique can be tested in polynomial time. For example, to verify maximality of a k-clique S, it suffices to show that there is no single vertex in $V \setminus S$ that could be added to S to form a larger k-clique. However, for the k-club model nonexistence of a vertex that could increase the size of the k-club by one is a necessary but not sufficient condition for its maximality. For example, in the graph shown in Figure 3.1(a), set $S_1 = \{1, 2, 3\}$ is a 2-clique and since sets $S_2 = S_1 \cup \{4\}$ and $S_3 = S_1 \cup \{5\}$ are not 2-cliques, it can be concluded that S_1 is a maximal 2-clique. In Figure 3.1(b), S_1 is not a maximal 2-clique which can be deduced by observing that both S_2 and S_3 are also 2-cliques. Set S_1 in the graph shown in Figure 3.1(b) forms a 2-club while sets S_2 and S_3 are not 2-clubs. But S_1 is not maximal because set $V = S_1 \cup \{4, 5\}$ is a larger 2-club containing S_1.

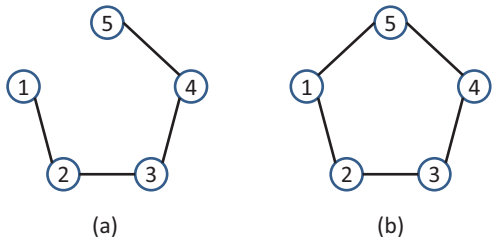

Figure 3.1: Inclusionwise maximality testing of 2-cliques and 2-clubs

3.2 NP-completeness of k-club maximality testing

An instance of k-CLUB MAXIMALITY TESTING is given by a simple undirected graph $G = (V, E)$ and a k-club D in G, and we ask if there exists a k-club D' in G such

that $D \subset D'$? The following theorem establishes that k-CLUB MAXIMALITY TESTING (k-CMT) is NP-complete.

Theorem 4 *k-club maximality testing is NP-complete for any fixed integer $k \geq 2$.*

Proof. We prove this theorem by a polynomial-time reduction from 3-SAT [23]. We assume that the 3-SAT instances satisfy the following restrictions: (a) No clause contains a literal and its negation; (b) There are at least 3 clauses in the 3-SAT formula. Note that the 3-SAT problem is still NP-complete under these restrictions. Before providing the transformation, we present some terminology that we use in the proof. The construction is slightly different depending on whether k is odd or even. A *k-chain* is a path of length k (on $k+1$ nodes). When k is even, the $(\frac{k}{2}+1)^{th}$ node is called the *midpoint* of the k-chain. When k is odd, the $\frac{k+1}{2}{}^{th}$ and $\frac{k+3}{2}{}^{th}$ nodes are called *midpoints*. Define $q = \frac{1}{2}[k-2+(k \mod 2)]$. A q-pendant is a path of length q. One endpoint of a q-pendant is called the *head* and the other is called the *tail*. The node preceding the tail on the path from head to tail is called the *penultimate* node. For a pendant or a chain P containing vertices v_1, v_2, by $P[v_1, v_2]$ we denote the subpath from v_1 to v_2 including both end-points. We use the convention that a single vertex path is a path of length zero. Denote the 3-SAT instance with n boolean variables, m clauses with $3m$ literals as $B = \bigwedge_{i=1}^{m}(p_{i1} \vee p_{i2} \vee p_{i3})$. For any such 3-SAT instance $< B >$, the following steps construct in polynomial time, an instance $< G, D >$ for k-CMT such that the 3-SAT instance is satisfiable if and only if D is not a maximal k-club in G.

Construction

1. For each clause $i = 1, \ldots, m$ in B, G contains a clique of size 3, with nodes labeled by the literals p_{ij}. We call these nodes in G, the *literal nodes.*

2. Every pair of literal nodes p_{ij} and p_{uv} that belong to different clauses ($i \neq u$)

and are not negations of each other ($p_{ij} \neq \overline{p}_{uv}$) are connected by a k-chain. Such pairs of literal nodes are called *dcnn-pairs*. Each such k-chain consists of $k-1$ new internal nodes with the dcnn-pair forming the endpoints. The internal nodes created will be called the *chain nodes*.

3. If k is even, we add all possible edges among the midpoints of all the k-chains so that they form a clique. If k is odd, we create a new node called the *nucleus node* and both midpoints of every k-chain are made adjacent to the nucleus node.

4. We always traverse all k-chains going from a lower index clause to a higher index clause, with the exception of k-chains between two clauses 1 and m. We traverse these k-chains going from clause m to clause 1. This imposes an orientation to these paths allowing us to refer to *succeeding* or *preceding* nodes on a path without ambiguity. We follow this convention in the remainder of this proof. Note that G is an undirected graph, and the direction is simply for path traversal. We refer to a k-chain that connects two literal nodes from two consecutive clauses $i, i+1$ as a *k-bridge* for $i = 1, \ldots, m-1$. The k-chains connecting literal nodes from clauses m to 1 are also called k-bridges. The set of all nodes thus created (literal nodes, chain nodes and the nucleus in case of odd k) are denoted by U. Note that $|U| = \text{total \# dcnn-pairs} \times (k-1) + 3m + (k \mod 2)$.

5. To every node $v \in U$, a q-pendant is attached by making its head adjacent to v. We refer to such a q-pendant as the *connector pendant of v*. We *associate* with every node $v \in U$, a q-pendant called the *opposing pendant of v* (can be taken to mean "indexed by v" as the opposing pendant of v will not be attached to v). Each opposing pendant will be connected to nodes in U once the notion of a *supporting node* is defined in the next step. Note that each such q-pendant of

length q from head to tail results in the creation of $q+1$ new nodes. We denote these q-pendant nodes by D where $|D| = 2(q+1)|U|$.

6. For two chain nodes i, j, we say i *supports* j if i immediately precedes j on some k-chain. Every literal node supports the first chain node on every k-chain starting from it, and it is supported by the last chain node on every k-bridge ending at it. In case of odd k, we say the nucleus node is supported by all the chain midpoints that are adjacent to it. So in set U, every chain node has exactly one supporting node. Every literal node p_{ij} corresponding to clause i $(i = 2, \ldots, m)$, has as many supporters as there are dcnn-pairs of p_{ij} in clause $i-1$. Every literal node p_{1j} has as many supporters as there are dcnn-pairs of p_{1j} in clause m. Every literal node p_{ij} corresponding to clause i $(i = 2, \ldots, m-1)$ supports as many chain nodes as there are dcnn-pairs of p_{ij} in clauses $i+1, \ldots, m$. Every literal node p_{1j} supports as many chain nodes as there are dcnn-pairs of p_{1j} in clauses $2, ..., m-1$ and every literal node p_{mj} supports as many chain nodes as there are dcnn-pairs of p_{mj} in clause 1.

7. Consider any k-bridge connecting literal node p_{ij} to $p_{i+1,v}$ (including k-bridges from p_{mj} to p_{1v}) with chain nodes $a_1, \ldots, a_{k-1}$. Each node in the subpath from p_{ij} to a_{k-1} is made adjacent to the head of the opposing pendant of the node it supports.

8. Consider any k-chain connecting literal node p_{ij} to p_{uv} (traversed from clause i to clause u) that is not a k-bridge with chain nodes $a_1, \ldots, a_{k-1}$. Each node in the subpath from p_{ij} to a_{k-2} is made adjacent to the head of the opposing pendant of the node it supports. Note that in this case, we stop at the chain node a_{k-2} since a_{k-1} doesn't support p_{uv}.

9. If k is odd, all midpoints adjacent to the nucleus are also made adjacent to the head of the nucleus' opposing pendant.

10. If k is even, for each $v \in U$, the tail of connector pendant of v is made adjacent to the tail of every other q-pendant except the tail of the opposing pendant of v. For odd k, for each $v \in U$, the penultimate node of connector pendant of v is made adjacent to the tail of every other q-pendant except the tail of the opposing pendant of v. For odd k, we also create a clique among tail nodes of all opposing pendants.

This completes the construction of graph $G = (D \cup U, E)$. Figure 3.2 illustrates the construction for even and odd k. We establish the desired NP-completeness result through a sequence of three claims. These claims essentially follow from construction and are provided for the sake of clarity since the construction is somewhat elaborate. The proofs for these claims are provided in Appendix A.

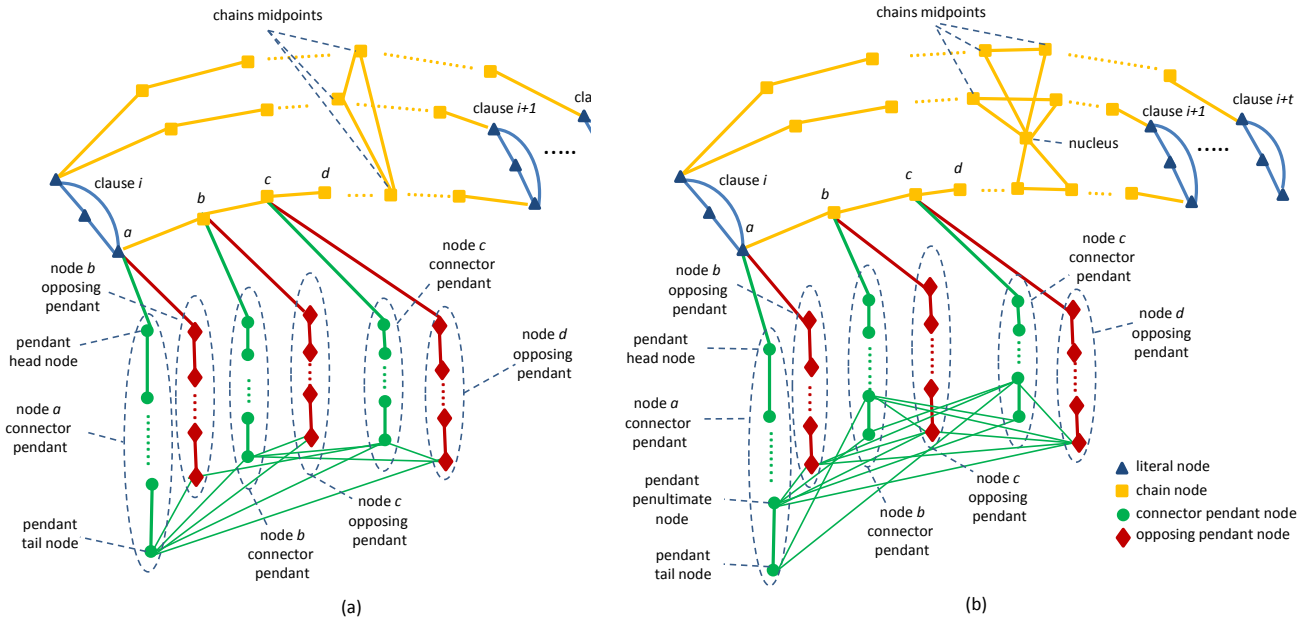

Figure 3.2: Illustration of the construction for (a) even k and (b) odd k

Claim 1 *D is a k-club.*

Claim 2 *For any $v_1 \in U$:*

a. If $v_2 \in D$ is the head of v_1's opposing pendant then $d_{G[D \cup \{v_1\}]}(v_1, v_2) > k$.

b. *If $v_2 \in D$ is not the head of v_1's opposing pendant then $d_{G[D\cup\{v_1\}]}(v_1, v_2) \leq k$.*

c. *If $v_2 \in D$ is the head of v_1's opposing pendant then for a set $S \subseteq U$, $d_{G[D\cup S\cup\{v_1\}]}(v_1, v_2) \leq k$, if and only if set S contains at least one supporter of v_1.*

Claim 3 *For any two literal nodes $v_1, v_2 \in U$, $d_G(v_1, v_2) > k$ if and only if $v_1 = \overline{v}_2$.*

We now show that the 3-SAT instance is satisfiable if and only if D is not a maximal k-club in G. Suppose the 3-SAT instance has a satisfying assignment. Let S be the set of all literal nodes corresponding to literals with value equal to 1 in this assignment and all chain nodes in all k-chains connecting them. For odd k, S also contains nucleus node. $S \subseteq U$ and contains at least one literal node from each clause. For every literal node in S, there exists at least one k-chain with this literal node as one of its endpoints. We show that $S \cup D$ is a k-club in G.

Consider any two literal nodes $v_1, v_2 \in S$, either they belong to the same clause or they are from two different clauses. In the first case, v_1 and v_2 are adjacent. In the second case, v_1 and v_2 form a dcnn-pair and there is a k-chain connecting them in $G[D \cup S]$.

Now let v_1 be a literal node in S and v_2 be any chain node in this set. Either v_2 belongs to a k-chain that has v_1 as one of its endpoints or not. In the first case, it is easy to see that $d_{G[D\cup S]}(v_1, v_2) \leq k$. In the second case, let c_1 be a k-chain that has v_1 as one of its endpoints and c_2 denote the k-chain which contains v_2. For even k, consider the path $c_1[v_1, midpoint(c_1)] - c_2[midpoint(c_2), v_2]$ in $G[D \cup S]$. The length of this path is less than or equal to $k/2 + 1 + k/2 - 1 = k$. For odd k, without loss of generality, consider the closest midpoint to v_1 and v_2 on c_1 and c_2 respectively, that are both adjacent to the nucleus. This yields a path of length less than or equal to $\frac{k-1}{2} + 2 + (\frac{k-1}{2} - 1) = k$.

Now suppose v_1 and v_2 are two different chain nodes in S. They either belong to the same k-chain or they belong to two different k-chains c_1 and c_2. In the first case,

it is easy to see that $d_{G[D\cup S]}(v_1, v_2) \leq k$. For the second case, for an even k, consider the path $c_1[v_1, midpoint(c_1)] - c_2[midpoint(c_2), v_2]$. The length of this path is at most $k/2 - 1 + 1 + k/2 - 1 = k - 1$. For an odd k, consider the closest midpoint to v_1 and v_2 on c_1 and c_2 respectively. The path through these nodes and the nucleus is of length at most $(\frac{k-1}{2} - 1) + 2 + (\frac{k-1}{2} - 1) = k - 1$. So for every two nodes $v_1, v_2 \in S$, $d_{G[D\cup S]}(v_1, v_2) \leq k$.

Consider any literal node v_1 in clause i, $2 \leq i \leq m$. S contains a literal node v_2 from clause $i - 1$ which forms a dcnn-pair with v_1. In case of $i = 1$, it contains such a dcnn-pair literal node from clause m. Because the k-chain connecting v_1 and v_2 is also in S, set S contains at least one supporter of v_1. Now consider any chain node $v_1 \in S$, since the k-chain containing v_1 is also in S, the supporter of v_1 is also available in S. So for any $v_1 \in S$, S contains at least one supporter of v_1. So by Claim 2, $d_{G[D\cup S]}(v_1, v_2) \leq k$ for each $v_1 \in S, v_2 \in D$. By Claim 1, $d_{G[D\cup S]}(v_1, v_2) \leq k$ for each $v_1, v_2 \in D$. So $G[D \cup S]$ is a k-club containing D and because $S \neq \emptyset$, D is not maximal.

Suppose D is not a maximal k-club and there exists a nonempty $S \subseteq U$ such that $S \cup D$ is a k-club. Select an arbitrary node $v_1 \in S$. v_1 is either a literal node or a chain node. If it is a chain node, by Claim 2c, S should contain a supporter of v_1. This supporter is again either a literal node or another chain node. By repeating this argument, we can conclude that S should contain at least one literal node v_1'. Suppose v_1' is located in clause i. Again by Claim 2c, the supporter of v_1' which is a chain node in some k-bridge should also be in S which results in the following conclusion. If $2 \leq i \leq m$, S should contain a literal node v_2 in clause $i - 1$ (in case of $i = 1$, the literal node v_2 is from clause m). Now by repeating this argument, we can conclude that set S should contain at least one literal node from each clause in B. Let $v_1, v_2 \in S$ be any two literal nodes. Since $d_{G[D\cup S]}(v_1, v_2) \leq k$, we can conclude that $d_G(v_1, v_2) \leq k$. Now by Claim 3, we have $v_1 \neq \overline{v_2}$. So by setting the value of the

literals corresponding to literal nodes in S to 1 and the rest of them to zero, we will have a satisfying assignment for B. This establishes that k-club maximality testing is NP-hard. It is easy to see that the problem is in class NP. This completes the proof of Theorem 4. ■

Contrasting this result with the polynomial-time solvability clique (and k-clique) maximality testing, it is essential to understand the reason for Theorem 4. Despite the conceptual similarity of k-clubs to k-cliques and cliques, there is a fundamental characteristic that distinguishes k-clubs. Cliques and k-cliques are hereditary properties as every subset of a k-clique is also a k-clique, for every fixed positive integer k. This allows us to verify maximality by inclusion of a k-clique C in polynomial time using the following algorithm. Let $C' = C$, pick any $v \in V \setminus C'$, if $C' \cup \{v\}$ is a k-clique, then add v to C', otherwise delete v from V and repeat until $V \setminus C' = \emptyset$. At termination, C' is a maximal k-clique containing C. However, this approach is not valid for finding maximal k-clubs for $k \geq 2$ as the k-club property is not hereditary. Consider the graph in Figure 1.2. Set $\{1, 5, 6\}$ is a 2-club in this graph. The sets $\{1, 5, 6, 4\}, \{1, 5, 6, 3\}, \{1, 5, 6, 2\}$ however are not 2-clubs. But the maximal 2-club containing $\{1, 5, 6\}$ is $\{1, 5, 6, 2, 4\}$.

3.3 Some implications of Theorem 4

As mentioned in Section 3.1, a graph property Π is said to be hereditary on induced subgraphs, if G is a graph with property Π then every vertex induced subgraph of G also satisfies property Π. Further, property Π is said to be *nontrivial* if it is true for a single vertex graph and is not satisfied by every graph. A property is said to be *interesting* if there are arbitrarily large graphs satisfying Π. Yannakakis [59] showed that the *maximum* Π *problem* to find the largest order induced subgraph that does not violate property Π is NP-hard for any property Π that is nontrivial,

interesting and hereditary on induced subgraphs. This result has a very broad scope as complete subgraphs, edgeless subgraphs, planar subgraphs, bipartite subgraphs, perfect subgraphs are examples of Π that are nontrivial, interesting and hereditary.

If Π is hereditary and $G = (V, E)$ is an arbitrary graph with $V \supseteq C' \supset C$ such that $G[C]$ and $G[C']$ satisfy Π, the elements in $C' \setminus C$ can be added to C in any order with all the intermediate sets inducing subgraphs satisfying Π. Hence, if Π is hereditary, proving maximality by inclusion of $C \subseteq V$ such that $G[C]$ satisfies Π is equivalent to establishing the nonexistence of any single vertex $v \in V \setminus C$ such that $G[C \cup \{v\}]$ satisfies Π. Our result in Section 3.2 shows that there is no polynomial-time algorithm available to test k-club maximality unless $P = NP$. Hence, there could be no generic polynomial-time algorithm (as the one described above for hereditary properties) unless $P = NP$, to test maximality by inclusion of subgraphs satisfying any nonhereditary property.

In addition to the complexity theoretic perspective afforded by the result of Yannakakis, we can also view the impact of nonhereditary graph properties from a matroid theory perspective. A combinatorial system described by the pair $M = (S, \mathcal{I})$ where S is a finite ground set and $\mathcal{I}$ is a collection of subsets of S satisfying some specified property Π, is a matroid if the following three axioms hold: (M0) $\emptyset \in \mathcal{F}$; (M1) If $J' \subseteq J \in \mathcal{F}$, then $J' \in \mathcal{F}$; (M2) For every $J \in \mathcal{F}$, every maximal (by inclusion) subset in $\mathcal{F}$ containing J has the same cardinality. A fundamental result in matroid theory is that the maximum Π problem for a matroid can be solved in polynomial time using the greedy algorithm [60]. If M satisfies axioms (M0) and (M1), it is called an *independence system* and the maximum Π problem in general is NP-hard on independence systems [61]. Note that collection of cliques in a graph form an independence system. In contrast, the greedy algorithm can be used to find a maximal by inclusion subset satisfying Π in polynomial time. In light of Theorem 4, combinatorial systems built on nonhereditary properties are much harder to deal with than

matroids or independence systems.

3.4 Graphs on which k-club maximality is polynomially verifiable

In this section, we characterize a class of graphs on which k-club maximality testing is polynomial-time solvable. Let us refer to a k-clique which induces a connected subgraph as a *connected k-clique*. In a given graph, if every connected k-clique is also a k-club then a k-club's maximality can be checked using a method similar to k-clique maximality testing. Since, by definition every k-club is a connected k-clique and in such graphs every connected k-clique is also a k-club, a maximal connected k-clique is also a maximal k-club. Given a graph $G = (V, E)$, maximality of a connected k-clique D can be checked by testing nodes in $V \setminus D$ that have at least one edge to some node in D, one at a time, to see if they can be added to D. In order to specify a class of graphs with polynomially verifiable maximal k-clubs, we need to define an *asymmetric partitionable cycle* as follows.

Definition 3.1 *A graph $G = (V, E)$ is called a partitionable cycle, if it contains a spanning cycle W and a pair of nonadjacent nodes i and j such that every edge in $E \setminus E(W)$ has one endpoint in $A_W(i, j)$ and the other endpoint in $A_W(j, i)$, where $A_W(i, j)$ and $A_W(j, i)$ are the internal nodes on the two paths between i and j in W. A partitionable cycle is said to be asymmetric if $|A_W(i, j)| \neq |A_W(j, i)|$. (See Figure 3.3)*

Theorem 5 characterizes the class of graphs on which maximality of a k-club can be verified in polynomial-time.

Theorem 5 *Given a graph $G = (V, E)$ and fixed integer $k \geq 2$, every connected k-clique is a k-club if G does not contain an asymmetric partitionable cycle on c vertices as an induced subgraph, where $5 \leq c \leq 2k + 1$.*

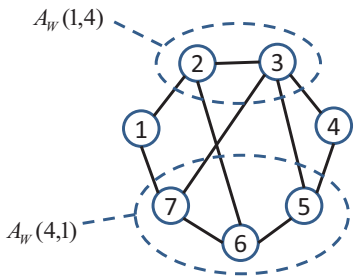

Figure 3.3: An asymmetric partitionable cycle with respect to nodes 1 and 4, and $W = 1 - 2 - \cdots - 7 - 1$

Proof. Suppose a given graph $G = (V, E)$ contains a connected k-clique D which is not a k-club. Then there exist $i, j \in D$ such that $2 \leq d_G(i, j) \leq k$ but $d_{G[D]}(i, j) = k+l < \infty$ where $1 \leq l < \infty$. Let $i = i_0 - i_1 - \cdots - i_{k+l} = j$ be a shortest path between i and j in $G[D]$. Now $P_1 : i = i_0 - i_1 - \cdots - i_{k+1}$ is a shortest path between i and i_{k+1} in $G[D]$ and $d_{G[D]}(i, i_{k+1}) = k + 1$. Since $i, i_{k+1} \in D$, there must be a shortest path P_2 between i and i_{k+1} in G with length $2 \leq l' \leq k$. Since P_1 is a shortest path in $G[D]$ and P_2 is shorter than P_1, there exists node $v \in P_2$ such that $v \notin D$ and therefore $v \notin P_1$.

