CHARACTERIZING STRUCTURALLY COHESIVE CLUSTERS IN NETWORKS: THEORY AND ALGORITHMS

A Dissertation

by

CHITRA BALASUBRAMANIAM

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Sergiy Butenko
Committee Members,	Anxiao(Andrew) Jiang
	Lewis Ntaimo
	Kiavash Kianfar
Head of Department,	Cesar O. Malave

August 2015

Major Subject: Industrial Engineering

ABSTRACT

This dissertation aims at developing generalized network models and solution approaches for studying cluster detection problems that typically arise in networks. More specifically, we consider graph theoretic relaxations of clique as models for characterizing structurally cohesive and robust subgroups, developing strong upper bounds for the maxiumum clique problem, and present a new relaxation that is useful in clustering applications.

We consider the clique relaxation models of k-block, and k-robust 2-club for describing cohesive clusters that are reliable and robust to disruptions, and introduce a new relaxation called s-stable cluster, for modeling stable clusters. First, we identify the structural properties associated with the models, and investigate the computational complexity of these problems. Next, we develop mathematical programming techniques for the optimization problems introduced, and apply them in presenting effective solution approaches to the problems.

We present integer programming formulations for the optimization problems of interest, and provide a detailed study of the associated polytopes. Particularly, we develop valid inequalities and identify different classes of facets for the polytopes. Exact solution approaches developed for solving the problems include simple branch and bound, branch and cut, and combinatorial branch and bound algorithms. In addition, we introduce many preprocessing techniques and heuristics to enhance their performance. The presented algorithms are tested computationally on a number of graph instances, that include social networks and random graphs, to study the capability of the proposed solution methods.

As a fitting conclusion to this work, we propose new techniques to get easily computable and strong upper bounds for the maximum clique problem. We investigate k-core and its stronger variant k-core/2-club in this light, and present minimization problems to get an upper bound on the maximization problems. Simple linear programming relaxations are developed and strengthened by valid inequalities, which are then compared with some standard relaxations from the literature. We present a detailed study of our computational results on a number of benchmark instances to test the effectiveness of our technique for getting good upper bounds.

DEDICATION

To my parents and sister.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Dr. Sergiy Butenko for his patience, guidance, support and encouragement that he gave me during the last five years. You have been a true mentor in every sense of the word. I am truly grateful to you for providing me with various opportunities to improve myself as a researcher, and for being such a great person to work with.

I am very grateful to my committee members Drs. Anxiao Jiang, Kiavash Kianfar, and Lewis Ntaimo for their time, patience, excellent feedback and support. I would also like to thank Dr.Alaa Elwany for his timely help and support.

I would like to thank the faculty of the department especially, Drs. Amarnath Bannerjee, George Kemble Bennett, Richard Feldman, Andrew Johnson, Georgia-Ann Klutke, and Gautam Natarajan, for providing me with several learning opportunities.

Many thanks are due to the staff members in the department, especially Judy Meeks, Erin Roady, and Jamie Vykukal for all their help and support. I am also grateful to the graduate student community in the department for their friendship and support.

I would like to thank my friends Kavitha Lakshmanan, Saranya Parthasarathy, Mukulika Ghosh, Mahima Agumbe Suresh, Pranavi Yerasi, Radhika Siravara, Neeharika Marukala, Reena Panda, Samyukta Sethuraman, Priyanka Chaudhary, Pushpita Dutta, and Jyotirmoy Dalal for making my life in College Station very enjoyable and cherishable.

Finally, my love for my parents and sister is beyond any expression. Their love

and support have made me the person that I am today, and I am very grateful to them for constantly encouraging me to achieve my dream. They have been a pillar of strength throughout and I am most blessed to have them as my family.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

A graph, defined with a vertex set, and an edge set that represents links between pairs of vertices, is the most popular and useful way of modeling many systems. For example, complex systems like social networks, arising in various applications are often modeled as a graph with entities as vertices, and there is an edge between the vertices if they have some pairwise interaction between them. Such a representation helps to gain very useful and insightful information about the network, and will aid in understanding the relationships between the components of the underlying system, and detect interesting structural properties it exhibits. For instance, when modeling *social networks* as graphs, some of the most useful information that can be retrieved include the number of jumps required for any two entities to connect and detecting the most well connected subgroup in the network.

In any system, the overall data representing it is non-homogeneous, and hence, graph theoretic tools have played a significant role in understanding and interpreting this data. Graph clustering is a very popular field of study that helps in determining the underlying structure in any network [55]. The resulting structure, called a graph cluster, is an association of vertices that share similar traits with one another. Over the years, graph clustering has become a very popular field of study, and a large number of clustering techniques and algorithms [55] have been proposed in literature, which are very useful in many data mining applications. One of the key properties that is very significant in determining a cluster is called a cluster measure, which simply put is the characteristic trait that the vertices share in the cluster. Here are some examples of measures used for characterizing clusters.

Protein interaction networks: When modeled as graphs, vertices represent proteins, and there is an edge between two vertices, if the corresponding proteins interact with each other. Protein clusters, where two different proteins interact either directly or through another protein, can be modeled using distance measure [6].

Social networks: A key property of social network is cohesiveness and a cohesive subgroup is one, where each entity has a large number of neighbors, that is familiarity, signifying some kind of similarity between them. Hence, it is easy to interpret a given social network by detecting large cohesive subgroups using familiarity as a measure [5].

This dissertation concentrates on three such clustering measures, namely connectivity, reachability for finding closely-knit subgroups, and stability for finding a stable cluster. In this work, models that can be used for finding clusters with these properties are discussed in detail. Next, we discuss the motivation behind choosing these measures of clustering, and their significance.

1.1 Motivation

Many application areas, like communication networks, information networks, energy networks etc., require that any two entities in the system have good and undisrupted connectivity, which calls for cohesive properties in the network. In addition, they also require that failure of one or more entities does not affect network functioning, thus ensuring network robustness. Hence, such networks require the existence of multiple disjoint paths between entities, so that failure of one or more paths does not affect the passage of information across the network. Moreover, to ensure a swift passage of information across the network, these paths must of short length [26]. Given these properties, one can find the level of cohesiveness and robustness of a network by detecting an underlying cluster that satisfies them.

A clique is a subset of pairwise adjacent vertices that induces a complete graph and is a tightly-knit structure, and has been widely used for identifying cohesive clusters in networks. A clique ensures that there are maximum number of disjoint paths of the smallest length possible between entities. However, requiring links to be present between every pair of vertices in a cluster might not always be a reasonable assumption for practical applications. Hence, a part of this dissertation studies models that can used for identifying clusters that are robust and cohesive, but are not as restrictive.

In addition to the above mentioned properties of cohesiveness and robustness, cliques have other structural properties that are desirable to be present in any network, namely, familiarity between entities, density of the network, and reachability among entities. One can use these properties to identify clusters, and hence, clique has been an ideal model for many clustering applications as it possess all of them. However, to overcome the restricted nature of clique, over the years, a number of models called *'clique relaxations'* have been introduced to model clusters satisfying one or more of them [52]. These models were obtained with the aim of characterizing clusters based on the chosen property, and include s-plex [57], a familiarity based relaxation of clique, s-clubs and s-cliques [42], reachability based relaxations, γ-quasi-clique [2], a density based relaxation, and k-blocks [40, 52], a robustness based relaxation. Such models aid in characterizing and interpreting clusters based on their structural properties. As a continuation of modeling clusters with different measures, we introduce a new model for characterizing clusters that holds a restriction on the maximum number of entities acting independently of each other. It is easy to see that a clique has only one such entity, and hence one can model clusters by relaxing this property of clique.

Different clustering models including the ones described above were originally introduced not just due to the restrictive (ideal) nature of clique in modeling clusters, but also due to the intractability of finding very large cliques in networks. Owing to the popularity that cliques enjoy in many applications, efforts have been made towards this end, that include a number of exact and enumerative algorithms for solving this problem. This has further led to developing good lower and upper bounds for the problem, which can then be used as a starting point for the exact algorithms. As a fitting conclusion to this dissertation, we propose a new and simple model for getting a tight upper bound on the largest clique found in a given graph.

1.2 Preliminaries

Before we describe the contributions made in this research work, in this section, some essential background required for this dissertation will be reviewed. In particular, some definitions, notations and terminology that will be used throughout this dissertation will be reviewed.

The reader is referred to Diestel [15] for some basic definition of graph-theoretic concepts ommitted here. In this work, we consider a simple graph $G = (V, E)$ with vertex set V, and edge set E. For any subset of vertices $S \subseteq V$, $G[S] = (S, E \cap S \times S)$ denotes the graph induced by S, and the complement graph of G is denoted by $\bar{G} = (V, \bar{E})$ with $\bar{E} = \{(u, v) \notin E : \forall u, v \in V\}$. The neighborhood of a vertex $v \in V$ is the number of vertices adjacent to v, and is denoted as $N(v) = \{u : (u, v) \in E\}$, and $N[v]$ denotes the closed neighborhood of v in G, that is $N[v] = N(v) \cup \{v\}$. The degree of a vertex $v \in V$ is the number of vertices adjacent to it, namely $|N(v)|$ and is denoted by $deg_G(v)$. The minimum and the maximum degrees of a graph G is denoted by $\delta(G)$ and $\Delta(G)$, respectively. The length of the shortest path between $i, j \in G$ is denoted by $d_G(i, j)$, and the diameter of a graph G is denoted as $diam(G) = \max_{i,j \in V} d_G(i, j)$.

Definition 1 *A subset $C \subseteq V$ is called a clique, if $G[C]$ induces a complete subgraph.*

Definition 2 *A subset $I \subseteq V$ is called an independent set or a stable set, if any pair of vertices in I is nonadjacent, that is, $\bar{G}[I]$ induces a complete subgraph.*

The *maximum clique problem* asks to find a clique of maximum cardinality (also called the clique number), in a given graph G, and the cardinality of the maximum clique is denoted by $\omega(G)$. Similarly, an independent set of maximum cardinality in G is denoted by $\alpha(G)$, and by the definition above, $\omega(G) = \alpha(\bar{G})$.

Definition 3 *A subset $S \subseteq V$ is called a k-core if $\delta(G[S]) \geq k$.*

Definition 4 *A graph G is connected, if there exists a path between every pair of vertices.*

Definition 5 *A vertex cut, also called a separating set of a connected graph G, is a subset of vertices whose removal disconnects G.*

Definition 6 *The* vertex connectivity *of G, denoted by $\kappa(G)$, is the greatest integer k such that G is k-connected.*

For a complete graph K_n on n vertices, we have $\kappa(K_n) = n - 1$

Definition 7 *Two vertices i, j in G are called vertex-disjoint if there are no common vertices between the paths except the end points.*

Definition 8 *A graph G with more than k vertices is k-vertex connected (or k-connected) if $G[V \backslash X]$ remains connected for every set $X \subseteq V$ with $|X| < k$.*

Definition 9 *A subset $S \subseteq V$ is called an s-plex, if $\deg_{G[S]}(v) \geq |S| - s, \forall v \in S$.*

Definition 10 *A subset $S \subseteq V$ is an s-club, if diam$(G[S]) \leq s$.*

Definition 11 *A subset $S \subseteq V$ is a dominating set for graph G if every vertex in G is in S or is adjacent to some vertex in S.*

The problems considered in this dissertation are clustering models, and a major focus is on providing good solution approaches. We study the computational complexity of the problems considered, in particular, with a number of clustering models in the literature proved to be NP-hard, we check to see if the models considered exhibit similar properties. For detailed review of complexity theory and approximation algorithms, we ask the reader to refer [27, 47] and [61]. In addition, we also provide a detailed study of the polyhedra associated with the optimization problems considered for the models. In this regard, we follow the standard notations and terminology used in [14, 44]

1.3 Contributions

This dissertation aims to study clustering models obtained by relaxing one or more properties of clique. The core objective is to design two kinds of models, one that enables to find clusters that are closely-knit in the sense that it is cohesive, robust, and reliable, and the other focuses on finding clusters with good stability. In particular, we study the graph theoretic relaxations of clique, called *k-block, k-robust s-club* to model structurally cohesive clusters, and a new relaxation, *s-stable cluster*, to model stable clusters. We consider the optimization version of the k-block, k-robust 2-club, and s-stable cluster models, with an eye to meet the objective mentioned above. We study the structural properties associated with these models, analyze the computational complexity of the problems, and show that these problems are hard to solve, which makes this work all the more interesting. Integer programming formulations are presented for the optimization problems, and a detailed study of the associated polytopes is conducted. More specifically, we introduce valid inequalities and different classes of facets for the polytopes. Finally, computational experiments are presented to validate the models. The solution approaches include, directly solving the integer programming formulations, designing a branch and cut algorithm by employing suitable facets, and employing other exact algorithms.

As mentioned in Section 1.1, the models considered in this dissertation were basically introduced due to the various shortcomings seen in clique, one of which is that the maximum clique problem is very hard to solve. The final work in this dissertation tries to address this issue, and uses a clique-relaxation approach to get a good upper bound on the clique number of a graph. Particularly, we use k-*core* and its stronger variants, and introduce minimization problems whose lower bound will give a good upper bound on the clique number. In this work, simple linear programming relaxations of the problems are used to achieve the objective, and we show that the upper bounds found are better than those obtained by the standard linear relaxations of the clique polytope. Computational results presented show that the considered problems have good potential to produce enhanced bounds.

1.4 Organization

The organization of this dissertation is as follows. In Chapter 2, we formally introduce the clustering models studied in this dissertation. This will include a discussion on the structural properties of the models, and their relation with some existing models in the literature. Chapter 3 will focus on cohesive and robust clustering models, and Chapter 4 will discuss the stability model for clusters. In particular, the computational complexity of the models will be discussed, integer programming formulations will be presented, and polyhedral results, valid inequalities and facets will be developed for all the models. In addition, exact algorithms and computational experiments will be presented with detailed analysis. Chapter 5 will introduce a new method for getting strong and easily computable upper bounds for the clique number of a graph. Finally, Chapter 6 will conclude this dissertation with a review of the research contribution made, and provide a brief discussion on directions for future research.

2. CLUSTERING MODELS BASED ON CLIQUE RELAXATIONS

In the previous chapter, we briefly described the broad concept of graph cluster detection, and discussed some of the drawbacks of cliques that led to the introduction of models called clique relaxations, that are obtained by relaxing one or more properties of clique. In this chapter, we discuss the clique relaxations that can be used for modeling cohesive clusters in networks, and we develop a new model for characterizing clusters based on the maximum size of an independent set in the cluster. In addition, we establish some of the basic structural properties of these models.

2.1 Cohesive and Robust Relaxations

The clique relaxation models were defined with the objective to characterize clusters, but at the same time inherit properties of a clique that are desirable in various applications. However, among all the clique-defining properties, robustness to disruption is one of the most important and essential characteristics of a network, and is especially vital in social, information and communication networks. In particular, in social networks, the robustness of a network is measured using the notion of structural cohesion. Moody and White [43] formally define structural cohesion of a group *as the minimum number of actors who, if removed from the group, would disconnect the group.* In graph theoretical terms, this represents the vertex connectivity of a network, and is formalized by the concepts of k-blocks [40], and k-robust s-clubs [62].

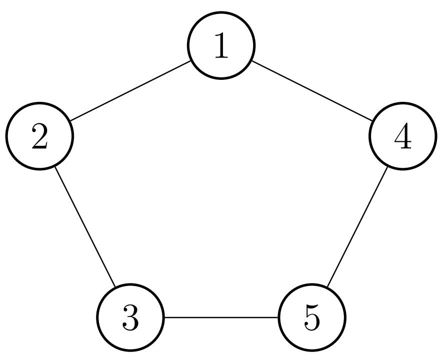
Definition 12 *A subset $S \subseteq V$ is called a k-block if $G[S]$ is k-connected.*

The original definition of a k-block as defined by Matula [40] requires maximality, however, it corresponds to maximal k-blocks in our terminology. This change in terminology is attributed to our purpose of using k-blocks as a model for cohesive subgroups. Hence, in this work a k-block is simply a k-connected graph. Note that, the condition of maximality will camouflage the set of vertices that actually are cohesive, and this will be evident from the way the optimization problem will be defined.

Menger's theorem states that the size of a minimum vertex cut separating a pair of nonadjacent vertices i and j is equal to the maximum number of pairwise disjoint paths between i and j in G. From this, it is easy to note that G is k-vertex connected if and only if thee exists atleast k-vertex disjoint paths between each such pair of vertices in G, which gives an alternative definition of k-block.

Definition 13 *Given a subset of verices, S, if for any pair of vertices the length of at least k-disjoint paths between them in $G[S]$ is restricted by a given parameter s, then S is a* k-robust s-club.

The k-robust s-club model was originally introduced by Veremyev and Boginski [62], as model for robust cluster, which ensures that the diameter or the reachability in the cluster is maintained even after the removal of $(k-1)$ vertices. Essentially, the main difference between a k-block and a k-robust 2-club is the restriction on path lengths in the definition of the latter, see Figure 2.1. A k-block ensures that the passage of information between any two entities in the network is not disrupted, as there exists pairwise disjoint paths between them. Given a parameter k, if there exists a k-block of size $k+1$, then it is a clique, which then becomes an ideal model for a cluster. Hence, it is easy to see that the closer the size of a k-block is to the value of k, the more closely-knit the cluster would be, which also explains our omission of maximality from the definition of k-blocks. However, if the size of the k-block is considerably larger than $k + 1$, then it is possible that the corresponding cluster will have a high diameter, potentially resulting in a high probability of failure in communication channels [62]. This issue is overcome by the k-robust s-club model which imposes restrictions on the length of the paths, and it is easy to note that the model will ensure that the network remains robust and information is passed on quickly, if the value of the path length parameter s is small. Hence, in this work, we concentrate on the special case of the k-robust s-club when $s = 2$, namely a k-robust 2-club.



(a) 2-block (b) 2-robust 2-club

Figure 2.1: Example of a 2-block and a 2-robust 2-club.

It is easy to see that if thee exists no k-block or k-robust s-club of size $k + 1$ in G, then the clique number $\omega(G)$ satisfies $\omega(G) \leq k$. Motivated by these special properties, a study of the following two optimization problems are presented in this dissertation.

Definition 14 (Minimum k-block) *Given a graph $G = (V, E)$ and a positive integer k, find a k-block of minimum cardinality in G. The size of a minimum k-block* *is denoted by $\mu_k(G)$.*

Definition 15 (Minimum k-robust s-club) *Given a graph $G = (V, E)$ and positive integers k and s, find a k-robust s-club of minimum cardinality in G. The size of a minimum k-robust s-club is denoted by $\mu_{k,s}(G)$.*

We study some basic properties associated with a k-block in the context of minimization, which is defined next.