Now consider $G[V(P_1) \cup V(P_2)]$. Moving along P_1 from i to i_{k+1}, let S denote the sequence of nodes in P_1 which also belong to P_2. So we have $S : i = s_0, s_1, \ldots, s_p = i_{k+1}$. If $(s_j, s_{j+1}) \in E$ for a $j \in \{0, \ldots, p-1\}$ then $(s_j, s_{j+1}) \in P_1$ and $(s_j, s_{j+1}) \in P_2$ because otherwise P_1 and P_2 are not shortest paths between i and i_{k+1} in $G[D]$ and G respectively. Further, if $(s_j, s_{j+1}) \in E$ for all $j \in \{0, \ldots, p-1\}$ then $S = P_1 = P_2$ which is a contradiction since length of P_2 is strictly less than length of P_1. So there exist $h \in \{0, \ldots, p-1\}$ such that $(s_h, s_{h+1}) \notin E$. Since $s_h, s_{h+1} \in P_1$, there exists at least one node between s_h and s_{h+1} on P_1. Let I_1 denote the set of all such nodes. Similarly for P_2, there exists at least one node between s_h and s_{h+1} on P_2 and let I_2 denote the set of all such nodes. $I_1 \cap I_2 = \emptyset$ because moving along P_1 from i to i_{k+1}, s_{h+1} is the first node in P_2 after s_h. $G[\{s_h, s_{h+1}\} \cup I_1 \cup I_2]$ is a partitionable cycle with respect to s_h, s_{h+1}, hence, $G[V(P_1) \cup V(P_2)]$ contains at least one partitionable

cycle.

Suppose every partitionable cycle in $G[V(P_1) \cup V(P_2)]$ is symmetric. Like before, moving along P_1 from i to i_{k+1}, let S be the sequence of the nodes in P_1 which are also in P_2 and moving along P_2 from i to i_{k+1}, let T be the sequence of the nodes in P_2 which are also in P_1. So $S : i = s_0, s_1, ..., s_p = i_{k+1}$ and $T : i = t_0, t_1, \ldots, t_p = i_{k+1}$. The elements of S and T are the same but their sequence might be different. We know length of $P_1 = \sum_{j=0}^{p-1} d_{G[D]}(s_j, s_{j+1})$ and length of $P_2 = \sum_{j=0}^{p-1} d_G(t_j, t_{j+1})$.

Now suppose the sequences S and T are the same, that is $s_j = t_j$ for each $j \in \{0, \ldots, p\}$. So length of $P_1 = \sum_{j=0}^{p-1} d_{G[D]}(s_j, s_{j+1})$ and length of $P_2 = \sum_{j=0}^{p-1} d_G(s_j, s_{j+1})$. Now if for some $j \in \{0, \ldots, p-1\}$, $(s_j, s_{j+1}) \in E$ then $d_{G[D]}(s_j, s_{j+1}) = d_G(s_j, s_{j+1}) = 1$ and if for some $j \in \{0, \ldots, p-1\}$, $(s_j, s_{j+1}) \notin E$, according to the previous discussion, there exists a partitionable cycle W with respect to s_j, s_{j+1}. Let $A_W(s_j, s_{j+1})$ contains the nodes between s_j and s_{j+1} on P_1 and $A_W(s_{j+1}, s_j)$ contains the nodes between s_j and s_{j+1} on P_2. We know $d_{G[D]}(s_j, s_{j+1}) = |A_W(s_j, s_{j+1})|+1$ and $d_G(s_j, s_{j+1}) = |A_W(s_{j+1}, s_j)|+1$. Since every partitionable cycle is symmetric, we have $|A_W(s_j, s_{j+1})| = |A_W(s_{j+1}, s_j)|$ which results in $d_{G[D]}(s_j, s_{j+1}) = d_G(s_j, s_{j+1})$. So finally we have length of P_1 is equal to the length of P_2 which is a contradiction.

Now suppose sequences S and T are different. Let s_j be the first node in sequence S which is different from the corresponding node in sequence T (which is t_j). Note that s_j is located after t_j in T and t_j is located after s_j in S. Hence, $d_{G[D]}(s_{j-1}, t_j) = d_{G[D]}(s_{j-1}, s_j) + d_{G[D]}(s_j, t_j)$ and since $d_{G[D]}(s_j, t_j) > 0$ we have,

$$d_{G[D]}(s_{j-1}, t_j) > d_{G[D]}(s_{j-1}, s_j). \tag{3.1}$$

Considering T, we know t_j is the first node after s_{j-1} and $(s_{j-1}, t_j) \notin E$ as otherwise P_1 is not a shortest path since s_j is located between s_{j-1} and t_j in P_1. So according to the previous discussion, there exists a partitionable cycle W with respect to s_{j-1}, t_j. Let $A_W(s_{j-1}, t_j)$ contain the nodes between s_{j-1} and t_j in P_1 and $A_W(t_j, s_{j-1})$ contain the nodes between s_{j-1} and t_j in P_2. Since every partitionable cycle is symmetric,

we have $|A_W(s_{j-1}, t_j)| = |A_W(t_j, s_{j-1})|$. So $d_{G[D]}(s_{j-1}, t_j) = d_G(s_{j-1}, t_j)$. Now using Equation 3.1 we have,

$$d_G(s_{j-1}, t_j) > d_{G[D]}(s_{j-1}, s_j). \tag{3.2}$$

In T, s_j is located after t_j. So,

$$d_G(s_{j-1}, t_j) < d_G(s_{j-1}, s_j). \tag{3.3}$$

Equations 3.2 and 3.3 imply $d_G(s_{j-1}, s_j) > d_{G[D]}(s_{j-1}, s_j)$ which is impossible.

So in $G[V(P_1) \cup V(P_2)]$, there must exists at least one asymmetric partitionable cycle. Since the length of P_1 is $k+1$ and maximum length of P_2 is k, the order of such an asymmetric partitionable cycle is at most $(k+1)+k = 2k+1$. On the other hand, it is at least 5 because graphs of smaller order will never form an asymmetric partitionable cycle.

■

Corollary 1 shows that a 2-club maximality can be polynomially verified in bipartite graphs.

Corollary 1 *In a bipartite graph $G = (X \cup Y, E)$, every connected 2-clique is a 2-club. Furthermore, every 2-club induces a complete bipartite subgraph.*

Proof. Since G is bipartite, it does not contain an odd cycle and hence, does not contain an asymmetric partitionable cycle of size 5. Any 2-club in G of size at least two, contains vertices $x \in X$ and $y \in Y$. If $(x, y) \notin E$ then $d_G(x, y) \geq 3$. Hence, the 2-club must induce a complete bipartite subgraph. ■

CHAPTER 4

COMBINATORIAL BRANCH-AND-BOUND FOR THE MAXIMUM k-CLUB PROBLEM*

In this chapter, new lower- and upper-bounding techniques for the k-club number of a graph are presented. A general framework for a combinatorial branch-and-bound (BB) algorithm for solving the maximum k-club problem is also developed and the experimental results from solving this problem on a test-bed of benchmark graphs by utilizing the proposed BB algorithm are presented.

First, we outline some algorithmic implications of the nonhereditary nature of k-clubs, especially for combinatorial algorithms. Since the k-club definition is more restrictive than the k-clique, it is possible that a maximal k-club is not a maximal k-clique. For instance, $\{1, 2, 3, 4\}$ is a maximal 2-club in Figure 1.2, but not a maximal 2-clique as it is contained in 2-clique $\{1, 2, 3, 4, 5\}$. On the other hand, if a maximal k-clique satisfies the diameter requirement it is also a maximal k-club. However, it is possible in a graph for $k \geq 2$ that no maximal k-clique is a k-club. Figure 4.1 shows a graph with exactly two maximal 2-cliques $\{1, 2, 3, 4, 5, 6, 7\}$ and $\{1, 2, 3, 5, 6, 7, 8\}$, neither of which is a 2-club. Hence, even enumerating all maximal k-cliques in the graph to select the ones satisfying the diameter-k condition may not identify a maximum k-club. Furthermore, in the extreme case, fail to identify a single k-club.

Furthermore, simple extensions of well known exact algorithms for maximum

*Parts of this chapter are reprinted with permission from F. Mahdavi Pajouh and B. Balasundaram: On inclusionwise maximal and maximum cardinality k-clubs in graphs. Discrete Optimization, 2012, DOI: http://dx.doi.org/10.1016/j.disopt.2012.02.002 ©Elsevier.

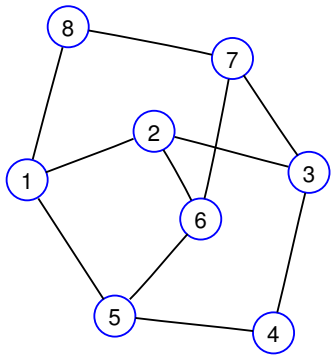

Figure 4.1: A graph in which every maximal 2-clique is not a 2-club

cliques such as the Carraghan-Pardalos algorithm [62], or the Östergård's algorithm [63] are not likely. The Carraghan-Pardalos algorithm is a back-tracking algorithm (a depth-first search order implicit enumeration) which requires that a maximal clique be found prior to back-tracking, which cannot be accomplished in polynomial time for k-clubs unless $P = NP$. Given $G = (V, E)$ with $V = \{1, \ldots, n\}$, Östergård's clique algorithm relies on the bounded increase in the size of a maximum clique going from $G[V_{i+1}]$ to $G[V_i]$ where $V_i = \{i, \ldots, n\}$. This is true for cliques as the size can at most go up by one vertex. Now consider the graph $G = (V, E)$ where $V = \{1, \ldots, n\}$ and $E = \{(1, i) : i = 2, \cdots, n\}$, *i.e.,* G is a *star graph* with central vertex 1 and leaves $2, \ldots, n$. The 2-club number of $G[V_i] = 1$ for each $i = 2, \ldots, n$, but $\bar{\omega}_2(G[V_1]) = n$. Even designing a basic branch-and-bound where the branching is carried out by a selected vertex being included or excluded to create two child nodes, presents interesting challenges. For instance, at some node of the search tree, let F^1 be the set of nodes fixed to be included and F^0 be the set of nodes deleted from consideration. For $k \geq 2$, we cannot fathom by infeasibility if $G[F^1]$ is not a k-club, as addition of vertices from $V \setminus \{F^0 \cup F^1\}$ could turn it into one. Likewise, if $G[F^1]$ is a k-club, we cannot fathom by feasibility unless we know it is indeed a maximal k-club. Hence, in the algorithm developed in Section 4.2, the pruning of the search tree is accomplished by using bounds in combination with other necessary conditions.

4.1 Bounding strategies for the k-club number of a graph

As discussed in Section 2.3, the only available exact combinatorial algorithm for solving the maximum k-club problem for $k \geq 2$ is the BB algorithm presented in [43]. In this approach, the authors find the k-clique number of the graph associated with the node of the BB tree to obtain an upper-bound. Note that the maximum k-clique problem must be solved to optimality to obtain a valid upper-bound, and it is NP-hard even on graphs of fixed diameter $k+1$ [37]. In the next section, we consider a dual to the maximum k-club problem so that any feasible solution of this dual could be used to obtain a valid upper-bound. In Section 4.1.2, we discuss a lower-bounding technique that builds on existing heuristic approaches for the problem [56].

4.1.1 Distance k-coloring based upper-bounding technique

In order to present the new upper-bounding approach, we need to provide the definition for a proper distance k-coloring of a graph as follows.

Definition 4.1 *A proper distance k-coloring of $G = (V, E)$ is a map $c : V \longrightarrow \{1, \ldots, n\}$ such that for every pair of vertices $u, v \in V$, if $c(u) = c(v)$ then $d_G(u, v) \geq k + 1$ [64]. For each $v \in V$, $c(v)$ is called the color of v, and the subset of nodes receiving the same color form a color class.*

Note that this definition reduces to classical graph coloring when $k = 1$ and is restrictive as k increases. Furthermore, every color class forms a k-independent set and hence, any k-club or k-clique in G can contain at most one vertex from a color class for every proper distance k-coloring of G. Denote the *minimum* number of colors to properly distance k-color G by $\chi_k^d(G)$. We have the following duality relationship for every fixed positive integer k:

$$\overline{\omega}_k(G) \leq \widetilde{\omega}_k(G) \leq \chi_k^d(G). \tag{4.1}$$

Figure 4.2 shows an example of a proper distance 2-coloring of a graph using 5 colors. Finding a minimum distance k-coloring is also an NP-hard problem [64], but for obtaining valid upper-bounds, we only need a proper distance k-coloring. By noting that distance k-coloring is equivalent to classical graph coloring on power graphs, we can employ fast coloring heuristics to yield a proper distance k-coloring [65]. A compromise must be made in the quality of the bound, as k-clique number which is a tighter bound can be hard to obtain, especially at the root node of the BB tree.

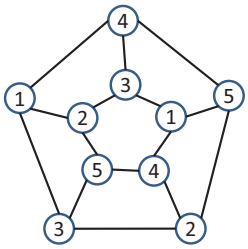

Figure 4.2: A proper distance 2-coloring (each number represents an specific color class)

4.1.2 Bounded enumeration based lower-bounding technique

The proposed lower-bounding technique is based on finding an initial k-club S, followed by a tree search that enumerates k-clubs containing S. If left unchecked, this worst-case complete enumeration would return a maximal k-club containing S. However, we employ pruning and termination criteria that would prevent this procedure from consuming an excessive amount of time. The "drop" and "constellation" heuristics described in [56] are used to identify the initial k-club. Both heuristics are run and the larger of the two k-clubs found is used as the initial solution S. The goal of the bounded enumeration search is to improve upon this initial solution if possible by spending a reasonable amount of time.

Given an initial k-club S, consider the graph H initially equal to G. We recursively check H for a vertex v such that $d_H(v, u) > k$ for some $u \in S$ and delete v. Note that

v cannot be a part of any k-club in G containing S. When this recursive procedure terminates, $S \subseteq V(H)$ and $d_H(v,u) \leq k$ for all $u \in S$ and $v \in V(H) \setminus S$. Every k-club in G that contains S is contained in $V(H)$, and we call $V(H) \setminus S$ the candidate set.

In this bounded enumeration search, at each node of the search tree, the corresponding graph $\mathcal{G} = G[F \cup U]$ consists of two types of vertices. The first is the set of vertices fixed to be included denoted by F, the second is the set of unexplored vertices denoted by U. The root node of the search tree is initialized by setting $F = S$ and $U = V(H) \setminus S$. The search order is depth-first search (DFS), with a vertex in U with the largest number of vertices at distance k or less in $\mathcal{G}$ chosen first to be included. Note that the search order and the greedy selection rule are chosen to encourage early detection of better feasible solutions.

If $V(H) \setminus S = \emptyset$ then there is no larger k-club that contains S. Now suppose $V(H) \setminus S \neq \emptyset$. For each integer $1 \leq l \leq |V(H) \setminus S|$, starting from $l = 1$, we initiate a bounded enumeration search with the goal of adding l vertices to the current feasible solution. If at any point in this search we are successful in adding l vertices, we update S, recompute the candidate set as described before, and repeat. If we are unsuccessful in this search for l vertices to add, either because it is not possible, or because we have reached a termination criterion specified in terms of number of tree nodes pruned, we increment l and restart the search with the same S and $V(H)$ as long as $l \leq |V(H) \setminus S|$. The overall procedure stops when either l exceeds $|V(H) \setminus S|$ or if a user-specified time limit is reached.

We maintain the following conditions for the graph $\mathcal{G}$ associated with any active tree node: (C1) F is a k-clique in $\mathcal{G}$; (C2) $d_{\mathcal{G}}(v,u) \leq k, \forall u \in F, v \in U$. Note that at each such node, $S \subseteq F$, $F \setminus S \subseteq V(H) \setminus S$, $U \subseteq V(H) \setminus S$ and vertices in $(V(H) \setminus S) \setminus (F \cup U)$ have been deleted leading to this node of the search tree. While processing a tree node created to include a vertex $i \in U$, if $|F \setminus S| = l$ then the present node is pruned since we are only interested in subsets of $V(H) \setminus S$ of size

l. Otherwise, vertex i is added to F and removed from U. Note that $F \cup \{i\}$ will be a k-clique in $\mathcal{G}$. However, (C2) may be violated by some vertices too far from i in the new U. Thus every vertex $v \in U \setminus \{i\}$ violating (C2) is recursively deleted until no such vertex exists. Similarly, while processing a tree node created to delete a vertex $i \in U$, if $|(V(H) \setminus S) \setminus (F \cup U)| = |V(H) \setminus S| - l$ then the present tree node is pruned. Otherwise, after removing vertex i, every vertex $v \in U \setminus \{i\}$ violating (C2) is recursively deleted until no such vertex exists.

In any of these two cases, if removing these vertices results in (C1) being violated, the present node is pruned by infeasibility. If (C1) and (C2) are still satisfied and $U = \emptyset$ or if $d_{\mathcal{G}}(u, v) \leq k, \forall u, v \in U$ then $\mathcal{G}$ is a k-club. In this case, the lower-bounding procedure is terminated and restarted with the initial solution $S = F \cup U$, the associated candidate set $V(H) \setminus S$ and $l = 1$. Otherwise, two new child nodes are created by using the branching strategy and added to the list of active search tree nodes. After processing a tree node, this node will be removed from the set of active nodes. Whenever the set of active nodes is empty, the bounded enumeration for the present value of l terminates. After terminating the search for the subsets of $V(H) \setminus S$ with size l, if $l = |V(H) \setminus S|$ then the lower-bounding algorithm terminates otherwise, a new bounded enumeration starts for the subsets of size $l+1$. The overall lower-bounding procedure is also limited by a user-specified time limit.

4.2 Branch-and-bound framework to find the k-club number of a graph

In order to study the effectiveness of the bounding techniques proposed in Section 4.1, we incorporate them in a BB algorithm. Two lower-bounding and two upper-bounding schemes (in total four combinations) are tested. Given a graph $G = (V, E)$ and an integer $k \geq 2$, the first lower-bounding technique is to select the larger of the two solutions found by drop and constellation heuristics (denoted by DC). The second technique is the bounded enumeration based lower-bounding

technique (denoted by BE) proposed in Section 4.1.2 which will return a solution at least as good as the first. The lower-bounding scheme is only used once at the beginning of the BB algorithm in order to initialize the incumbent solution.

The upper-bounding scheme is used at each node of the BB tree to find an upper-bound on the k-club number of the graph $\mathcal{G}$ associated with that node. The first technique is to use $\widetilde{\omega}_k(\mathcal{G})$ as the upper-bound on $\overline{\omega}_k(\mathcal{G})$ (denoted by KC). To find $\widetilde{\omega}_k(\mathcal{G})$, the maximum clique problem is solved on $\mathcal{G}^k$ by using Östergård's algorithm [63]. The second technique is the distance k-coloring based upper-bounding technique (denoted by CO) introduced in Section 4.1.1. To properly color $\mathcal{G}^k$, two heuristics are employed. First, a simple greedy heuristic that repeatedly colors the largest degree uncolored node in the power graph with the color not yet assigned to any of its neighbors. Second, we use the well known DSATUR heuristic [65] for graph coloring. In general, the greedy heuristic is faster than DSATUR, but the quality of the solution found by DSATUR is better. Since finding tighter upper-bounds in top levels of the search tree is much more critical, we use DSATUR while processing higher tree levels and use the greedy heuristic for finding upper-bounds in lower levels. A threshold parameter was used to determine the tree level after which the upper-bounding heuristic should be switched from DSATUR to greedy algorithm.

The structure of each BB tree node is similar to the one used for the proposed BE algorithm with the difference that the root node of the BB tree is initialized by setting $F = \emptyset$ and $U = V$. A vertex dichotomy branching rule is also employed in this algorithm. Based on the preliminary computational results, selecting a vertex in U with minimum number of vertices at distance at most k in $\mathcal{G}$ for branching along with the best bound search (BBS) strategy performed better than other branching and search strategies. The reason for this observation is that in this BB algorithm, unlike the proposed BE technique which focuses on feasibility, the emphasis is on finding a maximum k-club in G and proving optimality. In case of a tie in choosing

the branching vertex, the vertex with minimum degree in $\mathcal{G}$ will be selected to branch upon. If a tie still exists, the branching vertex will be selected randomly from tied vertices. Properties (C1) and (C2), mentioned in Section 4.1.2, are also maintained during the course of this BB algorithm. Note that at each node of the BB search tree, vertices in $V \setminus (F \cup U)$ have been deleted along the path connecting the root node of the search tree to the current node.

While processing a BB tree node, if the corresponding upper-bound is less than or equal to the size of the incumbent k-club, the node is then pruned by bound. Otherwise, if the BB node is created to delete a vertex $i \in U$, this vertex is removed from U and every vertex $v \in U \setminus \{i\}$ violating (C2) is recursively deleted until no such vertex exists. Similarly, if the BB node is created to include a vertex $i \in U$, this vertex is added to F and removed from U. Note that the new F will satisfy (C1) but (C2) can be violated by some vertices in $U \setminus \{i\}$ which are too far from i. Thus every vertex $v \in U \setminus \{i\}$ violating (C2) is recursively deleted until no such vertex exists.

If deleting these vertices in any of these two cases results in (C1) being violated, the current BB tree node is pruned by infeasibility. Note that it would be incorrect to prune a BB tree node by infeasibility if $diam(G[F]) > k$. We employ the necessary condition that F must be a k-clique in $\mathcal{G}$ for this purpose. If (C1) and (C2) are still satisfied and $U = \emptyset$ or if $d_{\mathcal{G}}(u, v) \leq k, \forall u, v \in U$ then $\mathcal{G}$ is a k-club. So the current BB tree node is pruned by feasibility and the incumbent solution is updated if necessary. Otherwise, the upper-bounding technique is used to find an upper-bound on the k-club number of the new $\mathcal{G}$. Again, if the estimated upper-bound is less than or equal to the size of the incumbent solution, the node is fathomed by bound otherwise by using the branching strategy, two new child nodes are created and added to the list of active BB tree nodes. The processed BB node then will be removed from the set of active BB nodes. Whenever the set of active BB nodes is empty, the BB algorithm terminates and the size of the incumbent solution will be equal to the k-club number

of G. In addition to this termination criterion, a maximum time allowed is also used to stop the BB algorithm. In case of termination by time limit, the gap between the best upper-bound (the largest upper-bound among all active BB nodes) and the incumbent solution is reported.

Note that for any active node, to compute the upper-bound by distance coloring $\mathcal{G}$, we only need to distance color $G[U]$ as F is a k-clique and every vertex in U is at distance no more than k from every vertex in F. That is, we need $|F|$ colors to color nodes in F and none of these can be used to color nodes in U. The upper-bound for the active node under consideration is equal to the number of colors used to color $G[U]$ plus the cardinality of set F. Another key operation which has a clear effect on the algorithm running time is updating the distance k-neighborhood of each vertex of the graph obtained by deleting vertices. After deleting a vertex i, we only need to update the k-neighborhood of vertices in $N^k_{\mathcal{G}}(i)$. This observation also helps reduce the time taken to update pairwise distances in $\mathcal{G} - i$, after vertex deletion.