2.1.1 Properties of k-Blocks

Given a k-connected graph $G = (V, E)$, a vertex $v \in V(G)$ is called an *essential* vertex if $G[V \backslash \{v\}]$ is not k-connected. A graph $G = (V, E)$ is called *critically k-vertex connected* if every $v \in V$ is essential [10]. We extend this property to more than one vertex and define a *minimal k-block* as a k-block S such that $G[S']$ is not k-connected for any proper subset $S' \subset S$. For a minimal k-block S, $G[S]$ is critically k-connected, but the converse is not true. This is illustrated in Fig. 2.2. The following lemma provides some basic properties of minimal k-blocks.

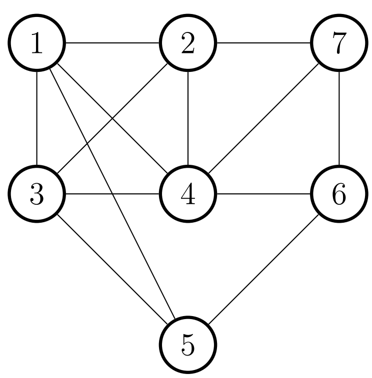


Figure 2.2: A critically 3-connected graph $G = (V, E)$, where V is a 3-block that is not minimal, since $S = \{1, 2, 3, 4\}$ is also a 3-block.

Lemma 2.1.1 *(a) If S is a minimal k-block and $k \geq 2$, then $\delta(G[S]) < \frac{3k-1}{2}$.*

(b) Given a critically k-connected graph $G = (V, E)$, it can be checked in polynomial time if V is a minimal k-block.

(c) For any positive integer $k \geq 1$, the ratio μ_{k+1}/μ_k can be arbitrarily large.

Proof (a) Follows from the result by [10] who have shown that for a critically k-connected graph G, $k \geq 2$, $\delta(G) < \frac{3k-1}{2}$ and the number $\frac{3k-1}{2}$ cannot be improved. (b) Checking whether V is a minimal k-block reduces to determining whether there exists a vertex $i \in V$ such that $G[V \backslash \{i\}]$ contains a k-block. Checking the existence of a k-block can be done in polynomial time [34]. (c) Let G be a graph with the set of vertices given by $A \cup B$, where A is a clique of size $k+1$, B is a tree of height l such that the degree of the internal vertices is $k+1$ and root node has degree k. Let r denote the root of this tree. For $a_1 \in A$, add an edge between every leaf vertex and every vertex in the set $A \backslash \{a_1\}$ and add an edge between r and a_1. Note that A is the minimum k-block in G, and the only $(k+1)$-block in G is $A \cup B$. Thus, $\mu_k = k+1$ and $\mu_{k+1} = k + 1 + \frac{k^{l+1}-1}{k-1}$. Note that we can construct such a graph for any $l \geq 1$. ∎

2.2 Stable Relaxations

In this section, a new clique relaxation is introduced, that characterizes clusters, based on the number of vertices acting independently in the cluster. An alternative description of a clique in G is a subset of vertices S, such that $\alpha(G[S]) = 1$. In other words, if we define "stability" of a cluster as the size of the maximum independent set in the cluster, then the stability of clique is 1, and is one of its very important properties. In addition to the various models discussed that were based on relaxing one or more of the clique-defining properties, stability is another important property that is used for characterizing clusters in various applications, particularly in fullerene chemistry. In fullerene chemistry [20, 25], experiments find a large number of isomers with n atoms, and an important problem is to characterize stable isomers and distinguish them from unstable ones. A key observation that isomers that minimize their independence number are more stable than others, has introduced independence number as a predictor of fullerene stability. However, these models do not always require that the stability of the structure is one. In particular, relaxing the stability property of a clique could help detect clusters with varying stability, which will in turn be useful in understanding the stability of the associated network associated with the cluster. Hence, in this work, we introduce a new clique relaxation model obtained by relaxing the "stability" property of a clique, called the "s-stable cluster", which is defined as follows.

Definition 16 *A subset of vertices S is called a s-stable cluster if the stability number of the induced subgraph $\alpha(G[S]) \leq s$.*

In other words, any independent set in the induced subgraph $G[S]$ must be of size at most s. Note that this model is an absolute relaxation, as the condition on $\alpha(G[S]$ is independent of the cardinality of set S. The s-stable cluster model corresponds to a clique when $s = 1$, and relaxes the size of the stable set in the induced subgraph when $s > 1$. We consider the following optimization problem associated with s-stable clusters.

Definition 17 (Maximum s-stable cluster) *Given a graph $G = (V, E)$, and a positive integer s, find an s-stable cluster of the largest size in G. The cardinality of a maximum s-stable cluster is denoted by $\omega_s(G)$.*

Next, we present some of the basic properties associated with s-stable clusters, and establish relations with some existing clique relaxations.

2.2.1 Properties of s-Stable Clusters

Given a graph $G = (V, E)$, any s-stable cluster in G can be connected or disconnected, see Figure 2.3. Any s-stable cluster that is disconnected can have at most s connected components, and if there are s components, then each component must be a clique. Also, note that any s-plex is an s-stable cluster, but the converse need not be true. Some of the other basic properties of s-stable clusters are described next.

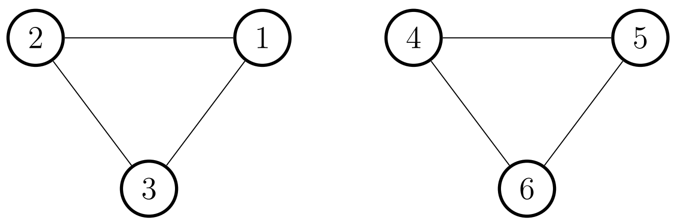


Figure 2.3: Example of a 2-stable cluster.

Lemma 2.2.1 *Let $G = (V, E)$ be a s-stable cluster. Then,*

(1) Any vertex-induced subgraph of G is an s-stable cluster.

(2) If G is connected, then $diam(G) \leq 2s - 1$.

Proof (1) As G is a s-stable cluster, $\alpha(G) \leq s$, then for any subset $S \subseteq V$ of vertices, we have $\alpha(G[S]) \leq s$, hence S is a s-stable cluster.

(2) Suppose $diam(G) > 2s - 1$. Then, there is a pair of vertices $i, j \in V$ such that $d_G(i, j) > 2s - 1$. This implies there exist aleast $2s - 1$ vertices, say, $v_1, v_2, \ldots, v_{2s-1}$ in the shortest path between u and v in G. Hence the vertices

$i, v_2, v_4, \ldots, v_{2s-2}, j$ form a stable set of size $s+1$, which cannot be true as G is an s-stable cluster. Hence, $diam(G) \leq 2s-1$. ∎

This implies that s-stable cluster is hereditary on induced subgraphs, and for small values of s, a connected s-stable cluster has good reachability properties. These properties, show the close relationship that an s-stable cluster has with other clique relaxation models. The similarities that an s-stable cluster enjoys with a clique will also become evident when we explore the polytope associated with the optimization problem in the forthcoming chapters.

2.3 Conclusion

In this chapter, cohesive, robust and stable models of clique relaxations were discussed. k-blocks and k-robust s-clubs considered here for designing cohesive and robust clusters are motivated by clusters that require a closely-knit structure, which may not be captured by the other clique relaxations. In addition, these models are much stronger than some other clique relaxation models, like k-cores or s-clubs, whose properties are in fact, encompassed by k-blocks and k-robust s-clubs, which makes studying these models interesting and worthwhile.

With the existence of so many models for studying and characterizing clusters satisfying various requirements, one might ask, if these existing models are not enough or in what way the s-stable cluster is superior, to consider it for clustering applications. At this point, we would like to point out that the existing models were introduced to satisfy specific needs in various applications that found clique to be of restrictive use. In the same sense, we introduce s-stable clusters as another model, which is most appropriate for applications that require models to minimize its independence number to maintain its stability.

3. THE MINIMUM K-BLOCK AND K-ROBUST 2-CLUB PROBLEMS

This chapter focuses on the cohesive and robust relaxation models defined and motivated in Chapter 2. In particular, we study the minimum k-block and the k-robust 2-club problems in detail. The k-block problem as defined by [40] is well studied, however using this model for modeling cohesive subgroups may not be a very interesting approach, as increasing the size of the k-block will not result in a cluster that is cohesive in terms of distance between two vertices. This is overcome by the minimum k-block and k-robust 2-club problems which will be discussed in this chapter.

3.1 Computational Complexity

In this section, we study the hardness of approximating the minimum k-block and the minimum k-robust 2-club problems. In addition, the computational complexity of the augmentation version of these two problems will be discussed in detail.

For $k = 1$ and $k = 2$, the minimum k-block problem coincides with the minimum k-core problem and is easy to solve. Indeed, for $k = 1$ any pair of adjacent vertices is an optimal solution, and for $k = 2$ one needs to compute the length of a shortest cycle in the graph. For $k \geq 3$, we will prove that the minimum k-block problem is hard to approximate by using a gap-preserving reduction from VERTEX COVER on k-regular graphs and the following fact.