4.3 Implementation details and computational test results

All algorithms were implemented in C++ and all numerical experiments were conducted on a Dell® workstation with Intel® Xeon® $W3550$ @ 3.07 GHz processor and 3.00 GB RAM. The test-bed of instances consists of benchmark graphs generated using the algorithm introduced in [43]. The edge density of the graphs produced by this algorithm is controlled by two parameters a and b. The expected edge density d is $(a + b)/2$ and vertex degree variance (VDV) increases with $b - a$. We considered $k = 2$ and 3 and edge densities $d = 0.0125, 0.025, 0.05, 0.1, 0.15, 0.2$ and 0.25 were studied. The experiments were performed on graphs of order $n = 50, 100, 150$ and 200. For each order and density, 10 samples with minimum VDV ($a = b = d$) and 10 samples with maximum VDV ($a = 0, b = 2d$) were generated.

It should be noted that we do not explicitly require the test instances to be con-

nected. Especially the sparse instances could consist of multiple connected components in which case the maximum k-club problem can be decomposed by component. This would be particularly useful to recognize in integer programming approaches, and is implicitly used in our BB algorithm. Also important to note is the fact that the maximum k-club may or may not be in the largest order or the most dense connected component. Table 4.1 shows the average size of the largest connected component in generated test instances. For a given n and for low edge densities, the average size of the largest connected component in instances with minimum VDV is larger than the one for instances with maximum VDV (except for $n = 50$ and density 0.025) and this difference disappears as the edge density increases.

Table 4.1: Average size of the largest connected component in generated test instances

n	VDV	Edge Density						
		0.0125	0.025	0.05	0.1	0.15	0.2	0.25
50	Min	5.6	16.6	43.6	49.5	50	50	50
	Max	4.4	21.6	41.8	48.8	49.1	50	50
100	Min	37.2	91.1	99.7	100	100	100	100
	Max	29.4	87.6	99.4	100	100	100	100
150	Min	112.5	147.6	149.9	150	150	150	150
	Max	109	144	149.6	150	150	150	150
200	Min	178.4	198	200	200	200	200	200
	Max	172.7	197.3	199.9	200	200	200	200

We also observed that some of the higher order, higher density test instances were trivial since their diameter was less than or equal to the value of k under consideration. While this is expected in general, it was interesting to note that in such cases, for any given k, the minimum VDV scheme appears to have a propensity for producing trivial instances. This is attributable to the fact that when VDV is maximum, there

is potential for both large k-clubs (higher degree vertices) as well as isolated vertices. Hence, the entire graph may not have low diameter. We have summarized the number of such trivial instances in Table 4.2.

Table 4.2: Number of trivial test instances in each sample of 10 instances

			d=0.1	d=0.15	d=0.2	d=0.25
k=2	n=150	Min VDV	–	–	–	7
		Max VDV	–	–	–	–
	n=200	Min VDV	–	–	–	10
		Max VDV	–	–	–	1
k=3	n=50	Min VDV	–	2	10	10
		Max VDV	–	1	1	10
	n=100	Min VDV	–	10	10	10
		Max VDV	–	7	10	10
	n=150	Min VDV	9	10	10	10
		Max VDV	1	10	10	10
	n=200	Min VDV	10	10	10	10
		Max VDV	7	10	10	10

4.3.1 Experiments with the lower-bounding techniques

We first consider the numerical results comparing the two lower-bounding techniques DC and BE discussed in Section 4.2. The maximum time allowed for BE is 600 seconds and the number of nodes pruned during BE is limited to 200 for each subset size. We summarize the key observations here but the detailed numerical results showing the performance of these two lower-bounding techniques on our test-bed of instances are provided in Tables B.1–B.4 in Appendix B. For a particular k, VDV and n, the average solution size increases as expected with edge density, and more inter-

estingly, the average running time reaches a peak and then drops. This peak occurs at the same or consecutive densities for both lower-bounding techniques indicating that finding good feasible solutions is challenging in these instances. Furthermore for a given k, we find the densities that are challenging to be the same across minimum and maximum VDV instances, and to be decreasing as n increases. This information is summarized in Table 4.3.

Table 4.3: Challenging densities (among the ones considered) where both BE and DC took the maximum time among all densities

	$n = 50$	$n = 100$	$n = 150$	$n = 200$
$k = 2$	0.25	0.2	0.15, 0.2	0.15
$k = 3$	0.1	0.1	0.05	0.05

The main result of these experiments is that the BE approach is beneficial compared to DC when used on such challenging densities (especially on 150 and 200 vertex instances) as opposed to the non-challenging ones where the DC approach is preferable. Recall that BE approach will take more time as it includes DC in it, but has the potential for returning larger k-clubs. Table 4.4 summarizes the results of these experiments.

4.3.2 Experiments with the branch-and-bound framework

The remainder of our computational experiments are designed to study the performance of four different BB algorithms obtained by using different strategies for lower-bounding and upper-bounding. The four BB algorithms are denoted by DC/CO, DC/KC, BE/CO and BE/KC (see Section 4.2) in the numerical results. Tables 4.5 and 4.6 report the computational results obtained by solving the maximum k-club problem on 200-vertex graphs for $k = 2$ and $k = 3$, respectively. Average best objective value, running time and optimality gap across the 10 samples in each category

Table 4.4: Minimum and maximum percentage increase in average best objective value found by BE over DC, and increase in average running time in seconds for the challenging densities (over 10 samples)

	Metric	$n = 50$	$n = 100$	$n = 150$	$n = 200$
$k = 2$	Best Obj	(1.48, 2.53)	(0.71, 3.92)	(0.25, 5.31)	(2.06, 17.44)
	Time	(0.75, 0.75)	(8.20,10.20)	(15.51, 36.93)	(98.81, 218.45)
$k = 3$	Best Obj	(0.69, 1.59)	(0.00, 0.11)	(4.22, 7.26)	(3.03, 5.35)
	Time	(0.68, 0.70)	(7.10, 7.64)	(25.24, 30.00)	(143.55, 159.93)

of test instances are reported in these tables.

Although we present the results for 200-vertex instances here, the results are similar across different orders and the subsequent observations are made considering all the instances in our test-bed. The complete set of computational results including instances of smaller orders ($n = 50, 100$ and 150) are provided in Tables B.5–B.12 in Appendix B.

The running time limit for all algorithms is 3600 seconds. This was enforced by checking the elapsed time after each BB node was processed, which in some instances exceeded the time limit due to the time it took to process the last BB node before termination. If an instance cannot be solved to optimality within the time limit, relative optimality gap is reported. The percentage gap is calculated as *100×(upper-bound - best solution size)/upper-bound*. The settings used for BE are as stated before. The tree level threshold parameter for switching the upper-bounding heuristic from DSATUR to simple greedy is set at $0.10 \times n$.

For a given k, VDV and n, as edge density increases the average incumbent solution size found by all algorithms increases, while generally the average running time and optimality gap increase up to a peak and then decrease. The peak average running time can be used to determine the challenging densities for these BB algorithms.

Table 4.5: Average size of the best 2-club found, average running time (in seconds), and percentage optimality gap for each BB algorithm on 200-vertex instances

VDV	Metric	Algorithm	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
Min	Best Obj	DC/CO	8.20	12.70	20.70	33.60	58.50	195.50	200.00
		DC/KC	8.20	12.70	20.70	33.60	58.50	195.50	200.00
		BE/CO	8.20	12.70	20.70	33.60	68.70	195.50	200.00
		BE/KC	8.20	12.70	20.70	33.60	68.70	195.50	200.00
	Time	DC/CO	2.54	71.62	298.98	3604.30	3613.16	880.00	0.00
		DC/KC	0.95	2.12	16.71	4554.49	5276.76	331.06	0.00
		BE/CO	2.65	71.93	300.04	3605.00	3618.83	913.00	0.00
		BE/KC	1.01	2.53	18.18	4548.06	4542.85	361.98	0.00
	Gap	DC/CO	0.00	0.00	0.00	37.78	56.14	0.00	0.00
		DC/KC	0.00	0.00	0.00	83.20	70.75	0.00	0.00
		BE/CO	0.00	0.00	0.00	37.66	48.78	0.00	0.00
		BE/KC	0.00	0.00	0.00	83.20	65.65	0.00	0.00
Max	Best Obj	DC/CO	9.20	14.30	23.40	39.10	126.20	178.20	196.40
		DC/KC	9.20	14.30	23.40	39.10	126.20	177.20	196.40
		BE/CO	9.20	14.30	23.40	39.10	128.80	178.20	196.40
		BE/KC	9.20	14.30	23.40	39.10	128.80	177.40	196.40
	Time	DC/CO	2.61	48.14	299.68	3608.07	3403.16	1664.25	777.07
		DC/KC	0.96	2.24	42.41	4884.64	4377.61	5603.51	116.33
		BE/CO	2.68	48.07	300.83	3604.49	3411.23	1747.34	813.20
		BE/KC	1.03	2.67	44.07	4878.12	4395.17	5606.31	151.69
	Gap	DC/CO	0.00	0.00	0.00	29.77	8.00	0.06	0.00
		DC/KC	0.00	0.00	0.00	80.45	36.90	10.35	0.00
		BE/CO	0.00	0.00	0.00	29.34	6.05	0.06	0.00
		BE/KC	0.00	0.00	0.00	80.45	35.60	10.25	0.00

Table 4.6: Average size of the best 3-club found, average running time (in seconds), and percentage optimality gap for each BB algorithm on 200-vertex instances

VDV	Metric	Algorithm	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
Min	Best Obj	DC/CO	13.60	25.40	89.70	200.00	200.00	200.00	200.00
		DC/KC	13.60	20.70	89.70	200.00	200.00	200.00	200.00
		BE/CO	13.60	25.40	94.50	200.00	200.00	200.00	200.00
		BE/KC	13.60	23.40	94.50	200.00	200.00	200.00	200.00
	Time	DC/CO	87.90	442.32	3616.94	0.00	0.00	0.00	0.00
		DC/KC	17.24	3684.29	4108.80	0.00	0.00	0.00	0.00
		BE/CO	88.28	431.66	3619.46	0.00	0.00	0.00	0.00
		BE/KC	17.98	3702.84	4160.96	0.00	0.00	0.00	0.00
	Gap	DC/CO	0.00	0.00	31.64	0.00	0.00	0.00	0.00
		DC/KC	0.00	89.30	55.15	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	28.13	0.00	0.00	0.00	0.00
		BE/KC	0.00	87.92	52.75	0.00	0.00	0.00	0.00
Max	Best Obj	DC/CO	14.80	31.90	119.10	199.70	200.00	200.00	200.00
		DC/KC	14.80	24.90	118.70	199.70	200.00	200.00	200.00
		BE/CO	14.80	31.90	122.50	199.70	200.00	200.00	200.00
		BE/KC	14.80	29.40	122.30	199.70	200.00	200.00	200.00
	Time	DC/CO	91.39	477.40	3275.85	262.39	0.00	0.00	0.00
		DC/KC	63.03	3760.07	4523.58	32.50	0.00	0.00	0.00
		BE/CO	91.75	464.38	3304.12	268.81	0.00	0.00	0.00
		BE/KC	62.81	3757.28	4526.92	38.81	0.00	0.00	0.00
	Gap	DC/CO	0.00	0.00	12.29	0.00	0.00	0.00	0.00
		DC/KC	0.00	87.39	40.65	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	9.64	0.00	0.00	0.00	0.00
		BE/KC	0.00	85.11	38.85	0.00	0.00	0.00	0.00

It can be observed that for a given k, the challenging densities are the same for both minimum and maximum VDV instances, and decrease as n increases. Table 4.7 identifies these challenging densities for all BB algorithms.

Table 4.7: Challenging densities (among the ones considered) where all four BB algorithms took the maximum time across all densities

	$n = 50$	$n = 100$	$n = 150$	$n = 200$
$k = 2$	0.2, 0.25	0.15, 0.2	0.15, 0.2	0.1, 0.15, 0.2
$k = 3$	0.1	0.05	0.05	0.05

For a given k, n, VDV and for densities that are not challenging, the quality of the solution found by all algorithms are nearly identical. Considering average running time for these non-challenging densities, DC/KC almost always performs better than the other algorithms. This observation is attributable to the fact that on these densities, BE does not significantly improve the quality of the initial solution. Furthermore, it appears that solving the maximum k-clique problem which returns potentially tighter upper-bounds is not as hard on these densities. Likewise for these non-challenging densities, all the algorithms return a zero optimality gap on nearly all the instances.

However, for challenging densities, BE/CO outperforms all the other algorithms in terms of solution quality. This observation can be explained by the fact that on these densities, BE often returns a larger initial solution in a reasonable amount of time. Furthermore, solving the maximum k-clique problem to optimality on these densities is relatively harder than finding a feasible distance k-coloring, requiring longer times to compute the upper-bound at each node. This results in fewer BB nodes being enumerated in a specified time limit, affecting the incumbent solution quality adversely. Due to similar reasons, both DC/CO and BE/CO perform better than the other two algorithms in terms of average running time metric. In terms

of optimality gap, DC/CO and BE/CO outperform the other two algorithms while BE/CO is slightly better than DC/CO.

CHAPTER 5

THE 2-CLUB POLYTOPE

In this chapter, we study the 2-club polytope of a graph and identify a new family of facet inducing inequalities for this polytope. This family of facets strictly contains all known nontrivial facets of the 2-club polytope and introduces new facets of this polytope to the literature. The separation complexity of the newly discovered facet inducing inequalities is studied in this chapter and it is shown that these facets along with the nonnegativity constraints completely describe the 2-club polytope of trees.

It is important to note that nonhereditary property of k-clubs also poses interesting challenges to polyhedral approaches. For instance, given a graph $G = (V, E)$ and a nonempty set $S \subset V$, some facet inducing inequalities for the k-club polytope of $G[S]$ will not necessarily be valid for the k-club polytope of G. This observation is interesting because of the fact that for the models with hereditary property like k-cliques, facet inducing inequalities for the k-clique polytope of $G[S]$ will remain valid for the k-clique polytope of G. While dealing with k-clubs, lifting some facets of the k-club polytope of $G[S]$ is necessary in order to generate valid inequalities for the k-club polytope of G. For example, consider the graph G shown in Figure 5.1 and let $S = \{1, 2, 3, 4, 5\}$. The inequality $x_1 + x_2 + x_3 + x_4 + x_5 \leq 1$ induces a facet of the 2-club polytope of $G[S]$ while it is not valid for the 2-club polytope of G as it cuts off the incidence vector of set $\{1, 2, 3, 4, 5, 6\}$ which is a 2-club in G. Lifting x_6 will yield inequality $x_1 + x_2 + x_3 + x_4 + x_5 - 4x_6 \leq 1$ which induces a facet of the 2-club polytope of G.

Before presenting the main results obtained for the 2-club polytope of a graph, let

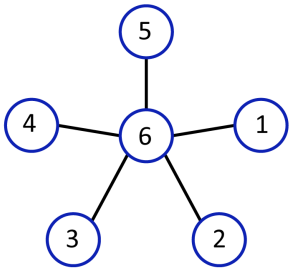

Figure 5.1: A graph in which $\sum_{i \in S} x_i \leq 1$ is not valid for the 2-club polytope while it induces a facet of the 2-club polytope of $G[S]$, where $S = \{1, 2, 3, 4, 5\}$

us introduce some notations and definitions which are used in this chapter. Given a set $S \subseteq V$, let $\Gamma(S)$ denote the set that contains all maximal 2-independent sets in $G[V \setminus N_G^2[S]]$. Given a graph $G = (V, E)$, let $\Omega(G)$ denote the set of all independent 2-dominating sets in G. Note that any maximal 2-independent set in G is also an independent 2-dominating set so $\Omega(G)$ contains all maximal 2-independent sets in G. Finally, for $i \in V$, we denote by e_i the vector in $\mathbb{R}^n$ with all components equal to 0, except for the i-th component which is 1.

5.1 Independent 2-dominating set inequalities

Theorem 7 characterizes the family of independent 2-dominating set (I2DS) facets for the 2-club polytope of a graph. In order to present Theorem 7, we first provide Lemma 5.1 and Theorem 6 which are used in proof of Theorem 7.

Lemma 5.1 *Let $G = (A, B, E)$ be a bipartite graph such that:*

1. $|A| \geq 2$, $|B| \geq 1$.
2. *For all $v \in B$, $N(v) \cap A \neq \emptyset$.*
3. *For all pairs $s, t \in A$, $N(s) \cap N(t) \neq \emptyset$.*

Then, G is connected.

Proof. By definition, there exists a path between $u, v \in A$. For a pair $u, v \in B$, suppose there does not exist a path. Then, they belong to two different components and hence, there exists a $S \subseteq A \cup B$ such that $u \in S$, $v \notin S$ and $\{(i,j) \in E : i \in S, j \notin S\} = \emptyset$. By Condition 2, there exists $a_1 \in N(u)$ and $a_2 \in N(v)$ such that $a_1 \in S$ and $a_2 \notin S$. But $N(a_1) \cap N(a_2) = \emptyset$, contracting Condition 3. Now consider, $u \in A$ and $v \in B$ such that there does not exist a u, v-path in G. Then, there exists $S \subseteq A \cup B$ such that $u \in S$, $v \notin S$ and $\{(i,j) \in E : i \in S, j \notin S\} = \emptyset$. By Condition 2, there exists $w \in N(v)$ such that $w \notin S$. Then $N(u) \cap N(w) = \emptyset$, contracting Condition 3. Hence, G is connected. ■

Theorem 6 *Given a graph $G = (V, E)$ and an independent set $C \subseteq V$, the following inequality is valid for the 2-club polytope of G.*

$$\sum_{i \in C} x_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ x_i \leq 1 \tag{5.1}$$

Proof. Consider a 2-club D in G. The inequality is trivially valid for a 2-club D such that $|D \cap C| \leq 1$. So we assume that $|D \cap C| \geq 2$. Under this assumption, we must have some $u \in D \setminus C$ such that $|N(u) \cap D \cap C| \geq 2$. Let,

$$J = \{i \in D \setminus C : |N(i) \cap C| \geq 2\} \neq \emptyset.$$

Then, the inequality reduces to showing that

$$|D \cap C| - \sum_{i \in J} (|N(i) \cap C| - 1) \leq 1$$

or equivalently,

$$|D \cap C| + |J| - 1 \leq \sum_{i \in J} |N(i) \cap C|.$$

Let,

$$H = \{i \in J : N(i) \cap D \cap C \neq \emptyset\} \neq \emptyset.$$

Consider the subgraph induced by vertices in $D \cap C$ and H. Ignoring edges with both endpoints in H, we obtain a bipartite graph with partitions $D \cap C$ and H. We have

$|D \cap C| \geq 2$, $|H| \geq 1$, for any $v \in H$, $N(v) \cap D \cap C \neq \emptyset$. Since D is a 2-club and C is independent, for any $u, v \in D \cap C$, $N(u) \cap N(v) \cap H \neq \emptyset$. By Lemma 5.1, this bipartite graph is connected. Hence, the number of edges in this graph is at least $|D \cap C| + |H| - 1$. That is,

$$|D \cap C| + |H| - 1 \leq \sum_{i \in H} |N(i) \cap D \cap C|.$$

So,

$$|D \cap C| + |H| - 1 \leq \sum_{i \in H} |N(i) \cap C|.$$

Now, for each $v \in J \setminus H$, $|N(i) \cap C| \geq 2$. Thus,

$$|D \cap C| + |H| + |J \setminus H| - 1 \leq \sum_{i \in H} |N(i) \cap C| + \sum_{i \in J \setminus H} |N(i) \cap C|$$

or equivalently,

$$|D \cap C| + |J| - 1 \leq \sum_{i \in J} |N(i) \cap C|.$$

■

Theorem 7 *[I2DS facet] Given a graph $G = (V, E)$ and an independent 2-dominating set $C \subseteq V$, the following inequality defines a facet for the 2-club polytope of G.*

$$\sum_{i \in C} x_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ x_i \leq 1 \tag{5.2}$$

Proof. Inequality 5.2 is valid for the 2-club polytope of G by Theorem 6. Now, partition V into C, C^1 and C^2 where $C^1 = \cup_{v \in C} N(v) \setminus C$ and $C^2 = V \setminus (C \cup C^1)$. For any $i \in C^2$, there exists $j \in C$ and $t \in C^1$ such that $t \in N(i) \cap N(j)$ as C is 2-dominating. In the following, we construct 2-clubs $S^1, \ldots, S^n$, whose incidence vectors satisfy Inequality 5.2 as equality and are linearly independent. For all $i \in C$, $S^i = \{i\}$, for all $i \in C^1$, $S^i = \{i\} \cup (N(i) \cap C)$, and for all $i \in C^2$, $S^i = \{i\} \cup S^j$ where $(i, j) \in E$ for some $j \in C^1$. This completes the proof of Theorem 7.

■

The family of I2DS facets strictly includes all previously known nontrivial facets for the 2-club polytope described in Theorems 1c, 1d, 2 and 3 of Section 2.2. Note that nonnegativity facets presented in Theorem 1b are trivial facets for the 2-club polytope. It can be easily verified that sets $\{i\}$ in Theorem 1c, I in Theorem 1d, $I \cup \{a, b\}$ in Theorem 2 and $I \cup \{a, b, c\}$ in Theorem 3 are all independent 2-dominating sets and their corresponding inequalities are special cases of inequality 5.2. It should be noted that the family of I2DS facets also introduces new facets of the 2-club polytope which were previously unknown. For example, in the graph shown in Figure 5.2, inequality $x_1 + x_3 + x_4 + x_7 + x_8 - 2x_2 - x_5 - x_6 \leq 1$ is an I2DS facet for the 2-club polytope of this graph which was previously unknown.

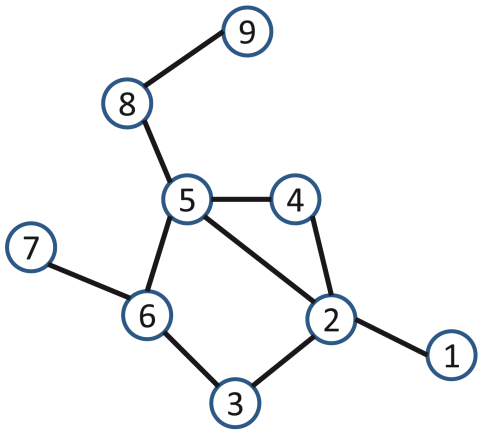

Figure 5.2: A graph in which $x_1 + x_3 + x_4 + x_7 + x_8 - 2x_2 - x_5 - x_6 \leq 1$ is an I2DS facet for the 2-club polytope which was previously unknown

5.2 Independent 2-dominating set inequalities separation complexity

The computational complexity of I2DS facet separation problem plays an important role in developing branch-and-cut algorithms for solving the maximum 2-club problem.

The 2-club polytope of a graph is contained in the polyhedron obtained by the linear programming relaxation of the maximum 2-club problem formulation provided

in Section 2.2. Let,

$$\mathcal{Q} = \{x \in [0,1]^n : x_i + x_j - \sum_{k \in N(i) \cap N(j)} x_k \leq 1 \ \ \forall (i,j) \notin E, i < j, \text{and } i,j \in V\}.$$

In this section, we are concerned with the complexity of identifying I2DS inequalities violated by a fractional point in $\mathcal{Q}$ (if any exists). An instance of the I2DS inequalities separation problem is given by a simple undirected graph $G = (V, E)$ and a feasible fractional point $x^* \in \mathcal{Q}$, and we ask if there exist a set $C \in \Omega(G)$ such that $\sum_{i \in C} x_i^* - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ x_i^* > 1$. The following theorem establishes that the I2DS inequalities separation problem is NP-complete.