Theorem 3.1.1 VERTEX COVER *is APX-complete on k-regular graphs, for any fixed $k \geq 3$.*

Proof The minimum vertex cover on k-regular graphs was proved to be APX-complete only for $k = 3$ (cubic graphs) and also on graphs with degree bounded by k [3, 48]. [23] proved that the vertex cover problem is "hardest to approximate in regular graphs." This, however, does not imply that the result is valid for k-regular graphs for each fixed k. We use the notion of L-reduction [48] defined next, that is widely used for establishing APX-hardness results. Given two optimization problems F and G, we say that F L-reduces to G if there are two polynomial-time algorithms f, g and constants $\alpha, \beta > 0$ such that for each instance x of F:

1. f produces an instance $f(x)$ of G, such that $OPT_G(f(x)) \leq \alpha OPT_F(x)$.

2. Given any solution of $f(x)$ with cost c', g produces a solution of x with cost c such that $|c - OPT_F(x)| \leq \beta |c' - OPT_G(f(x))|$.

If F L-reduces to G and there is a polynomial-time approximation algorithm for G with worst-case error ϵ, then there is a polynomial-time approximation algorithm for F with worst-case error $\alpha\beta\epsilon$. We now give the following L-reduction f from Min Vertex Cover on 3-regular graphs to Min Vertex Cover on k-regular graphs, $k \geq 4$. Given a 3-regular graph $G = (V, E)$ construct a k-regular graph $G' = (V', E')$ as follows. Let $|V(G)| = n$. Consider k identical copies $G_1, G_2, ..., G_k$ of G. Denote the vertex set and edge set of the r^{th} such copy respectively by V_r and $E_r, r = 1, \ldots, k$, where $V_r = \{1_r, \ldots, n_r\}$ and $E_r = \{(i_r, j_r) : (i, j) \in E(G)\}$. Let $R = \cup_{r=1}^{k} V_r$ and $E_R = \cup_{r=1}^{k} E_r$. For each $v \in V(G)$ consider a set of $(k-3)$ independent vertices $P_v = \{u_1^v, \ldots, u_{k-3}^v\}$, and let $P = \cup_{v \in V} P_v$. Put $V' = R \cup P$ and $E' = E_R \cup E_P$, where

$$E_P = \{(v_r, u_j^v) : v \in V, r = 1, \ldots, k, j = 1, \ldots, k-3\}$$

That is for each $v \in V$, there is an edge between v_r and u_j^v for $j = 1, \ldots, k-3$ and $r = 1, \ldots, k$. This completes the construction of $G' = (V', E')$.

It is easy to see that from every vertex cover $C \subseteq V$ of G we can construct a vertex cover $C' \subseteq V'$ of G' of size exactly $n(k-3)+3|C|$, by choosing $v_r \in V_r, r = 1, \ldots, k$ if $v \in C$ and by choosing the set P_v if $v \notin C$. Since G is 3-regular we have $n = |V| \leq |E| \leq \sum_{v \in C} deg(v) = 3|C|$. Then, $|C'| = n(k-3) + 3|C| \leq 3(k-3)|C| + 3|C| = 3(k-2)|C|$ and, this satisfies the first property of L-reduction with $\alpha = 3(k-2)$.

Conversely, given any vertex cover $C' \subseteq V'$ of G', we can transform it back to a vertex cover $C \subseteq V$ of G as follows. First note that if $v_r \notin C'$ for some $r \in \{1, \ldots, k\}$, then $P_v \subseteq C'$. Hence given any vertex cover C', if $C_1, C_2, \ldots, C_k \subseteq C'$ are the subsets of vertices selected respectively from $V_1, \ldots, V_k$, then C_i must be a vertex cover of G_i for each $i \in \{1, \ldots, k\}$ and hence, the corresponding vertices in V must be a vertex cover of G. Let h be such that $|C_h| = \min\{|C_1|, \ldots, |C_k|\}$. Then $C = \{v \in V : v_h \in C_h\}$ is a vertex cover of G and $|C| \leq \frac{1}{3}(|C'| - n(k-3))$. Together with the observation $|OPT_{VC}(G')| \leq 3|OPT_{VC}(G)| + n(k-3)$ from the previous paragraph, it is easy to see that f is a L-reduction with $\beta = 1$.

We now use this result and extend the construction used by [4] to prove the following:

Proposition 3.1.2 *The minimum k-block problem does not admit a PTAS for any fixed $k \geq 3$, unless $P = NP$, even if restricted to k-connected graphs.*

Proof : For any $k \geq 3$, we give a gap-preserving reduction [61] from Vertex Cover on k-regular graphs. Given a k-regular graph $G = (V, E)$ on n vertices, we construct an instance G'_k of the minimum k-block problem. Without loss of generality, assume that $|E(G)| = \frac{kn}{2} = k(k-1)^l$ for some integer $l \geq 2$.

The construction described by [4] is as follows. Let T be a tree of height $l+1$ such that all its internal vertices have degree k and the number of leaves is $k(k-1)^l$. The total number of vertices in T is $1 + \frac{k((k-1)^{l+1}-1)}{k-2}$. Let I and F represent the

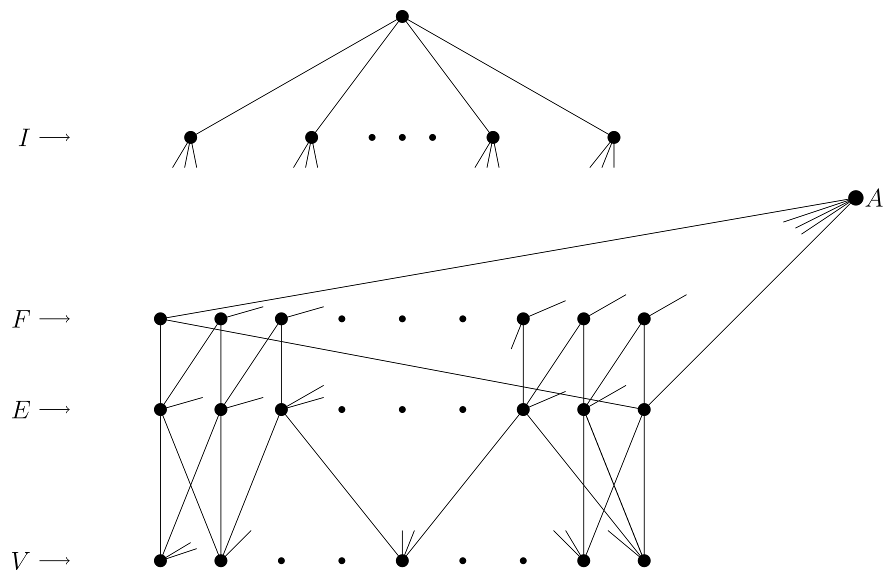


Figure 3.1: Graph G'_k constructed in Proposition 3.1.2 for $k = 4$.

set of internal vertices and leaves of T, respectively. Add a set of vertices E of cardinality $k(k-1)^l$ such that each of its vertices uniquely represents an edge from $E(G)$. Construct a Hamilton cycle with $E \cup F$ inducing a bipartite graph with E and F as the partition classes. Add the vertex set V. Since each vertex in E uniquely represents an edge in graph G, join the sets E and V according to the incidence relations in G, i.e., add an edge between a vertex in E that corresponds to the edge $e \in E(G)$ and a vertex $v \in V(G)$ if and only if edge e is incident to vertex v in G.

To complete the construction of $G'_k = (V'_k, E'_k)$ (see Fig. 3.1), we add a set of independent vertices A of cardinality $(k-3)$ and add two complete bipartite graphs with partition classes A and E, and A and F, respectively.

We prove that minimum k-blocks of G'_k correspond to minimum vertex covers of G and vice-versa. Note that any k-block H of G'_k must include at least one vertex from the set $V'_k \backslash V$, since V is an independent set in G'_k. Including any vertex from $V'_k \backslash V$ in H implies that $V'_k \backslash V \subseteq H$, because of degree constraints. Let $V_1 = V'_k \backslash V$.

By construction, there are $(k-1)$ disjoint paths between each pair of vertices in the set $E \cup F \cup A$. Hence, the set $E \cup F \cup A$ is a $(k-1)$-block. In the tree T, each vertex in level l is adjacent to at least $(k-1)$ vertices in level $(l+1)$. This implies that V_1 is a $(k-1)$-block. Notice that by construction, E represents the edge set of G and all its vertices have degree $k-1$ in $G'_k[V_1]$. Hence, it is necessary to include a minimum number of vertices from the set V, say $J \subseteq V$ to satisfy the degree requirements and such a set J will be a vertex cover of G. Let $H = V_1 \cup J$.

Next, we show that H is a k-block. Let S be any subset of H such that $|S| = k-1$. It is enough to prove that $G'_k[H \backslash S]$ is connected and for this, we will use the notion of k-fan, which is defined as follows. Given a subset $U \subseteq V$ and a vertex $v \in V \backslash U$, a set of $v - U$ paths is called a *$v - U$ fan* if any two paths have only v in common, and if $|U| = k$, then it is a k-fan.

Case 1: $S \subseteq V_1$. If $|A \cap S| < k-3$ then $G'_k[H \backslash S]$ is connected as $G'_k[V_1]$ is connected. If $A \subseteq S$, then it is enough to prove that $G'_k[H \backslash A]$ is 3-connected. Since for any vertex $t \in I$, there exists a set of k vertices $F_t \subseteq F$ such that $t - F_t$ forms a k-fan, removal of any 2 vertices from I does not disconnect $G'_k[H \backslash S]$. Removal of any 2 vertices from $E \cup F$ splits $E \cup F$ into two components with one of the following three possibilities: there is an isolated vertex $e \in E$, there is an isolated vertex $f \in F$, or both components have at least one vertex each from the sets E and F. In the first two cases, $G'_k[(E \cup J) \backslash S]$ and $G'_k[(I \cup F) \backslash S]$ remain connected, and in the last case $G'_k[(E \cup F \cup I) \backslash S]$ always remains connected. Hence, $G'_k[H \backslash S]$ is connected.

Case 2: $S \subseteq J$. J is an independent set and each vertex in J is adjacent to k vertices in E. Hence, $G'_k[H \backslash S]$ is connected.

Case 3: $S \cap V_1 \neq \emptyset$ and $S \cap J \neq \emptyset$. This is a subset of the previous two cases.

In all the three cases above, H is a k-block. This implies that since $V'_k \backslash V$ is included in any k-block H of G'_k, the minimum k-block problem in G'_k corresponds to finding the smallest subset of vertices in V covering all the vertices in E, which is exactly the minimum vertex cover problem for G, and the minimum vertex covers of G corresponds to minimum k-blocks of G'_k. Thus,

$$\begin{aligned}\mu_k(G'_k) &= |T| + |E| + |A| + |C^*| \\ &= \frac{k(k-1)^l(2k-3) - 2 + (k-2)(k-3)}{k-2} + |C^*|,\end{aligned}$$

where C^* is a minimum vertex cover of G.

Assume that for an $\epsilon > 0$, we can find in polynomial time a solution to the minimum k-block problem of size $S_k(G'_k)$ in G'_k, such that $S_k(G'_k) \leq (1+\epsilon)\mu_k(G'_k)$. Using the above argument, any minimum k-block in G' corresponds to a vertex cover of size $S_{vc}(G)$ in the given graph G such that

$$S_k(G'_k) = \frac{k(k-1)^l(2k-3) - 2 + (k-2)(k-3)}{k-2} + S_{vc}(G),$$

Then,

$$\begin{aligned}S_{vc}(G) &= S_k(G'_k) - \frac{k(k-1)^l(2k-3) - 2 + (k-2)(k-3)}{k-2} \\ &\leq (1+\epsilon)\mu_k(G'_k) - \frac{k(k-1)^l(2k-3) - 2 + (k-2)(k-3)}{k-2} \\ &= (1+\epsilon)\||C^*| + \epsilon(\frac{k(k-1)^l(2k-3) - 2 + (k-2)(k-3)}{k-2})\end{aligned}$$

Since G is a k-regular graph, we have $|E(G)| = k(k-1)^l = \frac{kn}{2}$, and any optimal vertex cover $|C^*| \geq \frac{kn}{2}$. Then,

$$\begin{aligned} S_{vc}(G) &\leq (1+\epsilon)\|C^*| + \epsilon(\frac{kn(2k-3)}{2(k-2)} + \frac{-2+(k-2)(k-3)}{k-2}) \\ &\leq (1+\epsilon)\|C*| + \epsilon(\frac{|C^*|(2k-3)}{k-2} + k) \\ &\leq (1+\epsilon(1+\frac{k(3k-5)}{k-2}))|C^*| \end{aligned}$$

Hence, existence of a PTAS for the minimum k-block problem implies a PTAS for the minimum vertex cover. The result follows from Proposition 3.1.1. ∎

Remark Given a graph $G = (V, E)$, a subset $S \subseteq V$ of vertices is called a k-connected d-dominating set (k-d-CDS) if $\kappa(G[S]) \geq k$ and $|N(i) \cap S| \geq d, \forall i \in V \backslash S$. In the graph G'_k constructed in Proposition 3.1.2, a minimum k-block in G'_k is a minimum k-d-CDS for G'_k for $d \leq k$ and vice-versa. This implies that finding a minimum k-d-CDS for $k \geq 3$, $d \leq k$ is equivalent to finding a minimum k-block in G'_k and hence, the minimum k-d-CDS problem, where $k \geq 3$, $d \leq k$, does not admit a PTAS, unless P=NP.

The minimum k-robust 2-club problem can be solved in polynomial time for $k = 1, 2$ as it coincides with the minimum k-core problem for $k = 1$, and asks to find the smallest cycle of length at most 4 for $k = 2$. For $k = 3$, the complexity of the problem remains open. We prove the following result for $k \geq 4$.

Theorem 3.1.3 *The minimum k-robust 2-club problem does not admit a PTAS for any fixed $k \geq 4$, unless P=NP, even if restricted to graphs with the set of vertices forming a k-robust 2-club.*

Proof For any $k \geq 4$, we give a gap-preserving reduction from vertex cover on $(k-1)$-regular graphs. Given a $(k-1)$-regular graph $G = (V, E)$ on n vertices, we construct an instance $G''_k = (V''_k, E''_k)$ of the minimum k-robust 2-club problem, for $k \geq 4$. Without loss of generality, assume that $|E(G)| = \frac{(k-1)n}{2} = (k-1)(k-2)^l$ for some integer $l \geq 2$.

Consider the graph G'_{k-1} constructed in Proposition 3.1.2. Put $V''_k = V'_{k-1} \cup C$, and $E''_k = E'_{k-1} \cup E_1$, where C is a set of k independent vertices and $E_1 = \{(u, v) : u \in C, v \in V'_{k-1}\}$. This completes the construction of $G''_k, \forall k \geq 4$. Then, with a similar argument as in the proof of Proposition 3.1.2, we can prove that the minimum k-robust 2-club, for any fixed $k \geq 4$ does not admit a PTAS, unless P=NP. ∎

Remark Given a graph $G = (V, E)$, a subset $S \subseteq V$ of vertices is called a d-dominating k-robust 2-club if S is a k-robust 2-club and $|N(i) \cap S| \geq d, \forall i \in V \backslash S$. In the graph G''_k constructed in Proposition 3.1.3, a minimum k-robust 2-club in G''_k is also a minimum d-dominating k-robust 2-club for G''_k, for $d \leq 2k-1, k \geq 4$ and vice-versa. This implies that finding a minimum d-dominating k-robust 2-club, for $d \leq 2k-1, k \geq 4$, is equivalent to finding a minimum k-robust 2-club in G''_k and hence, the minimum d-dominating k-robust 2-club, where $d \leq 2k-1$ and $k \geq 4$, does not admit a PTAS, unless P=NP.

3.1.1 Augmentation Problems

Consider the augmentation version of the minimum k-block problem [39], the k-Vertex Connected Subgraph Augmentation Problem (k-VCSAP) that is defined as follows: Given a k-vertex connected graph $G = (V, E)$ and a subset $S \subseteq V$, find a smallest subset $S' \subseteq (V \backslash S)$ of vertices, such that the set $S \cup S'$ is a k-block. k-VCSAP can be seen as a generalization of the minimum k-block problem, where the minimum k-connected network designed, must include a given set of vertices.

The augmentation version of the minimum k-block problem can be used for enhancing the reliabiliy of an existing network and has application in fault-tolerance and wireless networks [39]. In addition to connectivity, one of the measures that is most commonly used for designing reliable networks and has been well studied, is its diameter [22, 36, 63, 68]. This gives rise to the augmentation version of the k-robust 2-club problem, which can be defined in a similar fashion as the k-VCSAP.

[39] proved that k-VCSAP is APX-hard, and gave a lower bound of $O(\log(\log n))$ on the approximation ratio for polynomial-time algorithms under the assumption P≠NP. We prove an analogous result for the k-robust 2-club subgraph augmentation problem.

Proposition 3.1.4 *The k-robust 2-club subgraph augmentation problem does not admit a PTAS for $k \geq 4$, unless P=NP.*

Proof Consider the graph $G''_k = (V''_k, E''_k)$ constructed in Proposition 3.1.3. Note that V''_k is a k-robust 2-club. Then, given a $(k-1)$-robust 2-club $S = V''_k \backslash V$ in G''_k, finding a smallest set $S' \subseteq V''_k \backslash S$ such that $S' \cup S$ is a k-robust 2-club is equivalent to finding a minimum vertex cover of G. The result follows from Proposition 3.1.3. ∎

Given a set of elements $U = \{1, 2, ..., n\}$ and a set N of subsets of U whose union is U, the minimum set cover problem asks to find a smallest subset S of N whose union is U. It is known that unless P=NP, approximating the set cover problem within $c \log n$ is NP-hard for some constant c [54]. We use this result to prove the following.

Proposition 3.1.5 *Unless P=NP, approximating the k-robust 2-club subgraph augmentation problem within $c \log(\log n)$ for some constant c is NP-hard, where n is the number of vertices in the problem instance.*

Proof Given an instance (U, N) of the minimum set cover problem such that $|U| > k$, we construct an instance $(G = (V, E), S)$ of the k-robust 2-club subgraph augmentation problem such that a minimum cover of U implies a minimum set $S' \subseteq (V \backslash S)$

such that $S \cup S'$ induces a k-robust 2-club in G and vice-versa. The construction of G is as follows.