Theorem 8 *Given a graph $G = (V, E)$ and a fractional point $x^* \in \mathcal{Q}$, deciding whether there exists a violated I2DS inequality is NP-complete.*

Proof. A polynomial time reduction from 3-SAT [23] is used to prove this theorem. We assume that in the 3-SAT instances, there is no clause that contains a literal and its negation and there are at least 3 clauses in the 3-SAT formula. Note that the 3-SAT problem is still NP-complete under these restrictions. Denote the 3-SAT instance with n boolean variables, m clauses with $3m$ literals as $B = \bigwedge_{i=1}^{m} (p_{i1} \vee p_{i2} \vee p_{i3})$. For any such 3-SAT instance $< B >$, the following steps construct in polynomial time, an instance $< G, x^* >$ for I2DS inequalities separation problem such that the 3-SAT instance is satisfiable if and only if there exists a set $C \in \Omega(G)$ such that,

$$\sum_{k \in C} x_k^* - \sum_{k \in V \setminus C} (|N(k) \cap C| - 1)^+ x_k^* > 1.$$

1. For each clause $i \in \{1, \ldots, m\}$ in B, four vertices are created in G, three of which correspond to the three literals in clause i. We call these vertices, the *literal vertices* of clause i. The fourth vertex is called the *base vertex* of clause i. Additionally, for each literal pair p_{ij} and p_{uv} such that $p_{ij} = \overline{p}_{uv}$ (by assumption $i \neq u$), a vertex is added to G. These vertices are called *median vertices.* The

union of the literal, base and median vertex sets across all clauses in B will form the vertex set V of graph G. So we have $|V| = 4m + r$ where r is the total number of literal pairs that are negations of each other.

2. For each clause $i \in \{1, \ldots, m\}$ in B, each literal vertex in clause i is connected to this clause's base vertex. If literal p_{ij} is the negation of literal p_{uv} then their literal vertices are connected to the corresponding median vertex in G. Finally, the set of all base and median vertices form a clique in G. The union of these edge sets will form the edge set E of graph G.

3. Now arbitrarily assign an index k from 1 to $|V|$ to the elements of V. For each clause $i \in \{1, \ldots, m\}$, let $L(i)$ denote the set containing the indices of the three literal vertices in i and $b(i)$ refer to this clause's base vertex index. Furthermore, the index set of all median vertices in G is denoted by M. Let $x^*_k = \frac{2}{2m-1}$ for all $k \in V$. Since $m \geq 3$, it is easy to verify that $x^* \in \mathcal{Q}$.

If the 3-SAT instance is satisfiable then consider a particular satisfying assignment and let set C contain the indices of all literal vertices for which the corresponding literal is equal to one. Set C is independent in G and since there exist at least one literal with value one in each clause, the set C is also a 2-dominating set. So set C is an independent 2-dominating set. We have,

$$\sum_{k \in C} x^*_k - \sum_{k \in V \setminus C} (|N(k) \cap C| - 1)^+ x^*_k = \sum_{i=1}^{m} (\sum_{k \in L(i) \cap C} x^*_k - \sum_{k \in L(i) \setminus C} (|N(k) \cap C| - 1)^+ x^*_k$$
$$- (|N(b(i)) \cap C| - 1)^+ x^*_{b(i)}) - \sum_{k \in M} (|N(k) \cap C| - 1)^+ x^*_k.$$

Considering each clause $i \in \{1, \ldots, m\}$, we have $-(|N(b(i)) \cap C| - 1)^+ = -(t_i - 1)$ where t_i is the number of literals with value one in clause i and for each $k \in L(i) \setminus C$, $-(|N(k) \cap C| - 1)^+ = 0$. For each $k \in M$, $-(|N(k) \cap C| - 1)^+ = 0$ because exactly one of the literal vertices in the corresponding negating pair is inside C. So,

$$\sum_{k \in C} x^*_k - \sum_{k \in V \setminus C} (|N(k) \cap C| - 1)^+ x^*_k = \sum_{i=1}^{m} (t_i * (\frac{2}{2m-1}) - (t_i - 1) * (\frac{2}{2m-1})) =$$

$$\sum_{i=1}^{m} \frac{2}{2m-1} = \frac{2m}{2m-1} > 1.$$

Now suppose there exists a set $C \in \Omega(G)$ such that $\sum_{k\in C} x_k^* - \sum_{k\in V\setminus C}(|N(k) \cap C| - 1)^+ x_k^* > 1$. Given a clause $i \in \{1, \ldots, m\}$, let t_i denote the number of literal vertices corresponding to clause i which belong to C.

Set C does not contain any base or median vertex. To show this, suppose C contains a base vertex which corresponds to a clause $u \in \{1, \ldots, m\}$. Since all the base and median vertices form a clique in G, C cannot contain any other base or median vertex. So we have $-(|N(k)\cap C|-1)^+ = 0$ for all $k \in L(u)$. Furthermore, for a clause $i \in \{1, \ldots, m\} \setminus \{u\}$, we have $-(|N(b(i)) \cap C| - 1)^+ = -((t_i + 1) - 1)^+ = -t_i$ and for each $k \in L(i) \setminus C$, $-(|N(k) \cap C| - 1)^+ = 0$. So,

$$\sum_{k\in C} x_k^* - \sum_{k\in V\setminus C} (|N(k)\cap C|-1)^+ x_k^* = \frac{2}{2m-1} + \sum_{i\in\{1,\ldots,m\}\setminus\{u\}} (t_i * (\frac{2}{2m-1}) - t_i * (\frac{2}{2m-1})) -$$

$$\sum_{k\in M} (|N(k) \cap C| - 1)^+ (\frac{2}{2m-1}) \leq \frac{2}{2m-1} < 1$$

which is a contradiction.

Now suppose set C contains a median vertex $l \in M$ (which is adjacent to two negating literals p_{st} and p_{uv}). Again, since all the base and median vertices in G form a clique, C does not include any other base or median vertex. Additionally, for a clause $i \in \{1, \ldots, m\}$, we have $-(|N(b(i)) \cap C| - 1)^+ = -((t_i + 1) - 1)^+ = -t_i$ and for each $k \in L(i) \setminus C$, $-(|N(k) \cap C| - 1)^+ = 0$. So,

$$\sum_{k\in C} x_k^* - \sum_{k\in V\setminus C} (|N(k) \cap C| - 1)^+ x_k^* = \frac{2}{2m-1} - \sum_{k\in M\setminus\{l\}} (|N(k) \cap C| - 1)^+ (\frac{2}{2m-1}) +$$

$$\sum_{i\in\{1,\ldots,m\}} (t_i * (\frac{2}{2m-1}) - t_i * (\frac{2}{2m-1})) \leq \frac{2}{2m-1} < 1$$

which is again a contradiction.

Since there is no base or median vertex in C, this implies,

$$\forall k \in (\cup_{i=1}^{m} L(i)) \setminus C, \quad -(|N(k) \cap C| - 1)^+ = 0.$$

Set C should contain at least one literal vertex from each clause $i \in \{1, \ldots, m\}$. To show this, suppose C does not contain any literal vertex of a clause $u \in \{1, \ldots, m\}$. This means $-(|N(b(u)) \cap C| - 1)^+ = 0$. So we have,

$$\sum_{k \in C} x_k^* - \sum_{k \in V \setminus C} (|N(k) \cap C| - 1)^+ x_k^* \leq \sum_{i \in \{1,\ldots,m\} \setminus \{u\}} (t_i * (\frac{2}{2m-1}) - (t_i - 1)^+ * (\frac{2}{2m-1}))$$

$$\leq \frac{2(m-1)}{2m-1} < 1$$

which is again a contradiction.

It can also be shown that set C does not contain two literal vertices for which the corresponding literals are negation of each other. To prove this, suppose there exist literal vertices x and y in C such that correspond to a negating pair. Suppose x and y belong to clauses u and w, respectively where $u \in \{1, \ldots, m\}$, $w \in \{1, \ldots, m\}$ and $u \neq w$ and let z denote the median vertex connected to these literal nodes. We know $-(|N(z) \cap C| - 1)^+ = -1$. So,

$$\sum_{k \in C} x_k^* - \sum_{k \in V \setminus C} (|N(k) \cap C| - 1)^+ x_k^* \leq \sum_{i \in \{1,\ldots,m\}} (t_i * (\frac{2}{2m-1}) - (t_i - 1)^+ * (\frac{2}{2m-1})) -$$

$$\frac{2}{2m-1} - \sum_{k \in M \setminus \{z\}} (|N(k) \cap C| - 1)^+ (\frac{2}{2m-1}) \leq \frac{2m}{2m-1} - \frac{2}{2m-1} < 1$$

which is again a contradiction. So by setting the value of the boolean variables corresponding to literal vertices in C to 1 and the rest of them to zero, we will have a satisfying assignment for B.

This establishes that I2DS inequalities separation problem is NP-hard. It is easy to see that this problem is in class NP. This completes the proof of Theorem 8.

■

5.3 The 2-club polytope of trees

In this section, we derive the complete description of the 2-club polytope of trees using the collection of I2DS and nonnegativity inequalities. Given a graph $G = (V, E)$, let

P_G denote the polyhedron obtained by the set of I2DS and nonnegativity inequalities in graph G described as follows.

$$P_G = \{x \in \mathrm{R}_+^{|V|} : \sum_{k \in C} x_k - \sum_{k \in V \setminus C} (|N(k) \cap C| - 1)^+ x_k \leq 1,\ \forall C \in \Omega(G)\} \tag{5.3}$$

Considering any $j \in V$, we know there exists a maximal 2-independent set in G that contains j, say I^j. According to the description of P_G, we have $\sum_{i \in I^j} x_i \leq 1$ and $x_i \geq 0$ for all $i \in V$, so $x_j \leq 1$. Hence, $\emptyset \neq P_G \subseteq [0,1]^{|V|}$.

Theorems 9 and 10 show some useful results concerning polyhedron P_G, its extreme points and integral members.

Theorem 9 *The incidence vector of any 2-club in G is an extreme point of P_G.*

Proof. This result follows from $Q_2(G) \subseteq P_G \subseteq [0,1]^{|V|}$. We supply an alternate direct proof here. Let $\hat{x}$ be the incidence vector of a 2-club D in G. By Theorem 7, we know $\hat{x} \in P_G$. So it suffices to show that $\hat{x}$ is a basic solution in P_G. We now demonstrate $|V|$ linearly independent inequalities in description of P_G that are active at $\hat{x}$. Consider the nonnegativity inequalities $x_i \geq 0$ for all $i \in V \setminus D$ and consider a maximal 2-independent set inequality based on $I^i \in \Omega(G)$ for each $i \in D$ such that $i \in I^i$. Note that $D \cap I^i = \{i\}$. These $|V|$ inequalities can be easily verified as linearly independent and are active at the incidence vector $\hat{x}$ of D.

■

Remark. The purpose of this direct proof is to highlight the potential for degeneracy at the integral extreme points of P_G that correspond to 2-clubs in G. This argument also extends to $Q_2(G)$ since every one of its extreme points can be determined by maximal 2-independent set inequalities of the type used in the proof and nonnegativity, all of which are known to be facet inducing and hence, part of any description of $Q_2(G)$. Hence, I2DS facets and other facets increase the number of

active facets at an extreme point of $Q_2(G)$ beyond $|V|$, in general. Note that if G^2 is perfect and the 2-club polytope of G is indeed completely described by maximal 2-independent set inequalities and nonnegativity, the 2-club polytope of G and the 2-clique polytope of G coincide [66].

Theorem 10 *If $\hat{x} \in P_G$ is an integral vector, then it is an incidence vector of a 2-club in G.*

Proof. Let $S = \{i \in V : \hat{x}_i = 1\}$. Now suppose S is not a 2-club in G. So there exist $j, k \in S$ such that $d_{G[S]}(j,k) > 2$. If $d_G(j,k) > 2$ then there exists a maximal 2-independent set inequality in P_G violated by $\hat{x}$. So we have $d_G(j,k) \leq 2$. For any set $D \in \Gamma(\{j,k\})$, set $C = \{j,k\} \cup D$ is an independent 2-dominating set in G and $\sum_{i \in C} \hat{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \hat{x}_i = 2 + |D \cap S| - \sum_{i \in S \setminus C} (|N(i) \cap C| - 1)^+$. Since $d_{G[S]}(j,k) > 2$, so for all $i \in S \setminus C$, $|N(i) \cap C| \leq 1$ and we have $\sum_{i \in C} \hat{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \hat{x}_i > 1$ which is a contradiction with $\hat{x} \in P_G$. So $\hat{x}$ is an incidence vector of a 2-club in G. ■

Theorem 11 characterizes complete description of the 2-club polytope of trees using the collection of I2DS and nonnegativity inequalities.

Theorem 11 *If $G = (V, E)$ is a tree, then $P_G = Q_2(G)$.*

Proof. By Theorems 7 and 10, it suffices to show that P_G is integral. Suppose P_G is not integral. Then, it has an extreme point $\bar{x}$ which is not integral. Without loss of generality, suppose I2DS inequalities and nonnegativity constraints in description of P_G are indexed from 1 to $|\Omega(G)|$ and from $|\Omega(G)|+1$ to $|\Omega(G)|+|V|$ respectively. Since $\bar{x}$ is an extreme point, there are $|V|$ constraints with linearly independent coefficient vectors in the description of P_G that are active at this point. Let S denote the set that contains the indices of such constraints. Also suppose $S_1 \subseteq S$ is the set that contains all elements of S for which the corresponding constraint is an I2DS inequality

and let $S_2 = S \setminus S_1$. We know $|S| = |S_1| + |S_2| = |V|$ and $|S_1| \geq 1$ since $\bar{x}$ is not integral. Let A_1 be a $|S_1| \times |V|$ matrix in which each row is the coefficient vector of the corresponding constraint of each element in S_1. Similarly define A_2 which is a $|S_2| \times |V|$ matrix considering coefficient vectors of the elements of S_2 as its rows. Since the rows of A_1 and A_2 form $|V|$ linearly independent vectors, the following system has a unique solution which is $\bar{x}$.

$$A_1 x = \mathbf{1} \text{ and } A_2 x = \mathbf{0} \tag{5.4}$$

The following Claims 4 - 10 are used to complete the proof. The proof of these claims can be found in Appendix C.

Claim 4 *For any $j \in V$, we have*

$$\bar{x}_j \leq \sum_{k \in N_G(j)} \bar{x}_k \tag{5.5}$$

Now define set $V' = \{i \in V : \bar{x}_i > 0\}$ and let $G' = G[V']$. Since $V' \neq \emptyset$, G' has at least one connected component (say $\bar{G}$) that is a tree. Let $\tau(S_1)$ denote the set that contains all independent 2-dominating sets in G for which the index of the corresponding I2DS constraint belongs to S_1.

Case (I), $diam(\bar{G}) = 0$. Then, $\bar{G}$ is an isolated vertex (say a) and each $C \in \tau(S_1)$ contains vertex a. Otherwise, for any set $D \in \Gamma((C \cup \{a\}) \setminus N_G(a))$, set $C' = (C \cup \{a\} \cup D) \setminus N_G(a)$ is an independent 2-dominating set in G and $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i + \bar{x}_a > 1$ which contradicts with $\bar{x} \in P_G$. So e_a will be a solution for system 5.4 which is a contradiction with $\bar{x}$ being the unique solution of this system.

Case (II), $diam(\bar{G}) = 1$. Then, it contains two vertices (say a and b) that are connected with an edge. If $\bar{x}_a \neq \bar{x}_b$ (without loss of generality suppose $\bar{x}_a > \bar{x}_b$) then again each $C \in \tau(S_1)$ contains a. Otherwise, for any set $D \in \Gamma((C \cup \{a\}) \setminus N_G(a))$, set $C' = (C \cup \{a\} \cup D) \setminus N_G(a)$ is an independent 2-dominating set in G for which

$\sum_{i\in C'} \bar{x}_i - \sum_{i\in V\setminus C'}(|N(i)\cap C'|-1)^+\bar{x}_i \geq \sum_{i\in C} \bar{x}_i - \sum_{i\in V\setminus C}(|N(i)\cap C|-1)^+\bar{x}_i + \bar{x}_a - \bar{x}_b > 1$. This again contradicts with $\bar{x} \in P_G$. So e_a will solve system 5.4 and $\bar{x}$ is not the unique solution for this system which is a contradiction.

Now if $\bar{x}_a = \bar{x}_b$ then every $C \in \tau(S_1)$ contains exactly one vertex from set $\{a, b\}$. Because set C can not contain both a and b since it is an independent set and if it does not contain any of them, then again for any set $D \in \Gamma((C\cup\{a\})\setminus(N_G(a)\cup N_G(b)))$, set $C' = (C\cup\{a\}\cup D)\setminus(N_G(a)\cup N_G(b))$ is an independent 2-dominating set in G for which $\sum_{i\in C'} \bar{x}_i - \sum_{i\in V\setminus C'}(|N(i)\cap C'|-1)^+\bar{x}_i \geq \sum_{i\in C} \bar{x}_i - \sum_{i\in V\setminus C}(|N(i)\cap C|-1)^+\bar{x}_i + \bar{x}_a > 1$. This again is a contradiction with $\bar{x} \in P_G$. So the vector $\hat{e} = e_a + e_b$ will solve system 5.4 which contradicts with uniqueness of $\bar{x}$.

Case (III), diam$(\bar{G}) \geq 2$. Let p denote the longest path in $\bar{G}$. By considering an arbitrary direction for p, let a, b and c be the first, second and third vertex on path p while moving along this direction respectively. Since the length of p is at least two, vertices a, b and c exist. Clearly each vertex $i \in (N_G(b)\cap V')\setminus\{c\}$ is a leaf vertex of $\bar{G}$ because otherwise p is not the longest path in this graph. Considering the connected component $\bar{G}$, the following results can be derived.

Claim 5 *Any set $C \in \tau(S_1)$ contains at least one element from set $N_G[b] \cap V'$.*

Claim 6 *Consider a leaf vertex of $\bar{G}$ say vertex a. For any set $C \in \tau(S_1)$, the coefficient of x_a in the I2DS constraint corresponding to set C is either zero or one.*

Claim 7 *Given a set $C \in \tau(S_1)$, for any $j \in (N_G(b) \cap C)$, we have $\bar{x}_j > 0$.*

Claim 8 *There exists at least one set $C \in \tau(S_1)$ for which the coefficient of x_c in the corresponding I2DS constraint is negative.*

Claim 9 *There exists at least one set $C \in \tau(S_1)$ for which the coefficient of x_c in the corresponding I2DS constraint is one.*

Case (IIIA), $diam(\bar{G}) = 2$. So $N_G(c) \cap V' = \{b\}$. Thus c is a leaf vertex of $\bar{G}$ and by Claim 6, for any set $C \in \tau(S_1)$, the coefficient of x_c in the I2DS constraint corresponding to set C is either zero or one. This contradicts with the result of Claim 8.

Case (IIIB), $diam(\bar{G}) \geq 3$. Let vertex d be the forth vertex on path p moving along the aforementioned direction.

Claim 10 *For all $j \in (N_G(c) \cap V') \setminus \{d\}$, we have $\bar{x}_j \leq \bar{x}_c$.*

Since $diam(\bar{G}) \geq 3$ so $|N_G(c) \cap V'| \geq 2$. If $|N_G(c) \cap V'| = 2$ then for all $C \in \tau(S_1)$, the coefficient of x_c in the I2DS constraint corresponding to set C is either zero or one. To show this, suppose there exists a set $C \in \tau(S_1)$ for which this coefficient is negative. If $b \notin C$ then there exists a vertex $m \in (C \cap N_G(c))$ such that $\bar{x}_m = 0$. Given any set $D \in \Gamma(C \setminus \{m\})$, set $C' = (C \setminus \{m\}) \cup D$ is an independent 2-dominating set in G for which $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i + \bar{x}_c > 1$. This is a contradiction with $\bar{x} \in P_G$. On the other hand, if $b \in C$ then for any set $D \in \Gamma((C \cup \{a\}) \setminus N_G(a))$, set $C' = (C \cup \{a\} \cup D) \setminus N_G(a)$ is an independent 2-dominating set in G for which $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i - \bar{x}_b + \bar{x}_c + \bar{x}_a > 1$ using Claim 10. This again contradicts with $\bar{x} \in P_G$. So by Claim 8, we have $|N_G(c) \cap V'| \geq 3$.

Now if there exists at least one vertex $j \in (N_G(c) \cap V') \setminus \{b, d\}$ for which $\bar{x}_j = \bar{x}_c$ then for all $C \in \tau(S_1)$, $c \notin C$. To show this, suppose there exists a set $C \in \tau(S_1)$ which contains c. If the coefficient of x_b in the I2DS constraint corresponding to set C is negative then let set $W = N_G(j) \cup (\cup_{i \in N_G(j) \setminus \{c\}} N_G(i))$. For any set $D \in \Gamma((C \setminus W) \cup \{j\})$, set $C' = (C \setminus W) \cup \{j\} \cup D$ is an independent 2-dominating set in G for which $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i + \bar{x}_b > 1$ (Note that $(N_G(j) \cap V') \setminus \{c\}$ are all leaf vertices of $\bar{G}$ because otherwise p is not the longest path in this graph). This contradicts with $\bar{x} \in P_G$. If

the coefficient of x_b in the I2DS constraint corresponding to set C is zero, then define set $W = N_G(a) \cup N_G(j) \cup (\cup_{i \in N_G(j) \setminus \{c\}} N_G(i))$. For any set $D \in \Gamma((C \setminus W) \cup \{a, j\})$, set $C' = (C \setminus W) \cup \{a, j\} \cup D$ is an independent 2-dominating set in G for which $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i + \bar{x}_a > 1$. Again this is a contradiction with $\bar{x} \in P_G$. So by Claim 9, we have $\bar{x}_j < \bar{x}_c$ for all $j \in (N_G(c) \cap V') \setminus \{b, d\}$.

Now having $\bar{x}_j < \bar{x}_c$ for all $j \in (N_G(c) \cap V') \setminus \{b, d\}$ and $\bar{x}_b \leq \bar{x}_c$ (By Claim 10), we claim that for any $C \in \tau(S_1)$, the coefficient of x_c in the I2DS constraint corresponding to set C is nonnegative. To show this, suppose there exists a set $C \in \tau(S_1)$ for which this coefficient is negative. If $b \in C$ then for any set $D \in \Gamma((C \setminus N_G(a)) \cup \{a\})$, set $C' = (C \setminus N_G(a)) \cup \{a\} \cup D$ is an independent 2-dominating set in G for which $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i + \bar{x}_a - \bar{x}_b + \bar{x}_c > 1$. This contradicts with $\bar{x} \in P_G$. If $b \notin C$ then there exists at least one vertex $m \in (C \cap N_G(c))$ with $\bar{x}_m < \bar{x}_c$. Given a set $D \in \Gamma(C \setminus \{m\})$, set $C' = (C \cup D) \setminus \{m\}$ will be an independent 2-dominating set in G for which $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i - \bar{x}_m + \bar{x}_c > 1$. This is again a contradiction with $\bar{x} \in P_G$. So by Claim 8, we have $V' = \emptyset$ which is a contradiction. This completes the proof of Theorem 11. ■

5.4 Odd-mod-3 cycles

Naturally, there are classes of graphs for which the collection of I2DS facets and non-negativity constraints are not sufficient to completely describe their 2-club polytope. Theorem 12 describes a facet inducing inequality for the 2-club polytope of odd-mod-3 cycles that is distinct from I2DS facets.