Let $V = U' \cup N' \cup V_1 \cup V_2$, where the vertices in U' and N' distinctly represent the elements in U and the sets in N, respectively, and $G[V_1], G[V_2]$ are complete graphs, each of size $k-1$. Add edges between vertices in the sets V_1 and U', V_1 and N', V_2 and N', and U' and V_2. Two vertices $N_i \in N'$ and $u \in U'$ are connected if and only if $u \in N_i$ in the given instance of set cover. This completes the construction of G with $S = V \backslash N'$. It is easy to see that S is a $(k-1)$-robust 2-club, as there exists exactly $(k-1)$ paths of length at most 2 between any vertex $u \in V_1 \cup V_2$ and $v \in U'$. Also, there exists no k-robust 2-club $R \subseteq S$. Therefore, from the construction of G, it is easy to see that finding a minimum subset $S'$of N' such that $S \cup S'$ is a k-robust 2-club in G is equivalent to finding a minimum cover for the given instance of set cover.

Using the argument in [39], since N could be the power set of U, we have $|N| \leq 2^{|U|}$ and $n = |V| \leq 2^{|U|} + |V_1| + |V_2| + |U|$. Then, for some constants c, c' and a sufficiently large n, $|U| \geq c' \log n$ and

$$c \log |U| \geq c \log(c' \log n) = c \log c' + c \log \log n \geq c \log \log n$$

∎

3.2 Mixed Integer Programming Formulations

Mathematical programming formulations for the minimum k-block and minimum k-robust 2-club problems are presented, and the associated polytopes are studied in this section.

3.2.1 The k-Block Polytope

Consider a graph $G = (V, E)$. For any set $S \subseteq V$, let x_S denote the incidence vector of S. Given any two nonadjacent vertices $s, t \in V$, a set of vertices $T_{st} \subset V$ is called an s-t separator if its removal results in a disconnected graph that has s and t in different connected components.

The proposed formulation for the minimum k-block problem is based on the following equivalent characterization of a k-block. A subset $S \subseteq V$ is a k-block if and only if for all the minimal s-t separators T_{st} in G, $|S \cap T_{st}| \geq k$, for any pair of nonadjacent vertices $s, t \in S$. Let $\mathcal{T}_{st}$ denote the set of all minimal s-t separators in G. We have:

$$\mu_k(G) = \min \quad \sum_{i \in V} x_i \tag{3.1}$$

$$\text{subject to} \quad \sum_{v \in T_{st}} x_v \geq k(x_s + x_t - 1), \forall T_{st} \in \mathcal{T}_{st}, \forall s, t : (s, t) \notin E \tag{3.2}$$

$$\sum_{i \in V} x_i \geq k + 1 \tag{3.3}$$

$$x \in \{0, 1\}^{|V|}. \tag{3.4}$$

Constraint (5.2) ensures that the subgraph induced by the subset of vertices $\{i : x_i = 1\}$ is k-vertex connected. This being a minimization problem, constraint (3.3) ensures that zero is not included in the feasible solution. The number of constraints of type (5.2) can be exponential, which makes it very difficult to solve this problem.

Consider an alternative formulation based on multi-commodity flow [53, 64] for the minimum k-block problem as a consequence of Menger's theorem. Let f_{ij}^{st} be the flow from s to t that passes through edge $(i, j) \in E$. Then the minimum k-block problem can be formulated as:

$$
\begin{aligned}
\mu_k(G) = \min \quad & \sum_{i \in V} x_i && (3.5)\\
\text{subject to} \quad & \sum_{j \in V} f_{sj}^{st} - \sum_{j \in V} f_{js}^{st} \geq k(x_s + x_t - 1), \quad \forall s, t \in V && (3.6)\\
& \sum_{j \in V} f_{jt}^{st} - \sum_{j \in V} f_{tj}^{st} \geq k(x_s + x_t - 1), \quad \forall s, t \in V && (3.7)\\
& \sum_{j \in V} f_{vj}^{st} - \sum_{j \in V} f_{jv}^{st} = 0, \quad \forall v \in V \backslash \{s, t\}, \quad \forall s, t \in V && (3.8)\\
& \sum_{j \in V} f_{vj}^{st} \leq x_v, \quad \forall v \in V \backslash \{s, t\}, \quad \forall s, t \in V && (3.9)\\
& \sum_{i \in V} x_i \geq k + 1 && (3.10)\\
& 0 \leq f_{ij}^{st} \leq 1, \quad \forall (i, j) \in E, \quad \forall s, t \in V && (3.11)\\
& x \in \{0, 1\}^{|V|}. && (3.12)
\end{aligned}
$$

Constraints (3.6) & (3.7) ensure that there exists a flow of at least k units from origin $s \in V$ to destination $t \in V$, for every pair of vertices s, t included in the subset $\{i : x_i = 1\}$ that induces a k-vertex connected subgraph. Constraint (3.8) represents the flow balance constraints, and constraint (3.9) ensures that there is flow only in the subset of vertices $\{i : x_i = 1\}$ that induces a k-vertex connected subgraph. Finally, constraint (3.10) ensures that zero is not included in the feasible solution.

Let $P_k(G)$ denote the k-block polytope of G that does not include an empty k-block, and for any $v \in V$, let S_v denote a k-block in G such that $v \in S_v$.

Theorem 3.2.1 *Consider a graph $G = (V, E)$ with the following property: For each vertex $v \in V$, there exists a k-block S_v in G such that $v \in S_v$ is not an essential vertex of $G[S_v]$.*

1. *$dim(P_k(G)) = |V|$.*

2. *$x_v \geq 0$ induces a facet of $P_k(G)$ if for each $w \in V\backslash\{v\}$, there exists a k-block S_w in G such that $w \in S_w$ is not an essential vertex of $G[S_w]$ and $v \notin S_w$.*

3. *$x_v \leq 1$ induces a facet of $P_k(G)$ if for each $w \in V\backslash\{v\}$, there exists a k-block S_w in G such that $w \in S_w$ is not an essential vertex of $G[S_w]$ and $v \in S_w$.*

Proof 1. We prove there are no implicit equalities in $P_k(G)$. Suppose there is a valid inequality $\alpha^T x \leq \beta$ such that $\alpha^T x = \beta$, $\forall x \in P_k(G)$. For each $v \in V$ the incidence vectors $x_{S_v}, x_{S_v\backslash\{v\}} \in P_k(G)$, and hence, $\alpha^T x_{S_v} = \beta$ and $\alpha^T x_{S_v\backslash\{v\}} = \beta$. This implies that $\alpha_v = 0, \forall v \in V$ and $\beta = 0$. Hence, there is no such implicit equality in $P_k(G)$ and $P_k(G)$ is full dimensional.

2. Let $F_v^0 = \{x \in P_k(G) : x_v = 0\}$. Let there be a valid inequality $\alpha^T x \leq \beta$ such that $F = \{x \in P_k(G) : \alpha^T x = \beta\} \supseteq F_v^0$. For each $w \in V\backslash\{v\}$, $v \notin S_w$ and $x_{S_w}, x_{S_w\backslash\{w\}} \in F_v^0$. Then $\alpha^T x_{S_w} = \beta$, $\alpha^T x_{S_w\backslash\{w\}} = \beta$ implies that $\alpha_w = 0, \forall w \in V\backslash\{v\}$ and $\beta = 0$. Hence, $F_v^0 = F$ is a facet.

3. Let $F_v^1 = \{x \in P_k(G) : x_v = 1\}$. Let there be a valid inequality $\alpha^T x \leq \beta$ such that $F = \{x \in P_k(G) : \alpha^T x = \beta\} \supseteq F_v^1$. For each $w \in V\backslash\{v\}$, $v \in S_w$ and $x_{S_w}, x_{S_w\backslash\{w\}} \in F_v^1$. Then $\alpha^T x_{S_w} = \beta$, $\alpha^T x_{S_w\backslash\{w\}} = \beta$ implies that $\alpha_w = 0, \forall w \in V\backslash\{v\}$ and $\beta = \alpha_v$. Hence, $F_v^1 = F$ is a facet.

Note that given a graph $G = (V, E)$, if there exist $v \in V$ such that there exists no k-block in G containing v, then $P_k(G)$ is not full dimensional.

Corollary 3.2.2 *Given a $(k+1)$-connected graph $G = (V, E)$,*

1. *$dim(P_k(G)) = |V|$.*

2. *$x_v \geq 0$ induces a facet of $P_k(G)$ if $|T_{ij}^v| \geq k+2$, $\forall T_{ij}^v \in \mathcal{T}_{ij}$, and $\forall i, j \in V$, where T_{ij}^v is a minimum i-j separator that contains v.*

3. $x_v \leq 1$ *induces a facet of* $P_k(G)$ *for every* $i \in V$*.*

Theorem 3.2.3 *Consider a graph* $G = (V, E)$ *that satisfies the conditions of Theorem 3.2.1. Let* T_{ij} *denote a minimal* i*-*j *separator for any pair of vertices* $i, j \in V$*. Let* $R \subseteq V$ *be an inclusion-wise maximal set such that for any pair of vertices* $i, j \in R$*, there exists a* T_{ij} *with* $|T_{ij}| < k$*. Then,* $\sum_{i \in R} x_i \leq 1$ *induces a facet of* $P_k(G)$ *if for each vertex* $v \in V \backslash R$*, there exists a* k*-block* S_v *in* G *such that* $v \in S_v$ *is not an essential vertex of* $G[S_v]$ *and* $R \cap S_v \neq \emptyset$*.*

Proof The validity of the inequality follows from the equivalent characterization of k-block given above. We now prove that $F_R = \{x \in P_k(G) : \sum_{i \in R} x_i = 1\}$ is a facet. Let there be a valid inequality $\alpha^T x \leq \beta$ such that $F = \{x \in P_k(G) : \alpha^T x = \beta\} \supseteq F_R$. Then, for each $v \in V \backslash R$, there exists a k-block S_v in G such that $x_{S_v} \in F_R$. Then $\alpha^T x_{S_v} = \beta$, $\alpha^T x_{S_v \backslash \{v\}} = \beta$ implies that $\alpha_v = 0$, $\forall v \in V \backslash R$. Also, for each $i \in R$, $x_{S_i} \in F_R$. Hence, for each $i \in R$, $\alpha^T x_{S_i} = \beta$ and $\beta = \alpha_i$. Hence, $F_R = F$ is a facet. ∎

Note that maximality of R is essential in Theorem 3.2.3, since if R is not maximal, the last condition cannot be satisfied for each $v \in V \backslash R$ and $\sum_{i \in R} x_i \leq 1$ does not induce a facet of $P_k(G)$.