Theorem 12 *If G=(V,E) is a cycle of size n where* $n \geq 7$ *and* $n \not\equiv 0(\mod 3)$ *then the following inequality induces a facet of the 2-club polytope of G.*

$$\sum_{i \in V} x_i \leq 3 \tag{5.6}$$

Proof. Without loss of generality, suppose elements of set V are numbered from 1 to n. It is easy to verify that the largest 2-club size in G is 3 and Inequality 5.6 is valid for the 2-club polytope of this graph. For each $i \in V$, define a 0-1 vector $\hat{X}^i = \{\hat{x}^i_1, \hat{x}^i_2, \cdots, \hat{x}^i_n\}$ in which $\hat{x}^i_k = 1$ for all $k \in \{i\} \cup N_G(i)$ and $\hat{x}^i_k = 0$ otherwise. Since $n \not\equiv 0(\mod 3)$, it can be verified that $\hat{X}^1, \hat{X}^2, \cdots, \hat{X}^n$ are linearly independent incidence vectors of 2-clubs in G for which Inequality 5.6 holds as equality. ■

CHAPTER 6

MAXIMUM 2-CLUBS UNDER UNCERTAINTY

This chapter focuses on the maximum 2-club problem under uncertainty. Specifically, given a graph subject to probabilistic edge failures, we are interested in finding large "risk-averse" 2-clubs in this graph. Here, risk aversion is achieved by modeling the loss in 2-club property under probabilistic edge failures as a random loss function of the decision and uncertainty, and utilizing Conditional Value-at-Risk (CVaR) as a quantitative measure of risk. The well-known Benders' decomposition scheme [67] is utilized to develop a new decomposition algorithm for solving the CVaR constrained maximum 2-club problem. A preliminary numerical experiment is also used to compare the computational performance of the developed algorithm with our extension of an existing algorithm recently introduced in literature.

6.1 Background on Conditional Value-at-Risk (CVaR)

Conditional Value-at-Risk (CVaR) is a quantitative measure of risk in a system where random losses exceed a threshold ([68] and [69]). A loss function $L(x, Y)$ quantifies losses as a function of a decision vector x and uncertainty, represented by a random vector Y. Then, for $\alpha \in (0, 1)$, α-Value-at-Risk (VaR) is the α-quantile of the loss distribution $\Psi(x, \ell)$, that is,

$$\alpha\text{-VaR}[L(x, Y)] = \alpha\text{-VaR}(x) = \inf\{\ell : \Psi(x, \ell) \geq \alpha\}.$$

The α-CVaR is the conditional expectation of losses exceeding α-VaR. That is,

$$\alpha\text{-CVaR}[L(x,Y)] = \alpha\text{-CVaR}(x) = E_Y[L(x,Y)|L(x,Y) \geq \alpha\text{-VaR}(x)].$$

Figure 6.1 shows an illustration of the CVaR concept.

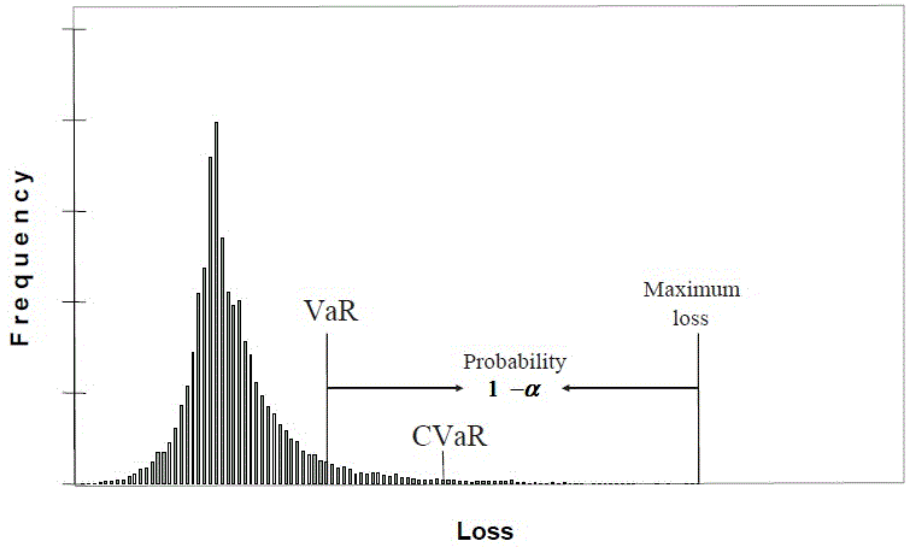

Figure 6.1: Illustration of the CVaR concept

The available literature on handling CVaR in optimization problems can be divided into two main categories which are, (1) the research work focused on finding a feasible decision that minimizes CVaR (CVaR minimization) and (2) the literature on finding the best (minimum cost) solution for which CVaR is bounded above by a user specified factor (CVaR constrained optimization).

6.1.1 CVaR minimization

Rockafellar and Uryasev [68, 69] pioneered the use of CVaR in optimization through a new approach to optimizing a portfolio of financial instruments. The focus of this work is on minimizing CVaR rather than minimizing VaR. A new technique for simultaneous calculation of VaR and optimization of CVaR for a broad class of problems is proposed in [70].

A method for credit risk optimization based on CVaR is introduced in [71]. The proposed model can simultaneously adjust all positions in a portfolio of financial

instruments in order to minimize CVaR subject to trading and return constraints. The credit risk distribution is generated by Monte Carlo simulations and the optimization problem is solved effectively by utilizing LP techniques.

The Benders' decomposition method is modified in [72] by using concepts from the Reformulation-Linearization Technique (RLT) and lift-and-project cuts in order to develop an approach for solving discrete optimization problems that yield integral subproblems. The authors demonstrated how cutting planes could be generated to derive a partial description of the convex hull representation as needed, in order to devise a finitely convergent solution procedure.

The non-convex minimization problem of the VaR that arises from financial risk analysis is considered in [73]. By considering this problem as a special linear program, the authors developed upper- and lower-bounds for the minimum VaR.

Optimization problems for minimizing CVaR from a computational point of view, with an emphasis on financial applications are considered in [74]. The authors reformulated these CVaR optimization problems as two-stage stochastic programming problems with recourse. Specializing the L-shaped method leads to a new algorithm for minimizing CVaR which the authors call *CVaRMin*. Minimizing CVaR as a mean-risk function is shown to be computationally tractable in [75]. A generic cutting plane algorithm for solving CVaR minimization problems with convex loss functions is also presented in [75]. An LP formulation of the minimization of CVaR measure defined with two different loss functions and the optimal solution for this particular problem are presented in [76].

A portfolio selection model in which the methodologies of robust optimization are used for minimization of CVaR of a portfolio of shares is investigated in [77]. An important feature of this approach is the use of robust optimization techniques to deal with uncertainty, in place of stochastic programming as proposed by [68]. Different approaches for the generation of input data, with special attention to the estimation

of expected returns are also suggested in this work.

The minimization of CVaR of the end-of-horizon yield as a two-stage model is presented in [78] and a decomposition technique for solving this problem is proposed in this work. The master problem in this decomposition is solved by an inexact version of the level method [79]. A two-phase approach that is suitable for solving CVaR minimization problems in portfolio optimization with a large number of price scenarios is proposed in [80]. In the first phase, conventional differentiable optimization techniques are used while circumventing non-differentiable points. In the second phase, a theoretically convergent, variable target value, non-differentiable optimization technique is employed.

6.1.2 CVaR constrained optimization

The portfolio optimization approach proposed by [68] is extended in [81] to address optimization problems with CVaR constraints. This approach is based on an optimization scheme for calculating VaR and optimizing CVaR simultaneously. This extended approach can be used to maximize the expected returns under CVaR constraints. The Conditional Tail Expectation (CTE) is investigated in [82]. The authors showed that CTE is a robust, convenient, practical, and coherent measure for quantifying financial risk exposure. The authors also considered some statistical properties of the methods that are commonly used to estimate the CTE and developed a simple formula for the variance of the CTE estimator that is valid in the large sample limit.

A two-step process that can handle portfolio optimization problems with variance terms in the objective function and a large number of CVaR constraints is introduced in [78]. In the first step, an approximation of the efficient frontier is constructed in order to help the decision maker in setting the upper-bounds on CVaR constraints. To this aim, the dual problem is decomposed and solved by the technique introduced in [78]. After finding appropriate parameters, in the second step, the dual optimal

solution is used in order to find an optimal solution of the primal problem. The CVaR constrained minimization is formulated as a two-stage stochastic programming problem with relatively complete recourse in [78]. The first stage problem is solved by the inexact level method [79] and a sharp-constrained version of the constrained level method [83] is used to find the solution of the second-stage problem.

An algorithm called "Iterative Estimation Maximization (IEM)" is presented in [84] to solve stochastic linear and convex programs with CVaR constraints. IEM iteratively constructs compact-sized, linear or convex optimization problems and solves them sequentially to find the optimal solution. The authors proved that IEM converges to the true optimal solution and gives a lower-bound on the number of samples required to probabilistically satisfy a CVaR constraint.

A decomposition algorithm for single-stage, stochastic linear programs with multiple CVaR constraints is provided in [85]. The proposed decomposition algorithm is based on a large polyhedral representation of the problem's feasible region followed by a column generation routine in the dual space of the proposed representation. In case of CVaR minimization, the proposed technique results in the aforementioned CVaRMin algorithm [74]. Furthermore, a scheme which utilizes the proposed decomposition algorithm is developed to address mixed-integer LP problems with multiple CVaR constraints.

6.2 The CVaR constrained maximum 2-club problem

The focus of this section is on the CVaR constrained maximum 2-club problem under probabilistic edge failures. Here, the support graph $G = (V, E)$ is subject to probabilistic edge failures meaning an edge $(i, j) \in E$ has a probability $p_{ij} > 0$ of survival. Define set P as follows.

$$P = \Big\{x \in \{0,1\}^n : x_i + x_j - \sum_{k \in N(i) \cap N(j)} x_k \leq 1 \quad \forall (i,j) \notin E\Big\},$$

is the set containing the incidence vectors of all 2-clubs in the support graph G. A random vector $Y \in \{0,1\}^{|E|}$ represents the edge set under uncertainty in which $Pr\{Y_{ij} = 1\} = p_{ij} > 0$ for all $(i,j) \in E$. We assume for a given 2-club S in the support graph G with incidence vector $x^0 \in P$, we incur a loss denoted by $L(x^0, Y)$ which is a random variable equal to the total number of pairs of nodes in S with distance more than two in $G[S]$ after the realization of uncertainty. For a given $x^0 \in P$, this loss function $L(x^0, Y)$ is given by,

$$L(x^0, Y) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (\phi(x^0, Y, i, j))^+$$

where

$$\phi(x^0, Y, i, j) = x_i^0 + x_j^0 - 1 - y_{ij} - \sum_{l \in N(i) \cap N(j)} y_{il} y_{jl} x_l^0, \qquad \text{if } \ (i,j) \in E,$$

and

$$\phi(x^0, Y, i, j) = x_i^0 + x_j^0 - 1 - \sum_{l \in N(i) \cap N(j)} y_{il} y_{jl} x_l^0 \qquad \text{if } \ (i,j) \notin E.$$

For a given realization of the random vector Y (say y^0), $L(x, y^0)$ is a piecewise linear convex function of x. Given an incidence vector of a 2-club (say x^0), let the set $S(x^0, y^0) = \{(i,j) : i \in \{1, \ldots, n\}, \ \ j \in \{i+1, \ldots, n\}$ and $\phi(x^0, y^0, i, j) > 0\}$. A supporting hyperplane of $L(x, y^0)$ at the point x^0, can be described as follows.

$$h_{(x^0, y^0)}(x) = \sum_{(i,j) \in S(x^0, y^0)} \phi(x, y^0, i, j)$$

Formulation (6.1)-(6.3) describes the CVaR constrained maximum 2-club problem which is to find the incidence vector of a largest 2-club in G (say x^*) for which α-CVaR$[L(x^*, Y)]$ is bounded above by a user specified parameter d.

$$\min \quad f \quad = \quad \textbf{-1}^T x \tag{6.1}$$

Subject to

$$\alpha\text{-CVaR}[L(x,Y)] \quad \leq \quad d, \tag{6.2}$$

$$x \quad \in \quad P \tag{6.3}$$

According to [68, 69], α-CVaR[.] can be reformulated as follows.

$$\alpha\text{-CVaR}[L(x,Y)] = min_{\zeta}\{\zeta + \frac{1}{1-\alpha}E[(L(x,Y)-\zeta)^{+}]\}$$

If the distribution of the random variable Y is approximated by a set of N samples with π_k, $k \in \{1,\ldots,N\}$ as their normalized sample probabilities [68, 69], α-CVaR[.] reformulation can be approximated as follows.

$$\alpha\text{-CVaR}[L(x,Y)] \simeq min_{\zeta}\{\zeta + \frac{1}{1-\alpha}\sum_{k=1}^{N}\pi_k(L(x,y_k)-\zeta)^{+}\}$$

Accordingly, (6.1)-(6.3) can be reformulated as,

$$\min \quad f \quad = \quad \textbf{-1}^T x \tag{6.4}$$

Subject to

$$min_{\zeta}\{\zeta + \frac{1}{1-\alpha}\sum_{k=1}^{N}\pi_k\,(L(x,y_k)-\zeta)^{+}\} \leq d, \tag{6.5}$$

$$x \quad \in \quad P \tag{6.6}$$

Based on a result from [81], the CVaR constraint in Formulation (6.4)-(6.6) can be replaced with the equivalent Inequality 6.8 resulting in the following formulation.

$$\min \quad f \quad = \quad \textbf{-1}^T x \tag{6.7}$$

Subject to

$$\zeta + \frac{1}{1-\alpha} \sum_{k=1}^{N} \pi_k \left(L(x, y_k) - \zeta\right)^+ \leq d, \tag{6.8}$$

$$x \quad \in \quad P \tag{6.9}$$

6.3 A decomposition algorithm

This section contains the details of our proposed decomposition technique for solving CVaR constrained integer programming (IP) problems with loss functions that are piecewise linear and convex with respect to the decision (x) for a given realization of uncertainty (y^0). The CVaR constrained IP problem of interest in this dissertation is the CVaR constrained maximum 2-club problem modeled as formulation (6.7)-(6.9). In order to present Algorithm 1, first we reformulate (6.7)-(6.9) as follows.

$$\min \quad f \quad = \quad \textbf{-1}^T x \tag{6.10}$$

Subject to

$$\zeta + \frac{1}{1-\alpha} \sum_{k=1}^{N} z_k \quad \leq \quad d, \tag{6.11}$$

$$z_k \geq \pi_k (L(x, y_k) - \zeta) \quad \forall k \in \{1, \ldots, N\} \tag{6.12}$$

$$z_k \quad \geq \quad 0 \quad \forall k \in \{1, \ldots, N\} \tag{6.13}$$

$$x \quad \in \quad P \tag{6.14}$$

Now, we use the technique introduced by Benders [67] to decompose and solve problem (6.10)-(6.14). Consider the following reformulation.

$$\min \quad f \quad = \quad \mathbf{-1}^T x + G(x) \tag{6.15}$$

Subject to

$$x \quad \in \quad P \tag{6.16}$$

where $G(x)$ is a large-scale linear programming problem defined as follows.

$$G(x) \quad = \quad \min \quad 0 \tag{6.17}$$

Subject to

$$\zeta + \frac{1}{1-\alpha} \sum_{k=1}^{N} z_k \quad \leq \quad d, \tag{6.18}$$

$$\pi_k \zeta + z_k \quad \geq \quad \pi_k L(x, y_k) \quad \forall k \in \{1, \ldots, N\} \tag{6.19}$$

$$z_k \quad \geq \quad 0 \quad \forall k \in \{1, \ldots, N\} \tag{6.20}$$

Formulation (6.21)-(6.25) is the dual of the LP formulation (6.17)-(6.20).

$$G'(x) = \max \quad dw + \sum_{k=1}^{N} \pi_k L(x, y_k) p_k \tag{6.21}$$

Subject to

$$w + \sum_{k=1}^{N} \pi_k p_k \quad = \quad 0, \tag{6.22}$$

$$\frac{1}{1-\alpha} w + p_k \quad \leq \quad 0 \quad \forall k \in \{1, \ldots, N\} \tag{6.23}$$

$$w \quad \leq \quad 0 \tag{6.24}$$

$$p_k \quad \geq \quad 0 \quad \forall k \in \{1, \ldots, N\} \tag{6.25}$$

which is equivalent to,

$$G'(x) = \max \quad \sum_{k=1}^{N} \pi_k (L(x, y_k) - d) p_k \tag{6.26}$$

Subject to

$$-\sum_{k=1}^{N} \pi_k p_k + (1-\alpha) p_k \leq 0 \quad \forall k \in \{1, \ldots, N\} \tag{6.27}$$

$$p_k \geq 0 \quad \forall k \in \{1, \ldots, N\} \tag{6.28}$$

For any given $x \in P$, the feasible region of problem (6.26)-(6.28) is a polyhedral cone with one extreme point which is the origin (the problem is always feasible). In case an optimal solution exists, then $G'^*(x) = 0$, and if the problem is unbounded then there does not exist $z \in \mathbb{R}_+^N$ and $\zeta \in \mathbb{R}$ such that (x, z, ζ) is a feasible solution to problem (6.10)-(6.14). So we are interested in $x \in P$ for which the problem (6.26)-(6.28) has an optimal solution (zero). So problem (6.10)-(6.14) can be reformulated as the following convex nonlinear integer programming problem,

$$\min \quad f = \mathbf{-1}^T x \tag{6.29}$$

Subject to

$$x \in P \tag{6.30}$$

$$\sum_{k=1}^{N} \pi_k (L(x, y_k) - d) r_k^{(j)} \leq 0 \quad \forall j \in ER \tag{6.31}$$

where ER is the index set of all extreme rays (denoted by $(r_1^{(j)}, \ldots, r_N^{(j)})$) of the polyhedral cone in problem (6.26)-(6.28). Since the number of extreme rays might be large, solving (6.29)-(6.31) will be a challenging task.

Algorithm 1 for solving problem (6.29)-(6.31) is described as follows. Let us refer to problem (6.26)-(6.28) as the subproblem. Given a $x \in P$, the subproblem is a LP problem with a large number of variables and constraints. Algorithm 1 is a row generation based algorithm. This algorithm starts by solving a relaxation of problem (6.29)-(6.31) which does not include the constraints (6.31). Let us refer

to this problem as the relaxed integer programming (RIP) problem. If the solution found for RIP problem is feasible to original problem (6.29)-(6.31) then the algorithm terminates; otherwise, the subproblem will be solved in order to find a violated valid inequality. Then, the RIP problem will be updated by adding the cutting plane identified and this procedure will be repeated. If during an iteration of this algorithm, the RIP problem becomes infeasible then problem (6.29)-(6.31) is infeasible and the algorithm terminates.

Algorithm 1 Decomposition algorithm for solving problem (6.29)-(6.31)

1: **procedure** RowGen$(G, (\pi_1, \ldots, \pi_N), (y_1, \ldots, y_N), d, \alpha)$

2: Construct RIP problem by removing all constraints (6.31) from problem (6.29)-(6.31) and let $t = 1$.

3: Solve RIP problem by a IP solver. If the problem is infeasible then **return** by infeasibility. Otherwise suppose $x^{*(t)}$ is the obtained optimal solution.

4: Solve the subproblem $G'(x^{*(t)})$ by a LP solver. If the subproblem has an optimal solution then **return** by $x^{*(t)}$ as the optimal solution to problem (6.29)-(6.31). Otherwise let $\bar{j} \in ER$ be the extreme ray found by the LP solver.

5: Update RIP problem by adding constraint $\sum_{k=1}^{N} \pi_k (h_{(x^{*(t)}, y_k)}(x) - d) r_k^{(\bar{j})} \leq 0$. Let $t = t + 1$ and go to step 3.

6: **end procedure**

The computational performance of Algorithm 1 is compared with that of an existing algorithm recently introduced in the literature for solving CVaR constrained IP problems [85]. The algorithm introduced in [85] is also a decomposition algorithm based on a large polyhedral representation of the problem's feasible region followed by a column generation routine in the dual space of the proposed representation. This algorithm is developed in order to solve CVaR constrained IP problems with loss functions that are linear with respect to x for a given y^0.

The loss function used to model the CVaR constrained maximum 2-club problem

in this chapter, is a piecewise linear convex function of x. Therefore, the algorithm proposed in [85] is slightly modified in this dissertation in order to handle IP problems with convex piecewise linear loss functions with respect to the decision vector for a given realization of uncertainty. This algorithm is described next.

According to [85], constraint (6.8) can be reformulated as the following inequality.

$$(1-\alpha)\zeta + \sum_{k=1}^{N} max[\pi_k(L(x,y_k)-\zeta),0] \leq d(1-\alpha) \tag{6.32}$$

An equivalent representation of constraint (6.32) is the following,

$$(1-\alpha)\zeta + \max_{A\in\ 2^{\mathcal{N}}} \sum_{k\in A} \pi_k(L(x,y_k)-\zeta) \ \leq \ d(1-\alpha) \tag{6.33}$$

$$\zeta \ \leq \ d \tag{6.34}$$

where $2^{\mathcal{N}}$ is the power set of $\mathcal{N} = \{1,\ldots,N\}$, which is the index set of the discrete sampling approximation. Now, constraint (6.33) can be reformulated as the following large constraint set.

$$(1-\alpha)\zeta + \sum_{k\in A} \pi_k(L(x,y_k)-\zeta) \ \leq \ d(1-\alpha) \ \ \forall A \in \ 2^{\mathcal{N}} \tag{6.35}$$

Finally, problem (6.7)-(6.9) can be reformulated as follows.

$$\min \ \ f \ = \ \mathbf{-1}^T x \tag{6.36}$$

Subject to

$$(1-\alpha)\zeta + \sum_{k\in A} \pi_k(L(x,y_k)-\zeta) \ \leq \ d(1-\alpha) \ \ \forall A \in \ 2^{\mathcal{N}} \tag{6.37}$$

$$\zeta \ \leq \ d \tag{6.38}$$

$$x \ \in \ P \tag{6.39}$$

Let LPR denote the linear programming relaxation of formulation (6.36)-(6.39). Let us also denote by RIP_2, a relaxation of IP problem (6.36)-(6.39) in which all

constraints (6.37) have been removed. Algorithm 2 shows the modified version of the decomposition technique introduced in [85] for solving CVaR constrained IP problems with piecewise linear convex loss functions.

Algorithm 2 Decomposition algorithm for solving problem (6.36)-(6.39)

1: **procedure** RowGen$(G, (\pi_1, \ldots, \pi_N), (y_1, \ldots, y_N), d, \alpha)$

2: Construct a relaxation of LPR (denoted by RLPR), by removing all constraints (6.37) from LPR and let $t = 1$.

3: Solve RLPR using a standard LP solver. If the problem is infeasible, terminate and declare infeasibility. Else, let the optimal solution be $(x^{*(t)}, \zeta^{*(t)})$. Compute the subset $A^* = \{k \in \mathcal{N} \mid L(x^{*(t)}, y_k) - \zeta^{*(t)} > 0\}$.

4: If $\sum_{k \in A^*} \pi_k (L(x^{*(t)}, y_k) - \zeta^{*(t)}) + (\zeta^{*(t)} - d)(1 - \alpha) \leq 0$ then $(x^{*(t)}, \zeta^{*(t)})$ is the optimal solution to LPR and go to Step (6). Else, go to Step (5).

5: Add constraint $(1 - \alpha)\zeta + \sum_{k \in A^*} \pi_k (h_{(x^{*(t)}, y_k)}(x) - \zeta) \leq d(1 - \alpha)$ to RLPR. Set $t = t + 1$, and go to Step (3).

6: Let μ_t denote the collection of all $t - 1$ constraints added to RLPR in step (5). Add all constraints in μ_t to RIP_2 and re-initialize $t = 1$.

7: For all $t \geq 1$, solve RIP_2 using a standard IP solver. If the problem is infeasible, terminate and declare infeasibility. Else, let the optimal solution be $(x^{*(t)}, \zeta^{*(t)})$. Compute the subset $A^* = \{k \in \mathcal{N} \mid L(x^{*(t)}, y_k) - \zeta^{*(t)} > 0\}$.

8: If $\sum_{k \in A^*} \pi_k (L(x^{*(t)}, y_k) - \zeta^{*(t)}) + (\zeta^{*(t)} - d)(1 - \alpha) \leq 0$ then $(x^{*(t)}, \zeta^{*(t)})$ is the optimal solution to problem (6.36)-(6.39) and terminate. Else, go to Step (9).

9: Add constraint $(1 - \alpha)\zeta + \sum_{k \in A^*} \pi_k (h_{(x^{*(t)}, y_k)}(x) - \zeta) \leq d(1 - \alpha)$ to RIP_2. Set $t = t + 1$, and go to Step (7).

10: **end procedure**

6.4 A brief numerical study

This section presents preliminary numerical results obtained by solving the CVaR constrained maximum 2-club problem, modeled as formulation (6.7)-(6.9), by Algorithms 1 and 2 on a randomly generated sample. Computational performance of Algorithm 1 is then compared to that of Algorithm 2 in terms of running time and number of iterations.

Both algorithms were implemented in C++ and all numerical experiments were conducted on an HP Z400 workstation with Intel® Xeon® $W3520$ @ 2.67 GHz processor and 3.00 GB RAM. The commercial solver IBM ILOG CPLEX Optimizer 11.2® was used to solve the associated IP and LP formulations. The selected instance is a graph of order $n = 50$ randomly generated by the algorithm used in [43]. As mentioned in Section 4.3, the edge density of the graphs produced by this algorithm is controlled by two parameters a and b. The expected edge density d is $(a + b)/2$ and vertex degree variance (VDV) increases with $b-a$. For our experiment in this section, we considered density $d = 0.15$ with $a = b = 0.15$.

The probability of survival of an arc was generated by using a uniform random distribution between 0 and 1. In our experiment, we considered $\alpha \in \{0.7, 0.8, 0.9, 0.99\}$ and $d \in \{5, 10, 15, 25\}$ resulting in a total of 16 combinations. We generated 100 random scenarios in order to model our problem and used the same set of scenarios with all 16 combinations of α and d. The time limit for each iteration of both algorithms was set to be 1800 seconds and the running time limit of each main decomposition algorithm was 10800 seconds. Upon reaching the time limit, either for one iteration or for the main decomposition algorithm, the main algorithm terminates and an upper-bound on the optimal solution will be reported. Table 6.1 shows the computational results obtained by solving the CVaR constrained maximum 2-club problem using Algorithms 1 and 2 on the selected test instance.

Before discussing the results, we emphasize the fact that this experiment is on a

single numerical instance of the problem (one graph on 50 vertices with 100 scenarios) and hence, no general deductions should be drawn. It was conducted with the intent of testing the codes developed for the algorithms discussed here, and to verify if some basic characteristics evident from the design of these algorithms are observed with an actual implementation, especially the sensitivity of the 2-club solutions to the choice of α and d. Note that we get more risk-averse as α increases or as d, decreases. The problem may be infeasible if d is sufficiently small and the CVaR constraint may be redundant if d is sufficiently large. A comprehensive experimental study will be reported in a forthcoming paper [86].

According to Table 6.1, with respect to the running time, Algorithm 1 outperforms Algorithm 2 in 12 problem instances out of 16. In the other 4 problem instances, Algorithm 2 performs better. In terms of number of iterations, in 6 problem instances out of 16, Algorithm 1 performs better than Algorithm 2, in 2 problem instances their performance is identical and in the other 8 problem instances, Algorithm 2 has smaller number of iterations.

It seems that Algorithm 2 finds the optimal solution in smaller number of iterations compared to Algorithm 1 while it takes more time to terminate. We suspect that Algorithm 2 spends more time in each iteration which might be the result of the computational effort needed to find a violated inequality.

The problem instances with higher α and lower d seem to be more challenging in terms of running time and number of iterations for both algorithms. For the most challenging problem instance ($\alpha = 0.99$ and $d = 5$), Algorithm 1 is 1.5 times faster than Algorithm 2 in terms of running time.

Figure 6.2 shows the solutions found for all 16 combinations of α and d by Algorithm 1. The arcs shown by darker color have higher survival probabilities and the color of arcs with smaller probabilities of survival are lighter. As shown in Table 6.1 and by Figure 6.2, for a given d, as α increases, the size of the optimal solution

Table 6.1: Computational results obtained by solving CVaR constrained maximum 2-club problem using Algorithms 1 and 2 on the selected test instance

Setting	Iterations		Solution Size		Running Time (Sec)	
	ALG 1	ALG 2	ALG 1	ALG 2	ALG 1	ALG 2
(α=0.99, d=5)	1211	1466	5	5	6381.08	9988.94
(α=0.9, d=5)	824	834	6	6	3540.75	3982.47
(α=0.8, d=5)	785	618	6	6	3178.73	2727.19
(α=0.7, d=5)	630	555	6	6	2426.61	2476.97
(α=0.99, d=10)	1223	1187	7	7	6195.95	6499.78
(α=0.9, d=10)	634	664	7	7	2043.05	2627.05
(α=0.8, d=10)	744	568	7	7	2801.49	2191.09
(α=0.7, d=10)	623	592	7	7	1975.14	2533.11
(α=0.99, d=15)	775	464	8	8	2597.94	1339.56
(α=0.9, d=15)	405	557	8	8	915.485	2353.56
(α=0.8, d=15)	448	622	8	8	1061.27	3694.53
(α=0.7, d=15)	388	386	8	8	899.11	1645.02
(α=0.99, d=25)	474	324	9	9	730.593	423.844
(α=0.9, d=25)	89	232	10	10	52.922	452.625
(α=0.8, d=25)	69	69	10	10	53.422	78.5
(α=0.7, d=25)	60	60	10	10	36.735	51.19

decreases. On the other hand, for a given α, as d increases, the size of the optimal solution increases. This behavior of the size of the optimal solution as a function of α and d is intuitively acceptable since decreasing α or increasing d relaxes constraint 6.2 which may result in a larger feasible region with larger 2-clubs.

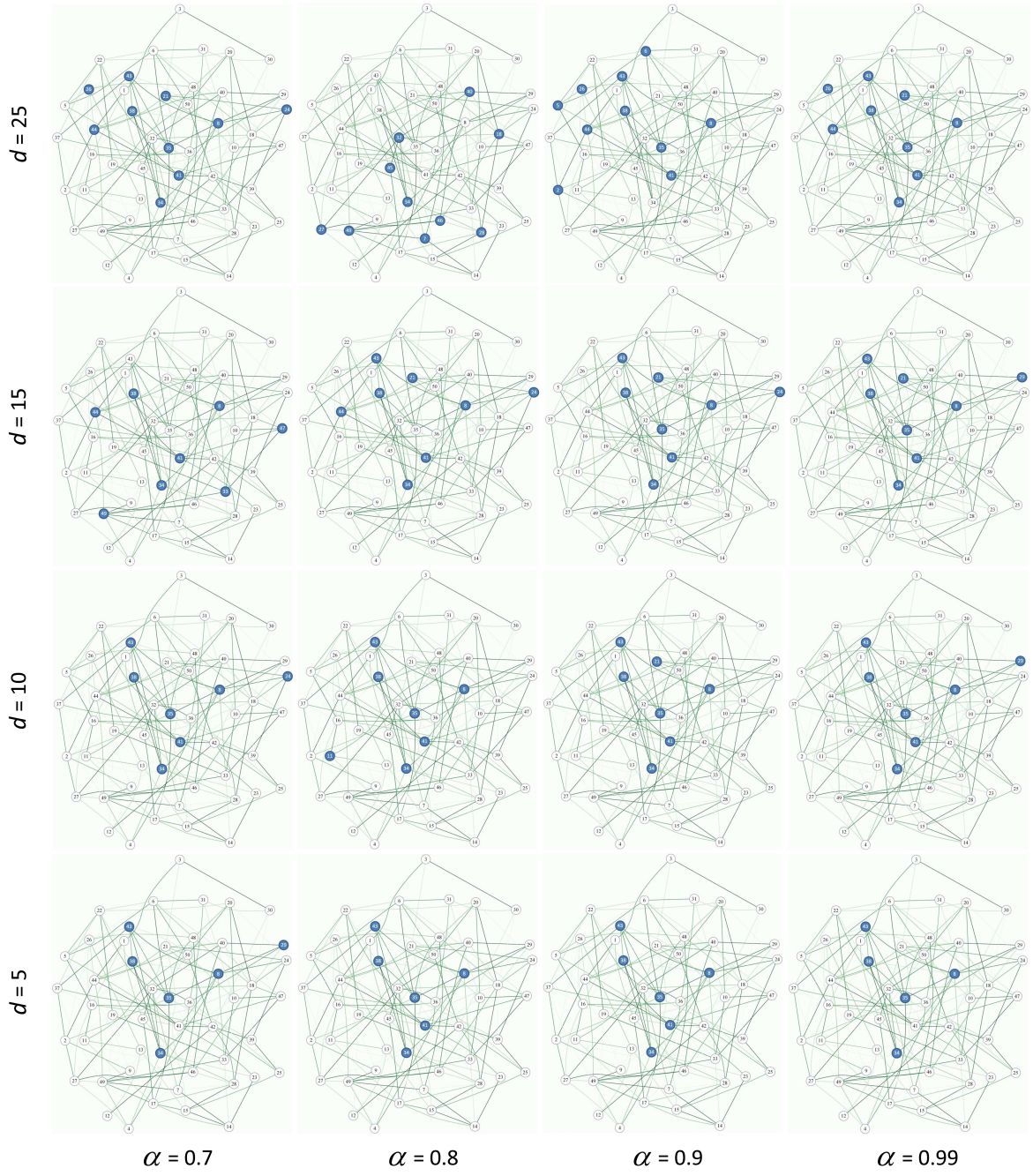

Figure 6.2: Solutions found for all 16 combinations of α and d by Algorithm 1

Figure 6.3 illustrates the solution found by Algorithm 1 for problem instance with $\alpha = 0.7$ and $d = 10$, where the edges with both endpoints in the 2-club found have higher survival probability (darker color).

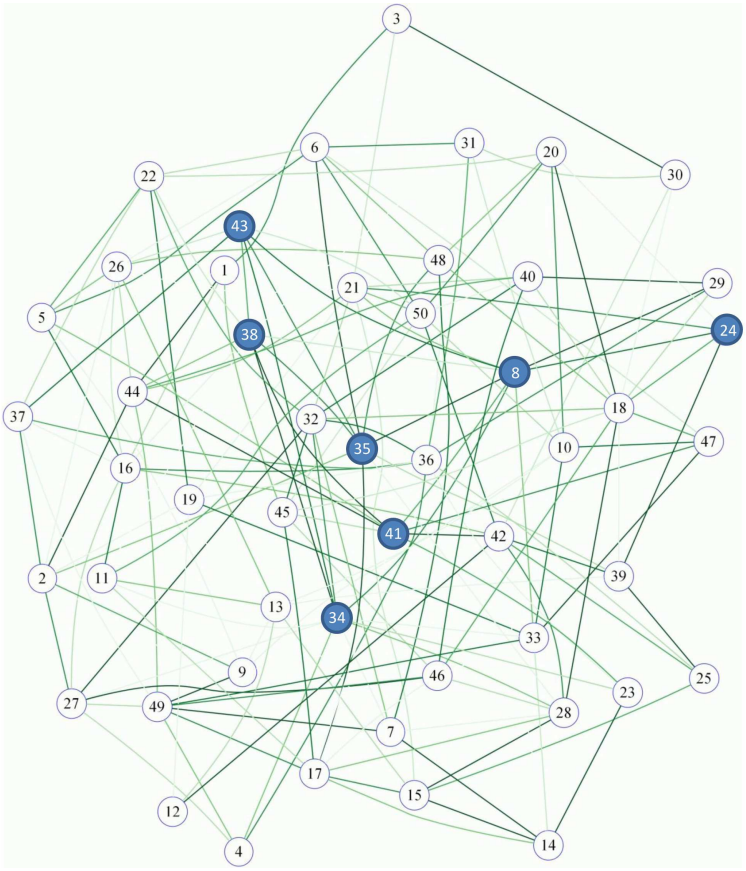

Figure 6.3: Solution found by Algorithm 1 for problem instance with $\alpha = 0.7$ and $d = 10$

CHAPTER 7

CONTRIBUTIONS AND FUTURE WORK

The k-clubs can be used to effectively model low-diameter clusters in graph-based data mining applications in biology, internet analytics and social sciences. This chapter summarizes our contributions in developing theory and algorithms for the optimization problems related to k-clubs in graph models of data, and provides some directions for future research in this area.

7.1 Contributions

In this dissertation, we settled a long remaining open problem and proved that k-club maximality testing is NP-complete for $k \geq 2$. Intractability of k-club maximality testing is due to their nonhereditary nature which imposes significant challenges in developing theory and algorithms for k-clubs. The nonhereditary property of k-clubs and its implications is discussed in great detail in this document. A class of graphs with polynomially verifiable maximal k-clubs has also been identified. A dual coloring based upper-bounding technique and a bounded enumeration based lower-bounding strategy for the k-club number of a graph have been proposed. A new combinatorial branch-and-bound framework for solving the maximum k-club problem is then developed and the computational performance of this algorithm with four different combinations of lower- and upper-bounding schemes is studied. It is shown that the branch-and-bound algorithm which utilizes the proposed bounding techniques, outperforms the other algorithms on challenging test instances.

The 2-club polytope of a graph is studied in this dissertation and a new family

of facet inducing inequalities for this polytope (I2DS inequalities) is presented. This family of facets unifies all known nontrivial facets of the 2-club polytope, and also introduces distinct new facets of this polytope to the literature. The separation complexity of the I2DS inequalities is proved to be NP-complete and it is shown that these facets, along with the nonnegativity constraints completely describe the 2-club polytope of trees. A facet distinct from the I2DS facets is demonstrated for odd-mod-3 cycles.

The maximum 2-club problem under uncertainty is also studied in this dissertation. Given a graph subject to probabilistic edge failures, the goal here is to find large "risk-averse" 2-clubs. To achieve risk aversion, the loss has been modeled as a random variable which is a function of the decision variables and uncertain parameters. Conditional Value-at-Risk (CVaR) of losses is then utilized as a quantitative measure of risk. A new decomposition algorithm for solving the CVaR constrained maximum 2-club problem is developed by utilizing Benders' decomposition scheme [67]. The computational performance of the developed algorithm is compared with the one for an existing algorithm [85] in literature in a brief numerical study that also demonstrates the sensitivity of the size of the optimal 2-clubs to parameters in the model that control risk aversion.

7.2 Future work

It will be beneficial to develop a branch-and-cut algorithm for solving the maximum 2-club problem using the I2DS facet inducing inequalities. Identifying graph classes on which maximum and maximal k-clubs can be found in polynomial-time is an interesting challenge.

Studying the 2-club polytope in order to discover more facet inducing inequalities would also be another research direction. Identifying graph classes (other than trees) for which the 2-club polytope can be completely described by the I2DS inequalities

and nonnegativity constraints, is another direction for future research.

It is also valuable to develop metaheuristic approaches for detecting k-clubs in large-scale real-life graphs, especially power-law graphs from bioinformatics applications. Validating the biological significance of the detected k-clubs in such biological networks is also another interesting direction for future work.

Investigations into other variations of the k-club model (edge-weighted, directed, r-robust k-club [55]) would enrich the literature in this area. Developing theory and algorithms for partitioning, covering and enumerative extensions of k-clubs is another interesting research direction.

An immediate direction for future research is to study the performance of the developed decomposition algorithm for solving the CVaR constrained maximum 2-club problem on a larger test-bed of instances using high performance parallel computing.

BIBLIOGRAPHY

[1] P.-N. Tan, M. Steingach, and V. Kumar, *Introdiction to Data Mining*. Addison-Wesley, 2006.

[2] D. J. Cook and L. B. Holder, "Graph-based data mining," *IEEE Intelligent Systems*, vol. 15, no. 2, pp. 32–41, 2000.

[3] T. Washio and H. Motoda, "State of the art of graph-based data mining," *SIGKDD Explor. Newsl.*, vol. 5, no. 1, pp. 59–68, 2003.

[4] L. M. Camarinha-matos and H. Afsarmanesh, "Collaborative networks: a new scientific discipline," *Journal of Intelligent Manufacturing*, vol. 16, pp. 439–452, 2005.

[5] J. Grossman, P. Ion, and R. D. Castro, "The Erdös Number Project," 1995. Online: http://www.oakland.edu/enp/. Accessed June 2012.

[6] S. Hill, F. Provost, and C. Volinsky, "Network-based marketing: Identifying likely adopters via consumer networks," *Statistical Science*, vol. 22, pp. 256–275, 2006.

[7] D. Iacobucci and N. Hopkins, "Modelling dyadic interactions and networks in marketing," *Journal of Marketing Research*, vol. 24, pp. 5–17, 1992.

[8] T. Smieszek, L. Fiebig, and R. Scholz, "Models of epidemics: when contact repetition and clustering should be included," *Theoretical Biology and Medical Modelling*, vol. 6, no. 1, p. 11, 2009.

[9] A. Broido and K. C. Claffy, "Internet topology: connectivity of ip graphs," in *Scalability and Traffic Control in IP Networks* (S. Fahmy and K. Park, eds.), (Bellingham, WA), pp. 172–187, SPIE Publications, 2001.

[10] L. Terveen, W. Hill, and B. Amento, "Constructing, organizing, and visualizing collections of topically related, web resources," *ACM Transactions on Computer-Human Interaction*, vol. 6, pp. 67–94, 1999.

[11] J. Abello, P. M. Pardalos, and M. G. C. Resende, "On maximum clique problems in very large graphs," in *External memory algorithms and visualization* (J. Abello and J. Vitter, eds.), vol. 50 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pp. 119–130, American Mathematical Society, 1999.

[12] V. Boginski, S. Butenko, and P. M. Pardalos, "On structural properties of the market graph," in *Innovation in Financial and Economic Networks* (A. Nagurney, ed.), (London), Edward Elgar Publishers, 2003.

[13] V. Boginski, S. Butenko, and P. Pardalos, "Statistical analysis of financial networks," *Computational Statistics & Data Analysis*, vol. 48, pp. 431–443, 2005.

[14] V. Boginski, S. Butenko, and P. Pardalos, "Mining market data: a network approach," *Computers & Operations Research*, vol. 33, pp. 3171–3184, 2006.

[15] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *Proceedings of the National Academy of Sciences of the USA*, vol. 98, no. 8, pp. 4569–4574, 2001.

[16] J. Gagneur, R. Krause, T. Bouwmeester, and G. Casari, "Modular decomposition of protein-protein interaction networks," *Genome Biology*, vol. 5, no. 8, pp. R57.1–R57.12, 2004.

[17] V. Spirin and L. A. Mirny, "Protein complexes and functional modules in molecular networks," *Proceedings of the National Academy of Sciences*, vol. 100, no. 21, pp. 12123–12128, 2003.

[18] X. Peng, M. A. Langston, A. M. Saxton, N. E. Baldwin, and J. R. Snoddy, "Detecting network motifs in gene co-expression networks through integration of protein domain information," in *Methods of Microarray Data Analysis V* (P. McConnell, S. M. Lin, and P. Hurban, eds.), pp. 89–102, New York: Springer, 2007.

[19] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, pp. 651–654, 2000.

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

[21] R. Diestel, *Graph Theory*. Berlin: Springer-Verlag, 1997.

[22] D. West, *Introduction to Graph Theory*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[23] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York: W.H. Freeman and Company, 1979.

[24] J. Håstad, "Clique is hard to approximate within $n^{1-\epsilon}$," *Acta Mathematica*, vol. 182, pp. 105–142, 1999.

[25] R. D. Alba, "A graph-theoretic definition of a sociometric clique," *Journal of Mathematical Sociology*, vol. 3, no. 1, pp. 113–126, 1973.

[26] G. J. Chang and G. L. Nemhauser, "The k-domination and k-stability problems on sun-free chordal graphs," *SIAM Journal on Algebraic and Discrete Methods*, vol. 5, pp. 332–345, 1984.

[27] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[28] A.-L. Barabási, R. Albert, H. Jeong, and G. Bianconi, "Power-law distribution of the World Wide Web," *Science*, vol. 287, no. 5461, p. 2115a, 2000.

[29] A.-L. Barabási, R. Albert, and H. Jeong, "Scale-free characteristics of random networks: The topology of the World Wide Web," *Physica A*, vol. 281, no. 1-4, pp. 69–77, 2000.

[30] P. Erdös and A. Rényi, "On random graphs," *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.

[31] P. Erdös and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hungar. Acad. Sci.*, vol. 5, pp. 17–61, 1960.

[32] F. Chung and L. Lu, *Complex Graphs and Networks*. CBMS Lecture Series, Providence, RI: American Mathematical Society, 2006.

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

[34] D. Watts and S. Strogatz, "Collective dynamics of "small-world" networks," *Nature*, vol. 393, pp. 440–442, 1998.

[35] S. Milgram, "The small world problem," *Psychology Today*, vol. 1, pp. 61–67, 1967.

[36] J. Miao and D. Berleant, "From paragraph networks to document networks," in *Proceedings of the International Conference on Information Technology: Coding and Computing, 2004 (ITCC 2004)*, vol. 1, pp. 295–302, april 2004.

[37] B. Balasundaram, S. Butenko, and S. Trukhanov, "Novel approaches for analyzing biological networks," *Journal of Combinatorial Optimization*, vol. 10, no. 1, pp. 23–39, 2005.