Let I denote a maximal independent set in G. Given a k-block S in G, if $v \in S \cap I$ then $|N_G(v) \cap S| \geq k$. Then $\sum_{v \in V \backslash I} x_v \geq k$ is valid for $P_k(G)$.

Theorem 3.2.4 *Consider a* $(k+1)$*-connected graph* $G = (V, E)$*. Then for any maximal independent set* I *of size* k *or more in* G*,* $\sum_{v \in V \backslash I} x_v \geq k$ *induces a facet of* $P_k(G)$ *if* $(i, j) \in E$ *for each* $i \in I$*,* $j \in V \backslash I$*.*

Proof Let $F_I^1 = \{x \in P_k(G) : \sum_{v \in V \backslash I} x_v = k\}$ denote the face induced. Let there be a valid inequality $\alpha^T x \leq \beta$ such that $F = \{x \in P_k(G) : \alpha^T x = \beta\} \supseteq F_I^1$. Let

$S = S' \cup I$, where $S' \subseteq V \backslash I$ and $|S'| = k$. Then $x_S \in F_I^1$. Consider a vertex $v \in S'$ and $w \in V \backslash S$ and let $R = (S \cup \{w\}) \backslash \{v\}$. Then $x_R \in F_I^1$ and $\alpha^T x_R = \beta$, $\alpha^T x_S = \beta$ implies $\alpha_v = \alpha_w$. Since S, v and w are arbitrary $\alpha_v = \mu, \forall v \in V \backslash I$, for some scalar μ. If $|I| > k$, consider a subset P' of V such that $|P'| = k$ (For the case $|I| = k$, since G is $(k+1)$-connected, we can find a set of vertices $P' \subseteq V \backslash I$ such that $G[P']$ is connected and $|P'| = k$). For each $i \in I$, let $P_i = P' \cup (I \backslash \{i\})$ and $P = P' \cup I$, then $x_{P_i}, x_P \in F$. Then $\alpha^T x_{P_i} = \beta$, $\alpha^T x_P = \beta$ implies $\alpha_i = 0, \forall i \in I$ and $\beta = k\mu$, proving that F_I^1 is a maximal face and hence a facet. ∎

3.2.2 The k-Robust 2-Club Polytope

Veremyev and Boginski [62] gave a compact formulation for the maximum k-robust 2-club problem. Let $A = [a_{ij}]_{i,j=1}^n$ be the adjacency matrix of G. Then, the minimum k-robust 2-club problem can be formulated as:

$$\mu_{k,2}(G) = \min \quad \sum_{i \in V} x_i \tag{3.13}$$

$$\text{subject to} \quad a_{ij} + \sum_{v \in N^{\cap}(i,j)} x_v \geq k(x_i + x_j - 1), \quad \forall i, j \in V \tag{3.14}$$

$$\sum_{i \in V} x_i \geq k + 1 \tag{3.15}$$

$$x \in \{0,1\}^{|V|}, \tag{3.16}$$

where $N^{\cap}(i,j)$ denotes the common neighborhood of vertices i, j in G, that is, $N^{\cap}(i,j) = N(i) \cap N(j)$. Constraint (3.14) ensures that for any pair of vertices in the subgraph induced by the subset of vertices $\{i : x_i = 1\}$, there exists k vertex-disjoint paths, each of length at most 2. Denote by $P_{k2}(G)$ the k-robust 2-club polytope of G that does not include an empty k-robust 2-club. As any k-robust 2-club is also a k-block, the results of the k-block polytope in Section 3.2.1 (except Theorem 4.10) can be easily extended to the k-robust 2-club polytope by analogously defining minimum a-b separators and essential vertices for a k-robust 2-club.

Given a graph $G = (V, E)$, a subset I of V is a 2-independent set in G if $d_G(i, j) > 2, \forall i, j \in I$.

Theorem 3.2.5 ([6]) *Let I be a maximal 2-independent set in G. Then $\sum_{i \in I} x_i \leq 1$ induces a facet for the 2-club polytope.*

Theorem 3.2.6 *Consider a graph $G = (V, E)$ that satisfies the following condition: For each $v \in V$ there exists a k-robust 2-club S_v such that $v \in S_v$ and $S_v \backslash \{v\}$ is a k-robust 2-club. Let I denote a maximal 2-independent set in G. Then $\sum_{i \in I} x_i \leq 1$ induces a facet of $P_{k2}(G)$ if $I \cap S_v \neq \emptyset$, for each $v \in V \backslash I$.*

Proof Since any inequality valid for the 2-club polytope is also valid for the k-robust 2-club polytope, using Theorem 3.2.5, $\sum_{i \in I} x_i \leq 1$ is valid for $P_{k2}(G)$. To prove that it is a facet, first notice that $dim(P_{k2}(G)) = |V|$ and let $F_I^2 = \{x \in P_{k2}(G) : \sum_{i \in I} x_i = 1\}$ be the face induced. Let there be a valid inequality $\alpha^T x \leq \beta$ such that $F = \{x \in P_{k2}(G) : \alpha^T x = \beta\} \supseteq F_I^2$. For each $v \in V \backslash I$, $x_{S_v}, x_{S_v \backslash \{v\}} \in F_I^2$. Then, $\alpha^T x_{S_v} = \beta$, $\alpha^T x_{S_v \backslash \{v\}} = \beta$ implies that $\alpha_v = 0$, $\forall v \in V \backslash I$. Also, for each $i \in I$, $x_{S_i} \in F_I^2$. Hence for each $i \in I$, $\alpha^T x_{S_i} = \beta$ implies $\beta = \alpha_i$. Hence, $F_I^2 = F$ is a facet. ∎

Theorem 3.2.7 *Given $\omega(G)$, the clique number of a graph $G = (V, E)$, if $k \geq \omega(G)$, then $\sum_{i \in V} x_i \geq k+3$ is valid for $P_{k2}(G)$ and subsumes the inequality $\sum_{i \in V} x_i \geq k+1$.*

Proof Since $k \geq \omega(G)$, $\mu_{k,2}(G) \geq k + 2$. We aim to show that a k-robust 2-club of size $k + 2$ cannot exist. Assume the contrary, i.e., let there be a k-robust 2-club S of cardinality $(k + 2)$. Then, there exists a pair of nonadjacent vertices $a, b \in S$ such that $|N_{G[S]}(a) \cap N_{G[S]}(b)| = k$. Consider any vertex $c \in N_{G[S]}(a) \cap N_{G[S]}(b)$. Since S

is a k-robust 2-club, $|N_{G[S]}(b) \cap N_{G[S]}(c)| = k - 1$. This implies that each neighbor c of b is adjacent to each of the $k - 1$ remaining neighbors of b. Hence, $\{b\} \cup N_{G[S]}(b)$ is a clique of size $k + 1$, which is a contradiction. ∎

Before we discuss the computational experiments for the minimum k-core and k-robust 2-club problems, we generalize the two problems and study their augmentation versions.

3.2.3 Augmentation Problems

Given a proper subset $S \subset V$, the augmentation version of the minimum k-block and k-robust 2-club problems can be formulated as,

$$\min_{x \in X} \sum_{i \in V} x_i \tag{3.17}$$

where, $X = \{x \in \{0,1\}^n : (5.2), (3.3) \ \& \ x_i = 1, \forall i \in S\}$ for the augmentation version of the k-block problem and $X = \{x \in \{0,1\}^n : (3.14), (3.15) \ \& \ x_i = 1, \forall i \in S\}$ for the augmentation version of the k-robust 2-club problem. When $|S| \geq 2$, the lower bound constraint $\sum_{i \in V} x_i \geq k + 1$ can be removed from the formulations.

3.3 Computational Experiments

In this section, computational experiments are performed to evaluate the effectiveness of the considered formulations for k-block and the k-robust 2-club models when solved directly using a standard solver. As mentioned in Section 3.2.1, the number of constraints of type (5.2) in the formulation (3.1)-(3.4) for the minimum k-block problem can be exponential, and hence it is very difficult to enumerate them. Consider the following relation between the minimum k-core, the minimum k-block and the minimum k-robust 2-club numbers of a graph G.

$$m_k(G) \leq \mu_k(G) \leq \mu_{k,2}(G), \tag{3.18}$$

where $m_k(G)$ is the minimum k-core number of a graph. With this relation, one can use the minimum k-robust 2-club and the minimum k-core problems, respectively, as upper and lower bounds for the minimum k-block problem.