[38] S. Pasupuleti, "Detection of protein complexes in protein interaction networks using n-clubs," in *In EvoBIO 2008: Proceedings of the 6th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pp. 153–164, Springer, 2008. volume 4973 of Lecture Notes in Computer Science.

[39] J. S. Bader, A. Chaudhuri, J. M. Rothberg, and J. Chant, "Gaining confidence in high-throughput protein interaction networks," *Nature Biotechnology*, vol. 22, no. 1, pp. 78–85, 2004.

[40] R. D. Luce and A. D. Perry, "A method of matrix analysis of group structure," *Psychometrika*, vol. 14, no. 2, pp. 95–116, 1949.

[41] R. D. Luce, "Connectivity and generalized cliques in sociometric group structure," *Psychometrika*, vol. 15, no. 2, pp. 169–190, 1950.

[42] R. J. Mokken, "Cliques, clubs and clans," *Quality and Quantity*, vol. 13, no. 2, pp. 161–173, 1979.

[43] J.-M. Bourjolly, G. Laporte, and G. Pesant, "An exact algorithm for the maximum k-club problem in an undirected graph," *European Journal Of Operational Research*, vol. 138, pp. 21–28, 2002.

[44] S. Butenko and O. Prokopyev, "On k-club and k-clique numbers in graphs," tech. rep., Texas A&M University, 2007.

[45] J. Marincek and B. Mohar, "On approximating the maximum diameter ratio of graphs," *Discrete Mathematics*, vol. 244, no. 1–3, pp. 323–330, 2002.

[46] Y. Asahiro, E. Miyano, and K. Samizo, "Approximating maximum diameter-bounded subgraphs," in *LATIN 2010: Theoretical Informatics* (A. Lpez-Ortiz, ed.), vol. 6034 of *Lecture Notes in Computer Science*, pp. 615–626, Springer Berlin / Heidelberg, 2010.

[47] R. G. Downey and M. R. Fellows, "Fixed-parameter tractability and completeness II: on completeness for W[1]," *Theoretical Computer Science*, vol. 141, no. 1-2, pp. 109–131, 1995.

[48] A. Schäfer, "Exact algorithms for s-club finding and related problems," Master's thesis, Diplomarbeit, Institut für Informatik, Friedrich-Schiller-Universität Jena, 2009.

[49] A. Schäfer, C. Komusiewicz, H. Moser, and R. Niedermeier, "Parameterized computational complexity of finding small-diameter subgraphs," *Optimization Letters*, pp. 1–9, 2011. DOI: 10.1007/s11590-011-0311-5.

[50] F. D. Carvalho and M. T. Almeida, "Upper bounds and heuristics for the 2-club problem," *European Journal of Operational Research*, vol. 210, no. 3, pp. 489–494, 2011.

[51] B. Balasundaram, *Graph Theoretic Generalizations Of Clique: Optimization and Extensions*. PhD thesis, Texas A&M University, College Station, Texas, USA, 2007.

[52] M. W. Padberg, "On the facial structure of set packing polyhedra," *Mathematical Programming*, vol. 5, no. 1, pp. 199–215, 1973.

[53] M. T. Almeida and F. D. Carvalho, "The k-club problem: new results for $k = 3$," Tech. Rep. CIO Working Paper 3/2008, CIO-Centro de Investigação Operacional, 2008.

[54] M. T. Almeida and F. D. Carvalho, "Integer models and upper bounds for the 3-club problem," *Networks*, 2012. DOI: 10.1002/net.21455.

[55] A. Veremyev and V. Boginski, "Identifying large robust network clusters via new compact formulations of maximum k-club problems," *European Journal of Operational Research*, vol. 218, no. 2, pp. 316–326, 2012.

[56] J.-M. Bourjolly, G. Laporte, and G. Pesant, "Heuristics for finding k-clubs in an undirected graph," *Computers & Operations Research*, vol. 27, pp. 559–569, 2000.

[57] J. Edachery, A. Sen, and F. J. Brandenburg, "Graph clustering using distance-k cliques," *Lecture Notes in Computer Science*, vol. 1731, pp. 98–106, 1999.

[58] S. Shahinpour and S. Butenko, "Algorithms for the maximum k-club problem in graphs," *Journal of Combinatorial Optimization*, 2012. DOI: 10.1007/s10878-012-9473-z.

[59] M. Yannakakis, "Node-and edge-deletion NP-complete problems," in *STOC '78: Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pp. 253–264, New York, NY: ACM Press, 1978.

[60] J. G. Oxley, *Matroid Theory*. Oxford, UK: Oxford University Press, 1992.

[61] R. Euler, M. Jünger, and G. Reinelt, "Generalizations of cliques, odd cycles and anticycles and their relation to independence system polyhedra," *Mathematics of Operations Research*, vol. 12, pp. 451–462, 1987.

[62] R. Carraghan and P. Pardalos, "An exact algorithm for the maximum clique problem," *Operations Research Letters*, vol. 9, pp. 375–382, 1990.

[63] P. R. J. Östergård, "A fast algorithm for the maximum clique problem," *Discrete Applied Mathematics*, vol. 120, pp. 197–207, 2002.

[64] S. T. McCormick, "Optimal approximation of sparse hessians and its equivalence to a graph coloring problem," *Mathematical Programming*, vol. 26, pp. 153–171, 1983.

[65] D. Brélaz, "New methods to color the vertices of a graph," *Communications of the ACM*, vol. 22, no. 4, pp. 251–256, 1979.

[66] G. Cornuéjols, *Combinatorial Optimization: Packing and Covering*. CBMS-NSF Regional Conference Series in Applied Mathematics, Philadelphia: SIAM, 2001.

[67] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, pp. 238–252, 1962.

[68] R. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *The Journal of Risk*, vol. 2, no. 3, pp. 21–41, 2000.

[69] R. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1443–1471, 2002.

[70] S. Uryasev, "Conditional value-at-risk: optimization algorithms and applications," *Proceedings of the IEEE/IAFE/INFORMS Conference on Computational Intelligence for Financial Engineering, 2000. (CIFEr)*, pp. 49–57, 2000.

[71] F. Andersson, H. Mausser, D. Rosen, and S. Uryasev, "Credit risk optimization with conditional value-at-risk criterion," *Math. Program.*, vol. 89, pp. 273–291, 2001.

[72] H. D. Sherali and B. M. P. Fraticelli, "A modification of benders decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse," *Journal of Global Optimization*, vol. 22, pp. 319–342, 2002.

[73] J. Pang and S. Leyffer, "On the global mininmization of the value-at-risk," *Optimization Methods and Software*, vol. 19, pp. 611–631, 2004.

[74] A. Künzi-Bay and J. Mayer, "Computational aspects of minimizing conditional value-at-risk," *Computational Management Science*, vol. 3, pp. 3–27, 2006.

[75] S. Ahmed, "Convexity and decomposition of mean-risk stochastic programs," *Mathematical Programming: Series A and B*, vol. 106(3), pp. 433–446, 2006.

[76] J. Gotoh and Y. Takano, "Newsvendor solutions via conditional value-at-risk minimization," *European Journal of Operation Research*, vol. 179, pp. 80–96, 2007.

[77] A. G. Quaranta and A. Zaffaroni, "Robust optimization of conditional value at risk and portfolio selection," *Journal of Banking and Finance*, vol. 32, pp. 2046–2056, 2008.

[78] C. I. Fábián, "Handling cvar objectives and constraints in two-stage stochastic models," *European Journal of Operational Research*, vol. 191, no. 3, pp. 888–911, 2008.

[79] C. I. Fábián, "Bundle-type methods for inexact data," *Central European Journal of Operations Research*, vol. 8, pp. 35–55, 2000.

[80] C. Lim, H. D. Sherali, and S. Uryasev, "Portfolio optimization by minimizing conditional value-at-risk via nondifferentiable optimization," *Comput Optim Appl*, vol. 46, pp. 391–415, 2010.

[81] P. Krokhmal, J. Palmquist, and S. Uryasev, "Portfolio optimization with conditional value-at-risk objective and constraints," *The Journal of Risk*, vol. 4(2), pp. 11–27, 2002.

[82] B. J. Manistre and G. H. Hancock, "Variance of the CTE estimators," *North American Actuarial Journal*, vol. 9, pp. 1–28, 2003.

[83] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov, "New variants of bundle methods," *Mathematical Programming*, vol. 69, pp. 111–147, 1995.

[84] P. Huang and D. Subramanian, "Interative estimation maximization for stochastic linear and convex programs with conditional value-at-risk constraints," *Mathmatics*, pp. 1–18, 2008.

[85] D. Subramanian and P. Huang, "An efficient decomposition algorithm for static, stochastic, linear and mixed-integer linear programs with conditional-value-at-risk constraints," Tech. Rep. RC24752, IBM Research Report, Feb 2009.

[86] F. M. Pajouh, E. Moradi, and B. Balasundaram, "Robust low-diameter cluster detection under probabilistic edge failures using conditional-value-at-risk," In preparation.

[87] B. Balasundaram and S. Butenko, "Network clustering," in *Analysis of Biological Networks* (B. H. Junker and F. Schreiber, eds.), pp. 113–138, New York: Wiley, 2008.

[88] S. Butenko and W. Wilhelm, "Clique-detection models in computational biochemistry and genomics," *European Journal of Operational Research*, vol. 173, pp. 1–17, 2006.

APPENDIX A

PROOF OF CLAIMS 1-3 IN SECTION 3.2

Proof. of Claim 1

Consider any two nodes v_1 and v_2 from D, there are four possibilities: (1) both belong to the same q-pendant; (2) v_1 belongs to connector pendant p_1 and v_2 belongs to a different connector pendant p_2; (3) v_1 belongs to connector pendant p_1 and v_2 belongs to an opposing pendant p_2; (4) v_1 belongs to opposing pendant p_1 and v_2 belongs to a different opposing pendant p_2.

Suppose k is even. In the first case, $d_{G[D]}(v_1, v_2) \leq q < k$. For any of the remaining three cases, there exists a connector pendant, say p_3, with its tail node adjacent to the tail nodes of p_1 and p_2. Now considering the path $p_1[v_1, tail(p_1)] - tail(p_3) - p_2[tail(p_2), v_2]$, we have $d_{G[D]}(v_1, v_2) \leq 2q + 2 = k$.

Suppose k is odd. As before in the first case, $d_{G[D]}(v_1, v_2) \leq q < k$. In the second case, since the penultimate node of p_1 is adjacent to the tail of p_2 we have the path $p_1[v_1, penultimate(p_1)] - p_2[tail(p_2), v_2]$, and $d_{G[D]}(v_1, v_2) \leq 2q = k - 1$. For the third case, there exists an opposing pendant p_3 whose tail is adjacent to the penultimate node of p_1 and the tail of p_2. Considering the path $p_1[v_1, penultimate(p_1)] - tail(p_3) - p_2[tail(p_2), v_2]$, we have $d_{G[D]}(v_1, v_2) \leq 2q + 1 = k$. In case (4), since the tail of all opposing pendants form a clique, we have the path $p_1[v_1, tail(p_1)] - p_2[tail(p_2), v_2]$ and $d_{G[D]}(v_1, v_2) \leq 2q + 1 = k$. ■

Proof. of Claim 2

a. Suppose k is even. In the graph $G[D \cup \{v_1\}]$, by construction, every path from v_1 to v_2 contains v_1's opposing pendant (of which v_2 is the head) and at least one of the q-pendants connected to v_1. The q-pendant connected to v_1 used on a shortest path could be v_1's connector pendant or some opposing pendant. Since the tail nodes of two opposing pendants are not adjacent, and the tail nodes of connector and opposing pendants of v_1 are not adjacent, the shortest path is through the tail node of some other connector pendant. The length of this shortest path is $q + 1 + 2 + q = 2q + 3 = k + 1$.

   Suppose k is odd. In the graph $G[D \cup \{v_1\}]$, by construction, every path from v_1 to v_2 contains v_1's opposing pendant (of which v_2 is the head) and, it must contain the path from v_1 to the penultimate node of v_1's connector pendant or contain some opposing pendant connected to v_1 from head to tail. The shortest paths through v_1's connector which uses the tail node of some other opposing pendant to reach the tail of v_1's opposing pendant are of length $1+(q-1)+2+q = 2q + 2 = k + 1$. The shortest path through some opposing pendant connected to v_1 is of length $1 + q + 1 + q = 2q + 2 = k + 1$. Recall that the tails of all opposing pendants form a clique when k is odd.

b. This claim is easily verified from the construction.

c. To prove necessity, suppose for a set $S \subseteq U$, $d_{G[D \cup S \cup \{v_1\}]}(v_1, v_2) \leq k$ and this set doesn't contain any supporter of v_1. According to Claim 2a, we know $d_{G[D \cup \{v_1\}]}(v_1, v_2) > k$ so the shortest path between v_1 and v_2 with length less than or equal to k must contain at least one element from $S \setminus \{v_1\}$. Suppose $v_3 \in S \setminus \{v_1\}$ is an internal node on some shortest path from v_1 to v_2 in $d_{G[D \cup S \cup \{v_1\}]}(v_1, v_2)$ and let p_3 denote v_3's connector pendant. Let p_1 denote v_1's opposing pendant. Since v_3 is not adjacent to v_2 (as it is not v_1's supporter), the shortest path between v_3 and v_2 for even k is $v_3 - p_3 - p_1[tail(p_1), v_2]$ with

length $2q + 2 = k$. For an odd k, this shortest path is $v_3 - p_3[head(p_3) - penultimate(p_3)] - p_1[tail(p_1), v_2]$ of length $2q + 1 = k$. Since the shortest path length between v_3 and v_2 is k, the shortest path between v_1 and v_2 that uses v_3 as internal node is at least $k + 1$ which contradicts $d_{G[D \cup S \cup \{v_1\}]}(v_1, v_2) \leq k$. So S should contain at least one supporter of v_1.

To establish sufficiency of this statement, let $v_3 \in S$ be a supporter of v_1. So v_3 is adjacent to v_1 and v_2. So $d_{G[D \cup S \cup \{v_1\}]}(v_1, v_2) \leq 2 \leq k$.

■

Proof. of Claim 3

To show necessity, suppose there exist literal nodes $v_1, v_2 \in U$ such that $d_{G[D \cup U]}(v_1, v_2) > k$ but $v_1 \neq \overline{v}_2$. v_1 and v_2 cannot belong to the same clause as they would form a clique in this case. Hence, v_1 and v_2 form a dcnn-pair, and by construction are connected by a k-chain contradicting $d_G(v_1, v_2) > k$. Thus, $v_1 = \overline{v_2}$.

To prove sufficiency, suppose there exist literal nodes $v_1, v_2 \in U$ such that $v_1 = \overline{v_2}$. So v_1 and v_2 belong to different clauses and there is no direct k-chain linking them. Suppose k is even. A shortest path from v_1 to v_2 traversing respective connector pendants as their tails are adjacent is of length $2q + 3 = k + 1$. The other type of shortest path partially traverses two k-chains from v_1 and v_2 to some v_3 and v_4 that respectively form dcnn-pairs up to their midpoints (which form a clique). In this case again the length is $2(\frac{k}{2}) + 1$. Now suppose k is odd, then one type of shortest path traverses v_1's connector pendant up to its penultimate node which is adjacent to the tail of v_2's connector pendant, again of length $q + 1 + q + 1 = k + 1$. The other type of shortest path traverses two k-chains from v_1 and v_2 up to the first midpoint which are all adjacent to the nucleus. These are also of length, $2\frac{(k-1)}{2} + 2 = k + 1$. ■

APPENDIX B

DETAILED NUMERICAL RESULTS OF THE COMPUTATIONAL EXPERIMENTS DESCRIBED IN SECTION 4.3

Table B.1: Average size of the best 2-club found by DC compared to BE, and their average running time (in seconds) on minimum VDV instances

n	Metric	LB	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
50	Best Obj	DC	3.70	4.90	6.70	11.00	15.20	21.40	31.60
		BE	3.70	4.90	6.70	11.00	15.30	22.10	32.40
	Time	DC	0.05	0.04	0.05	0.14	0.30	0.46	0.52
		BE	0.06	0.05	0.07	0.23	0.50	0.93	1.28
100	Best Obj	DC	6.10	8.80	11.60	18.90	26.10	63.70	95.00
		BE	6.10	8.80	11.60	18.90	26.60	66.20	95.00
	Time	DC	0.15	0.18	0.43	1.57	3.75	5.09	1.59
		BE	0.17	0.22	0.60	2.34	6.18	15.29	6.49
150	Best Obj	DC	6.90	11.00	16.80	26.80	37.70	133.10	149.40
		BE	6.90	11.00	16.80	26.80	39.70	133.60	149.40
	Time	DC	0.31	0.54	1.77	7.63	18.60	11.41	0.80
		BE	0.36	0.67	2.36	10.81	34.10	46.43	3.57
200	Best Obj	DC	8.20	12.70	20.70	33.60	58.50	195.20	200.00
		BE	8.20	12.70	20.70	33.60	68.70	195.20	200.00
	Time	DC	0.56	1.35	5.05	24.03	59.61	9.60	0.00
		BE	0.68	1.74	6.54	33.38	278.06	42.79	0.00

Table B.2: Average size of the best 2-club found by DC compared to BE, and their average running time (in seconds) on maximum VDV instances

n	Metric	LB	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
50	Best Obj	DC	4.30	5.20	7.40	11.60	15.80	22.90	33.70
		BE	4.30	5.20	7.40	11.60	16.30	23.00	34.20
	Time	DC	0.04	0.03	0.05	0.15	0.26	0.38	0.42
		BE	0.05	0.05	0.07	0.23	0.48	0.83	1.17
100	Best Obj	DC	5.90	10.10	12.70	22.00	34.50	70.00	87.90
		BE	5.90	10.10	12.70	22.00	36.00	70.50	87.90
	Time	DC	0.14	0.18	0.46	1.78	3.84	4.39	2.87
		BE	0.16	0.23	0.61	2.65	7.97	12.58	9.46
150	Best Obj	DC	8.30	11.80	19.50	31.70	75.10	122.40	143.70
		BE	8.30	11.80	19.50	31.70	77.00	122.70	143.70
	Time	DC	0.31	0.55	1.90	7.96	17.39	14.40	6.15
		BE	0.36	0.73	2.57	11.20	44.64	51.32	26.70
200	Best Obj	DC	9.20	14.30	23.40	39.10	126.20	177.20	196.40
		BE	9.20	14.30	23.40	39.10	128.80	177.40	196.40
	Time	DC	0.60	1.42	5.42	24.58	47.45	31.35	9.80
		BE	0.71	1.83	7.07	33.50	146.27	125.72	45.07

Table B.3: Average size of the best 3-club found by DC compared to BE, and their average running time (in seconds) on minimum VDV instances

n	Metric	LB	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
50	Best Obj	DC	4.60	6.60	10.90	28.90	47.30	50.00	50.00
		BE	4.60	6.60	10.90	29.10	47.30	50.00	50.00
	Time	DC	0.05	0.04	0.15	0.88	0.37	0.00	0.00
		BE	0.06	0.06	0.21	1.58	0.90	0.00	0.00
100	Best Obj	DC	8.40	13.60	22.80	93.90	100.00	100.00	100.00
		BE	8.40	13.80	25.50	93.90	100.00	100.00	100.00
	Time	DC	0.18	0.67	4.92	5.00	0.00	0.00	0.00
		BE	0.22	0.83	7.36	12.10	0.00	0.00	0.00
150	Best Obj	DC	11.30	18.50	41.30	149.90	150.00	150.00	150.00
		BE	11.30	19.60	44.30	149.90	150.00	150.00	150.00
	Time	DC	0.58	4.19	39.67	0.55	0.00	0.00	0.00
		BE	0.69	4.88	64.91	1.43	0.00	0.00	0.00
200	Best Obj	DC	13.50	20.70	89.70	200.00	200.00	200.00	200.00
		BE	13.50	23.40	94.50	200.00	200.00	200.00	200.00
	Time	DC	1.82	19.09	167.13	0.00	0.00	0.00	0.00
		BE	2.11	22.83	327.05	0.00	0.00	0.00	0.00

Table B.4: Average size of the best 3-club found by DC compared to BE, and their average running time (in seconds) on maximum VDV instances

n	Metric	LB	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
50	Best Obj	DC	4.40	7.80	11.20	31.50	43.60	48.30	50.00
		BE	4.40	7.80	11.20	32.00	43.60	48.30	50.00
	Time	DC	0.04	0.05	0.15	0.72	0.50	0.35	0.00
		BE	0.05	0.08	0.21	1.39	1.15	0.93	0.00
100	Best Obj	DC	7.90	15.80	27.40	93.00	99.70	100.00	100.00
		BE	7.90	15.80	29.80	93.10	99.70	100.00	100.00
	Time	DC	0.17	0.81	5.44	5.51	0.45	0.00	0.00
		BE	0.20	1.02	8.19	13.16	1.44	0.00	0.00
150	Best Obj	DC	12.00	20.20	59.20	148.20	150.00	150.00	150.00
		BE	12.00	20.80	61.70	148.20	150.00	150.00	150.00
	Time	DC	0.61	5.09	41.83	5.89	0.00	0.00	0.00
		BE	0.70	5.83	71.83	16.68	0.00	0.00	0.00
200	Best Obj	DC	14.30	24.90	118.70	199.70	200.00	200.00	200.00
		BE	14.30	29.40	122.30	199.70	200.00	200.00	200.00
	Time	DC	1.98	22.18	140.47	3.27	0.00	0.00	0.00
		BE	2.30	26.85	284.01	9.64	0.00	0.00	0.00

Table B.5: Average size of the best 2-club found, average running time (in seconds), and percentage optimality gap for each BB algorithm on 50-vertex instances

VDV	Metric	Algorithm	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
Min	Best Obj	DC/CO	3.70	4.90	6.70	11.00	15.50	23.50	34.60
		DC/KC	3.70	4.90	6.70	11.00	15.50	23.50	34.60
		BE/CO	3.70	4.90	6.70	11.00	15.50	23.50	34.60
		BE/KC	3.70	4.90	6.70	11.00	15.50	23.50	34.60
	Time	DC/CO	0.08	0.09	0.59	2.43	6.57	12.60	16.14
		DC/KC	0.06	0.05	0.08	0.98	16.62	61.96	63.63
		BE/CO	0.10	0.10	0.61	2.52	6.59	12.45	16.39
		BE/KC	0.06	0.07	0.11	1.05	16.14	60.99	60.82
	Gap	DC/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		DC/KC	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		BE/KC	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Max	Best Obj	DC/CO	4.30	5.20	7.40	11.60	16.30	24.10	34.60
		DC/KC	4.30	5.20	7.40	11.60	16.30	24.10	34.60
		BE/CO	4.30	5.20	7.40	11.60	16.30	24.10	34.60
		BE/KC	4.30	5.20	7.40	11.60	16.30	24.10	34.60
	Time	DC/CO	0.07	0.10	0.17	2.98	6.06	9.18	6.87
		DC/KC	0.05	0.06	0.08	1.60	12.42	20.34	12.82
		BE/CO	0.08	0.11	0.19	3.12	6.10	9.41	7.33
		BE/KC	0.06	0.07	0.10	1.75	12.15	20.39	12.88
	Gap	DC/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		DC/KC	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		BE/KC	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table B.6: Average size of the best 2-club found, average running time (in seconds), and percentage optimality gap for each BB algorithm on 100-vertex instances

VDV	Metric	Algorithm	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
Min	Best Obj	DC/CO	6.10	8.80	11.60	18.90	29.30	71.20	95.90
		DC/KC	6.10	8.80	11.60	18.90	26.10	63.70	95.90
		BE/CO	6.10	8.80	11.60	18.90	29.30	71.20	95.90
		BE/KC	6.10	8.80	11.60	18.90	26.60	66.20	95.90
	Time	DC/CO	0.33	0.77	14.64	57.21	2005.57	1083.22	67.34
		DC/KC	0.20	0.27	0.78	1302.60	4622.95	4763.65	42.90
		BE/CO	0.35	0.83	14.64	59.09	1969.87	1083.24	72.05
		BE/KC	0.22	0.38	0.94	1290.13	4608.93	4746.29	47.53
	Gap	DC/CO	0.00	0.00	0.00	0.00	1.67	0.00	0.00
		DC/KC	0.00	0.00	0.00	1.67	73.90	35.09	0.00
		BE/CO	0.00	0.00	0.00	0.00	1.67	0.00	0.00
		BE/KC	0.00	0.00	0.00	7.65	73.40	32.50	0.00
Max	Best Obj	DC/CO	5.90	10.10	12.70	22.10	40.70	72.50	88.20
		DC/KC	5.90	10.10	12.70	22.00	34.50	70.10	88.20
		BE/CO	5.90	10.10	12.70	22.10	40.70	72.50	88.20
		BE/KC	5.90	10.10	12.70	22.00	36.00	70.60	88.20
	Time	DC/CO	0.31	0.85	14.30	69.06	746.99	213.78	85.88
		DC/KC	0.19	0.29	0.81	2327.69	3706.72	4073.51	491.32
		BE/CO	0.34	0.88	14.43	71.30	736.52	217.26	92.13
		BE/KC	0.20	0.37	0.96	2303.39	3723.32	4056.79	472.59
	Gap	DC/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		DC/KC	0.00	0.00	0.00	15.99	64.69	25.04	0.00
		BE/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		BE/KC	0.00	0.00	0.00	22.58	62.95	22.73	0.00

Table B.7: Average size of the best 2-club found, average running time (in seconds), and percentage optimality gap for each BB algorithm on 150-vertex instances

VDV	Metric	Algorithm	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
Min	Best Obj	DC/CO	6.90	11.00	16.80	26.80	37.70	136.70	149.40
		DC/KC	6.90	11.00	16.80	26.80	37.70	133.70	149.40
		BE/CO	6.90	11.00	16.80	26.80	39.70	136.70	149.40
		BE/KC	6.90	11.00	16.80	26.80	39.70	134.20	149.40
	Time	DC/CO	0.97	6.70	89.72	1231.60	3604.96	866.54	82.49
		DC/KC	0.48	0.86	4.06	4159.46	4661.69	7493.11	10.49
		BE/CO	1.00	6.84	90.44	1248.68	3606.13	894.55	85.42
		BE/KC	0.51	0.99	4.64	4167.56	4620.92	7464.31	13.27
	Gap	DC/CO	0.00	0.00	0.00	0.00	49.01	0.08	0.00
		DC/KC	0.00	0.00	0.00	82.13	74.87	9.10	0.00
		BE/CO	0.00	0.00	0.00	0.00	46.46	0.08	0.00
		BE/KC	0.00	0.00	0.00	82.13	73.53	8.76	0.00
Max	Best Obj	DC/CO	8.30	11.80	19.50	31.70	75.40	123.60	143.80
		DC/KC	8.30	11.80	19.50	31.70	75.10	122.40	143.80
		BE/CO	8.30	11.80	19.50	31.70	77.30	123.60	143.80
		BE/KC	8.30	11.80	19.50	31.70	77.00	122.70	143.80
	Time	DC/CO	0.97	8.91	80.94	1358.68	3335.84	567.21	281.31
		DC/KC	0.48	0.90	4.26	4494.86	4284.58	6847.17	647.13
		BE/CO	1.00	9.20	80.71	1370.06	3341.99	589.15	302.28
		BE/KC	0.50	1.06	4.94	4486.42	4250.77	6841.50	668.37
	Gap	DC/CO	0.00	0.00	0.00	0.00	10.40	0.00	0.00
		DC/KC	0.00	0.00	0.00	78.87	49.93	18.34	0.80
		BE/CO	0.00	0.00	0.00	0.00	8.29	0.00	0.00
		BE/KC	0.00	0.00	0.00	78.87	48.67	18.14	0.80

Table B.8: Average size of the best 2-club found, average running time (in seconds), and percentage optimality gap for each BB algorithm on 200-vertex instances

VDV	Metric	Algorithm	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
Min	Best Obj	DC/CO	8.20	12.70	20.70	33.60	58.50	195.50	200.00
		DC/KC	8.20	12.70	20.70	33.60	58.50	195.50	200.00
		BE/CO	8.20	12.70	20.70	33.60	68.70	195.50	200.00
		BE/KC	8.20	12.70	20.70	33.60	68.70	195.50	200.00
	Time	DC/CO	2.54	71.62	298.98	3604.30	3613.16	880.00	0.00
		DC/KC	0.95	2.12	16.71	4554.49	5276.76	331.06	0.00
		BE/CO	2.65	71.93	300.04	3605.00	3618.83	913.00	0.00
		BE/KC	1.01	2.53	18.18	4548.06	4542.85	361.98	0.00
	Gap	DC/CO	0.00	0.00	0.00	37.78	56.14	0.00	0.00
		DC/KC	0.00	0.00	0.00	83.20	70.75	0.00	0.00
		BE/CO	0.00	0.00	0.00	37.66	48.78	0.00	0.00
		BE/KC	0.00	0.00	0.00	83.20	65.65	0.00	0.00
Max	Best Obj	DC/CO	9.20	14.30	23.40	39.10	126.20	178.20	196.40
		DC/KC	9.20	14.30	23.40	39.10	126.20	177.20	196.40
		BE/CO	9.20	14.30	23.40	39.10	128.80	178.20	196.40
		BE/KC	9.20	14.30	23.40	39.10	128.80	177.40	196.40
	Time	DC/CO	2.61	48.14	299.68	3608.07	3403.16	1664.25	777.07
		DC/KC	0.96	2.24	42.41	4884.64	4377.61	5603.51	116.33
		BE/CO	2.68	48.07	300.83	3604.49	3411.23	1747.34	813.20
		BE/KC	1.03	2.67	44.07	4878.12	4395.17	5606.31	151.69
	Gap	DC/CO	0.00	0.00	0.00	29.77	8.00	0.06	0.00
		DC/KC	0.00	0.00	0.00	80.45	36.90	10.35	0.00
		BE/CO	0.00	0.00	0.00	29.34	6.05	0.06	0.00
		BE/KC	0.00	0.00	0.00	80.45	35.60	10.25	0.00

Table B.9: Average size of the best 3-club found, average running time (in seconds), and percentage optimality gap for each BB algorithm on 50-vertex instances

VDV	Metric	Algorithm	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
Min	Best Obj	DC/CO	4.60	6.60	11.10	30.20	47.40	50.00	50.00
		DC/KC	4.60	6.60	11.10	30.20	47.40	50.00	50.00
		BE/CO	4.60	6.60	11.10	30.20	47.40	50.00	50.00
		BE/KC	4.60	6.60	11.10	30.20	47.40	50.00	50.00
	Time	DC/CO	0.08	0.11	1.69	7.80	3.71	0.00	0.00
		DC/KC	0.06	0.07	1.77	15.79	1.50	0.00	0.00
		BE/CO	0.09	0.13	1.75	8.42	4.18	0.00	0.00
		BE/KC	0.07	0.09	1.82	16.46	2.00	0.00	0.00
	Gap	DC/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		DC/KC	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		BE/KC	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Max	Best Obj	DC/CO	4.40	7.80	12.00	32.00	43.60	48.30	50.00
		DC/KC	4.40	7.80	12.00	32.00	43.60	48.30	50.00
		BE/CO	4.40	7.80	12.00	32.00	43.60	48.30	50.00
		BE/KC	4.40	7.80	12.00	32.00	43.60	48.30	50.00
	Time	DC/CO	0.08	0.15	1.39	6.20	3.39	3.94	0.00
		DC/KC	0.05	0.08	1.26	6.34	1.18	1.28	0.00
		BE/CO	0.09	0.18	1.46	5.72	4.13	4.56	0.00
		BE/KC	0.06	0.11	1.30	4.66	1.84	1.84	0.00
	Gap	DC/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		DC/KC	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		BE/KC	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table B.10: Average size of the best 3-club found, average running time (in seconds), and percentage optimality gap for each BB algorithm on 100-vertex instances

VDV	Metric	Algorithm	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
Min	Best Obj	DC/CO	8.40	14.60	28.00	94.00	100.00	100.00	100.00
		DC/KC	8.40	14.60	25.00	94.00	100.00	100.00	100.00
		BE/CO	8.40	14.60	28.00	94.00	100.00	100.00	100.00
		BE/KC	8.40	14.60	26.50	94.00	100.00	100.00	100.00
	Time	DC/CO	0.48	15.45	107.91	70.40	0.00	0.00	0.00
		DC/KC	0.27	11.52	2800.19	59.08	0.00	0.00	0.00
		BE/CO	0.52	13.58	109.81	77.18	0.00	0.00	0.00
		BE/KC	0.31	7.23	2787.81	65.86	0.00	0.00	0.00
	Gap	DC/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		DC/KC	0.00	0.00	43.39	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		BE/KC	0.00	0.00	41.69	0.00	0.00	0.00	0.00
Max	Best Obj	DC/CO	7.90	15.80	31.30	93.20	99.70	100.00	100.00
		DC/KC	7.90	15.80	28.60	93.20	99.70	100.00	100.00
		BE/CO	7.90	15.80	31.30	93.20	99.70	100.00	100.00
		BE/KC	7.90	15.80	30.40	93.20	99.70	100.00	100.00
	Time	DC/CO	0.43	14.13	119.51	69.16	17.80	0.00	0.00
		DC/KC	0.25	9.19	3712.98	43.31	3.37	0.00	0.00
		BE/CO	0.48	14.06	118.65	76.62	18.72	0.00	0.00
		BE/KC	0.29	9.36	3689.42	50.63	4.40	0.00	0.00
	Gap	DC/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		DC/KC	0.00	0.00	43.08	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		BE/KC	0.00	0.00	41.19	0.00	0.00	0.00	0.00

Table B.11: Average size of the best 3-club found, average running time (in seconds), and percentage optimality gap for each BB algorithm on 150-vertex instances

VDV	Metric	Algorithm	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
Min	Best Obj	DC/CO	11.70	20.50	41.30	149.90	150.00	150.00	150.00
		DC/KC	11.70	20.50	41.30	149.90	150.00	150.00	150.00
		BE/CO	11.70	20.50	44.30	149.90	150.00	150.00	150.00
		BE/KC	11.70	20.50	44.30	149.90	150.00	150.00	150.00
	Time	DC/CO	7.02	110.76	3603.82	28.23	0.00	0.00	0.00
		DC/KC	4.66	416.51	4901.93	4.01	0.00	0.00	0.00
		BE/CO	7.16	108.82	3602.51	29.15	0.00	0.00	0.00
		BE/KC	4.82	248.94	4851.43	4.95	0.00	0.00	0.00
	Gap	DC/CO	0.00	0.00	30.20	0.00	0.00	0.00	0.00
		DC/KC	0.00	0.00	72.47	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	25.28	0.00	0.00	0.00	0.00
		BE/KC	0.00	0.00	70.47	0.00	0.00	0.00	0.00
Max	Best Obj	DC/CO	12.00	22.90	61.60	148.20	150.00	150.00	150.00
		DC/KC	12.00	21.70	59.20	148.20	150.00	150.00	150.00
		BE/CO	12.00	22.90	63.50	148.20	150.00	150.00	150.00
		BE/KC	12.00	21.90	61.70	148.20	150.00	150.00	150.00
	Time	DC/CO	8.55	117.32	3119.97	254.46	0.00	0.00	0.00
		DC/KC	0.90	1450.18	4619.79	36.90	0.00	0.00	0.00
		BE/CO	8.30	116.16	3129.74	266.00	0.00	0.00	0.00
		BE/KC	1.02	1368.11	4555.77	47.56	0.00	0.00	0.00
	Gap	DC/CO	0.00	0.00	12.97	0.00	0.00	0.00	0.00
		DC/KC	0.00	7.39	60.53	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	10.15	0.00	0.00	0.00	0.00
		BE/KC	0.00	25.16	58.87	0.00	0.00	0.00	0.00

Table B.12: Average size of the best 3-club found, average running time (in seconds), and percentage optimality gap for each BB algorithm on 200-vertex instances

VDV	Metric	Algorithm	Edge Density						
			0.0125	0.025	0.05	0.1	0.15	0.2	0.25
Min	Best Obj	DC/CO	13.60	25.40	89.70	200.00	200.00	200.00	200.00
		DC/KC	13.60	20.70	89.70	200.00	200.00	200.00	200.00
		BE/CO	13.60	25.40	94.50	200.00	200.00	200.00	200.00
		BE/KC	13.60	23.40	94.50	200.00	200.00	200.00	200.00
	Time	DC/CO	87.90	442.32	3616.94	0.00	0.00	0.00	0.00
		DC/KC	17.24	3684.29	4108.80	0.00	0.00	0.00	0.00
		BE/CO	88.28	431.66	3619.46	0.00	0.00	0.00	0.00
		BE/KC	17.98	3702.84	4160.96	0.00	0.00	0.00	0.00
	Gap	DC/CO	0.00	0.00	31.64	0.00	0.00	0.00	0.00
		DC/KC	0.00	89.30	55.15	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	28.13	0.00	0.00	0.00	0.00
		BE/KC	0.00	87.92	52.75	0.00	0.00	0.00	0.00
Max	Best Obj	DC/CO	14.80	31.90	119.10	199.70	200.00	200.00	200.00
		DC/KC	14.80	24.90	118.70	199.70	200.00	200.00	200.00
		BE/CO	14.80	31.90	122.50	199.70	200.00	200.00	200.00
		BE/KC	14.80	29.40	122.30	199.70	200.00	200.00	200.00
	Time	DC/CO	91.39	477.40	3275.85	262.39	0.00	0.00	0.00
		DC/KC	63.03	3760.07	4523.58	32.50	0.00	0.00	0.00
		BE/CO	91.75	464.38	3304.12	268.81	0.00	0.00	0.00
		BE/KC	62.81	3757.28	4526.92	38.81	0.00	0.00	0.00
	Gap	DC/CO	0.00	0.00	12.29	0.00	0.00	0.00	0.00
		DC/KC	0.00	87.39	40.65	0.00	0.00	0.00	0.00
		BE/CO	0.00	0.00	9.64	0.00	0.00	0.00	0.00
		BE/KC	0.00	85.11	38.85	0.00	0.00	0.00	0.00

APPENDIX C

PROOF OF CLAIMS 4-10 IN SECTION 5.3

Proof. of Claim 4

Suppose there exists $j \in V$ such that $\bar{x}_j > \sum_{k \in N_G(j)} \bar{x}_k$. We claim all the elements in the j^{th} column of A_1 and A_2 are equal to 1 and 0 respectively. Since $\bar{x}_j > 0$, all elements in the j^{th} column of A_2 are equal to 0. Now suppose there exists a row in A_1 in which the element in the j^{th} column is not 1 and let C be the corresponding I2DS for this row. For any set $D \in \Gamma((C \cup \{j\}) \setminus N_G(j))$, set $C' = (C \cup \{j\} \cup D) \setminus N_G(j)$ is an I2DS for which $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i + \bar{x}_j - \sum_{k \in N_G(j)} \bar{x}_k > 1$ which contradicts with $\bar{x} \in P_G$. This proves the validity of our claim. According to this observation, the vector e_j is a solution for system 5.4 which contradicts with $\bar{x}$ being the unique solution of this system. ■

Proof. of Claim 5

Otherwise for a set $D \in \Gamma((C \setminus (N_G(a) \cup N_G(b))) \cup \{a\})$, set $C' = (C \setminus (N_G(a) \cup N_G(b))) \cup \{a\} \cup D$ is an I2DS in G for which $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i + \bar{x}_a > 1$. This contradicts with $\bar{x} \in P_G$. ■

Proof. of Claim 6

Suppose there exists $C \in \tau(S_1)$ for which the coefficient of x_a in the I2DS constraint corresponding to set C is negative. Then, there exists a vertex $m \in (C \cap N_G(a))$ such that $\bar{x}_m = 0$. For any set $D \in \Gamma(C \setminus \{m\})$, set $C' = (C \setminus \{m\}) \cup D$ is an I2DS in G for which $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i + \bar{x}_a > 1$. This contradicts with $\bar{x} \in P_G$. ■

Proof. of Claim 7

Suppose for a set $C \in \tau(S_1)$, there exists a vertex $m \in (N_G(b) \cap C)$ such that $\bar{x}_m = 0$. So $b \notin C$ and by Claim 5, the coefficient of x_b in the I2DS constraint corresponding to set C is negative. For any set $D \in \Gamma(C \setminus \{m\})$, set $C' = (C \setminus \{m\}) \cup D$ is an I2DS in G for which $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i + \bar{x}_b > 1$. This contradicts with $\bar{x} \in P_G$. ■

Proof. of Claim 8

Suppose for any $C \in \tau(S_1)$, the coefficient of x_c in the I2DS constraint corresponding to set C is nonnegative. Then by Claims 5, 6 and 7, the vector $\hat{e} = \sum_{i \in (\{b\} \cup N_G(b)) \cap V'} e_i$ will solve system 5.4 which is a contradiction with the uniqueness of $\bar{x}$. ■

Proof. of Claim 9

If for all $C \in \tau(S_1)$, we have $c \notin C$ then by Claims 5, 6 and 7, the vector $\hat{e} = \sum_{i \in (\{b\} \cup N_G(b)) \setminus \{c\}) \cap V'} e_i$ is a solution for system 5.4 which contradicts with $\bar{x}$ being the unique solution of this system. ■

Proof. of Claim 10

Consider a vertex $m \in (N_G(c) \cap V') \setminus \{d\}$. If m is a leaf vertex of $\bar{G}$ then by Inequality 5.5, we have $\bar{x}_m \leq \bar{x}_c$. If not, then all vertices in $(N_G(m) \cap V') \setminus \{c\}$ are the leaf vertices of $\bar{G}$ because otherwise p is not the longest path in this graph. Suppose $\bar{x}_m > \bar{x}_c$, then for all $C \in \tau(S_1)$, we have $c \notin C$. Otherwise, consider a $C \in \tau(S_1)$ which contains c. Define set $W = N_G(m) \cup (\cup_{i \in N_G(m) \setminus \{c\}} N_G(i))$. For any set $D \in \Gamma((C \setminus W) \cup \{m\})$, set $C' = (C \setminus W) \cup \{m\} \cup D$ is an I2DS in G for which $\sum_{i \in C'} \bar{x}_i - \sum_{i \in V \setminus C'} (|N(i) \cap C'| - 1)^+ \bar{x}_i \geq \sum_{i \in C} \bar{x}_i - \sum_{i \in V \setminus C} (|N(i) \cap C| - 1)^+ \bar{x}_i - \bar{x}_c + \bar{x}_m > 1$. This contradicts with $\bar{x} \in P_G$. So by Claim 9, we have $\bar{x}_m \leq \bar{x}_c$. ■

VITA

Foad Mahdavi Pajouh

Candidate for the Degree of

Doctor of Philosophy

Dissertation: POLYHEDRAL COMBINATORICS, COMPLEXITY & ALGORITHMS FOR k-CLUBS IN GRAPHS

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Hamedan, Iran on July 24, 1981.

Education:

Received the B.S. degree from Sharif University of Technology, Tehran, Iran, 2004, in Industrial Engineering
Received the M.S. degree from Tarbiat Modares University, Tehran, Iran, 2006, in Industrial Engineering
Completed the requirements for the degree of Doctor of Philosophy with a major in Industrial Engineering and Management at Oklahoma State University in August, 2012.

Experience:

Mr. Mahdavi Pajouh has research interests in theoretical, computational and applied optimization. He has research experience in theoretical and computational optimization, applied optimization and stochastic modeling. He has worked as a graduate research assistant in the School of Industrial Engineering and Management at Oklahoma State University (OSU). He has teaching experience as an instructor and has taught two courses for the School of Industrial Engineering and Management during Fall 2011 and Spring 2012 semesters. He worked as an industrial engineering expert at Iran Khodro Company (IKCO) which is a large automobile manufacturer in the middle east. He also collaborated with Hamedan management and planning organization as a system analyst. He will join the department of Industrial and Systems Engineering at University of Florida as an adjunct assistant professor in August 2012.

Name: Foad Mahdavi Pajouh Date of Degree: July, 2012

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: POLYHEDRAL COMBINATORICS, COMPLEXITY & ALGORITHMS FOR k-CLUBS IN GRAPHS

Pages in Study: 114 Candidate for the Degree of Doctor of Philosophy

Major Field: Industrial Engineering and Management

A k-club is a distance-based graph-theoretic generalization of clique, originally introduced to model cohesive subgroups in social network analysis. The k-clubs represent low diameter clusters in graphs and are suitable for various graph-based data mining applications. Unlike cliques, the k-club model is nonhereditary, meaning every subset of a k-club is not necessarily a k-club. This imposes significant challenges in developing theory and algorithms for optimization problems associated with k-clubs.

We settle an open problem establishing the intractability of testing inclusion-wise maximality of k-clubs for fixed $k \geq 2$. This result is in contrast to polynomial-time verifiability of maximal cliques, and is a direct consequence of k-clubs' nonhereditary nature. A class of graphs for which this problem is polynomial-time solvable is also identified. We propose a distance coloring based upper-bounding scheme and a bounded enumeration based lower-bounding routine and employ them in a combinatorial branch-and-bound algorithm for finding a maximum k-club. Computational results on graphs with up to 200 vertices are also provided.

The 2-club polytope of a graph is studied and a new family of facet inducing inequalities for this polytope is discovered. This family of facets strictly contains all known nontrivial facets of the 2-club polytope as special cases, and identifies previously unknown facets of this polytope. The separation complexity of these newly discovered facets is proved to be NP-complete and it is shown that the 2-club polytope of trees can be completely described by the collection of these facets along with the nonnegativity constraints.

We also studied the maximum 2-club problem under uncertainty. Given a random graph subject to probabilistic edge failures, we are interested in finding a large "risk-averse" 2-club. Here, risk-aversion is achieved via modeling the loss in 2-club property due to edge failures, as random loss, which is a function of the decision variables and uncertain parameters. Conditional Value-at-Risk (CVaR) is used as a quantitative measure of risk that is constrained in the model. Benders' decomposition scheme is utilized to develop a new decomposition algorithm for solving the CVaR constrained maximum 2-club problem. A preliminary experiment is also conducted to compare the computational performance of the developed algorithm with our extension of an existing algorithm from the literature.

ADVISOR'S APPROVAL: Dr. Balabhaskar Balasundaram