We present some computational results for the minimum k-block and the minimum k-robust 2-club problems, respectively obtained by solving the multi-commodity flow formulation (3.5)-(3.12) given in Section 3.2.1, and the IP formulation (3.13)-(3.16) given in Section 3.2.2. All numerical experiments were run on Dell Computer with Intel® Xeon® E5620 2.40 GHz processor and 12GB of RAM, and FICO Xpress-Optimizer 7.7 [24] solver was used. We consider instances from the second and the tenth DIMACS implementation challenges [16, 17], Trick's coloring instances [13], and random instances [19] to perform the experiments. The random instances are based on $G(n, p)$ model [19], where n represents the number of vertices and p denotes the probability that two vertices are adjacent. For example, in Table 3.1, the instance "n50p05" denotes that $n = 50$ and $p = 0.05$. The instances from the tenth DIMACS challenge are typically large and sparse. Since any k-block and k-robust 2-club is also a k-core, reduction in the graph size of these instances can be achieved by finding the largest k-core possible in the given instance and then solving for the k-block and the k-robust 2-club models in the reduced instance.

We present the results of the computational experiments in Tables 3.1–3.4. The columns of these tables show the parameters of the graph instances: graph order, size, density, clique number; number of vertices, edges and the density of the reduced instances and their preprocessing time for reduction, the minimum k-block number

Table 3.1: The minimum k-block number for DIMACS, Trick's coloring and random instances.

Instance	Vertices	Edges	Density %	$\omega(G)$	$\mu_k(G)$ for $k =$			
					$\omega(G)-1$	$\omega(G)$	$\omega(G)+1$	$\omega(G)+2$
anna.col	138	493	5.22	11	11	-	-	-
david.col	87	406	10.85	11	11	-	-	-
huck.col	74	301	11.14	11	11	-	-	-
jean.col	80	254	8.04	10	10	-	-	-
miles.col	128	387	4.76	8	8	-	-	-
myciel3.col	11	20	36.36	2	2	4	8	-
myciel4.col	23	71	28.06	2	2	4	8	11
myciel5.col	47	236	21.83	2	2	4	6	8
mug88_1.col	88	146	3.81	3	3	-	-	-
mug88_25.col	88	146	3.81	3	3	-	-	-
mug100_1.col	100	166	3.35	3	3	-	-	-
mug100_25.col	100	166	3.35	3	3	-	-	-
1-Fullins_3.col	30	100	22.99	3	3	5	7	16
1-Insertions_4.col	67	232	10.49	2	2	4	8	15
2-Fullins_3.col	52	201	15.16	4	4	7	13	-
2-Insertions_3.col	37	72	10.81	2	2	4	16	-
3-Insertions_3.col	56	110	7.14	2	2	4	20	-
queen5_5.col	25	160	53.33	5	5	8	9	12
queen6_6.col	36	290	46.03	6	6	9	12	13
celegans_metabolic.graph	453	2025	1.98	9	9	14	19	-
chesapeake.graph	39	170	22.94	5	5	7	11	-
cond-mat-2005.graph	40421	175691	0.02	30	30	-	-	-
dolphins.graph	62	159	8.41	5	5	-	-	-
email.graph	1133	5451	0.85	12	12	-	-	-
ieeebus.graph	118	179	2.59	4	4	-	-	-
jazz.graph	198	2742	14.06	30	30	-	-	-
karate.graph	34	78	13.90	5	5	-	-	-
kreb.graph	62	153	8.09	6	6	-	-	-
polbooks.graph	105	441	8.08	6	6	9	-	-
power.graph	4941	6594	0.05	6	6	-	-	-
rgg_n_2_17_s0.graph	131072	728753	0.01	15	15	-	-	-
rgg_n_2_19_s0.graph	524288	3269766	0.002	17	17	-	-	-
n50p05	50	62	5.06	3	3	-	-	-
n50p10	50	143	11.67	4	4	10	-	-
n50p15	50	186	15.18	4	4	10	16	-
n50p20	50	252	20.57	4	4	6	10	17
n55p05	55	66	4.44	2	2	4	-	-
n55p10	55	164	11.04	3	3	5	15	-
n55p15	55	209	14.07	3	3	6	11	21
n60p05	60	86	4.86	3	3	-	-	-
n60p10	60	168	9.49	3	3	5	15	-
n60p15	60	277	15.65	4	4	7	14	22
n65p05	65	116	5.58	3	3	14	-	-
n65p10	65	228	10.96	3	3	5	11	32
n70p05	70	120	4.97	3	3	9	-	-
n70p10	70	243	10.06	3	3	5	10	35

Table 3.2: DIMACS, Trick's coloring and random instances running time (in sec) for the minimum k-block problem.

Instance	Reduced			Time for $k =$			
	Vertices	Edges	Density %	$\omega(G)-1$	$\omega(G)$	$\omega(G)+1$	$\omega(G)+2$
anna.col	n\r	n\r	n\r	287.37	235.13	227.02	225.52
david.col	n\r	n\r	n\r	96.70	76.07	74.39	73.69
huck.col	n\r	n\r	n\r	48.59	36.28	35.81	39.46
jean.col	n\r	n\r	n\r	77.64	40.11	38.64	38.37
miles.col	n\r	n\r	n\r	482.69	174.91	157.64	151.19
myciel3.col	n\r	n\r	n\r	0.21	0.44	0.38	0.06
myciel4.col	n\r	n\r	n\r	1.68	3.63	5.88	4.09
myciel5.col	n\r	n\r	n\r	24.72	113.89	83.36	93.68
mug88_1.col	n\r	n\r	n\r	158.72	200.08	32.25	29.36
mug88_25.col	n\r	n\r	n\r	301.65	218.53	35.08	30.31
mug100_1.col	n\r	n\r	n\r	242.03	379.87	47.50	44.80
mug100_25.col	n\r	n\r	n\r	483.23	481.68	51.00	44.97
1-Fullins_3.col	n\r	n\r	n\r	7.58	9.75	9.68	7.72
1-Insertions_4.col	n\r	n\r	n\r	46.41	307.32	378.56	927.21
2-Fullins_3.col	n\r	n\r	n\r	72.25	122.14	102.23	12.41
2-Insertions_3.col	n\r	n\r	n\r	5.65	15.64	21.97	2.45
3-Insertions_3.col	n\r	n\r	n\r	17.48	89.98	285.59	8.79
queen5_5.col	n\r	n\r	n\r	7.01	17.07	17.90	23.31
queen6_6.col	n\r	n\r	n\r	29.61	89.38	288.17	276.83
celegans_metabolic.graph	51	383	30.04	138.34	129.17	85.89	25.81
chesapeake.graph	33	152	28.79	16.45	31.91	20.91	3.63
cond-mat-2005.graph	30	435	100.00	7.28	6.94	7.00	6.99
dolphins.graph	36	109	17.30	72.34	12.01	11.39	10.83
email.graph	12	66	100.00	0.20	0.19	0.21	0.19
ieeebus.graph	4	6	100.00	0.02	0.02	0.02	0.02
jazz.graph	30	435	100.00	7.27	6.94	6.81	6.91
karate.graph	10	25	55.56	0.76	0.07	0.07	0.07
kreb.graph	11	34	61.82	0.52	0.09	0.09	0.09
polbooks.graph	65	300	14.42	113.87	173.78	38.22	32.38
power.graph	12	36	54.55	0.65	0.15	0.14	0.14
rgg_n_2_17_s0.graph	34	262	46.70	32.81	6.78	6.74	6.64
rgg_n_2_19_s0.graph	19	170	99.42	3.08	1.15	1.17	1.19
n50p05	n\r	n\r	n\r	19.54	4.66	4.57	4.48
n50p10	n\r	n\r	n\r	79.27	45.49	9.86	9.44
n50p15	n\r	n\r	n\r	114.80	196.11	131.36	11.62
n50p20	n\r	n\r	n\r	61.49	144.96	543.87	634.01
n55p05	n\r	n\r	n\r	8.96	44.98	6.15	5.98
n55p10	n\r	n\r	n\r	54.40	163.83	230.07	13.40
n55p15	n\r	n\r	n\r	32.49	199.87	382.58	586.41
n60p05	n\r	n\r	n\r	68.32	9.39	8.92	8.65
n60p10	n\r	n\r	n\r	32.18	147.14	88.53	17.13
n60p15	n\r	n\r	n\r	113.84	464.08	1544.58	1602.36
n65p05	n\r	n\r	n\r	88.02	40.95	13.98	13.34
n65p10	n\r	n\r	n\r	50.56	269.19	656.61	462.81
n70p05	n\r	n\r	n\r	66.26	123.52	18.29	16.63
n70p10	n\r	n\r	n\r	105.32	503.13	1718.66	1987.75

for $k = \omega(G)-1, \omega(G), \omega(G)+1, \omega(G)+2$ and their respective computational times, and the minimum k-robust 2-club number for $k = \omega(G)-1, \omega(G)$ and their respective computational times. As the tenth DIMACS instances are large and sparse, the minimum k-block and the minimum k-robust 2-club numbers in these instances are respectively obtained by solving the multi-commodity flow formulation (3.5)-(3.12), and the IP formulation (3.13)-(3.16) on the largest $(\omega(G)-1)$-core in the instances. In the tables, '-', denotes that no solution of size at least $k+1$ exists in the instance, '$*$' denotes that optimal solution was not found within a 3-hour limit, and 'n\r' denotes that the instance was not reduced by preprocessing.

From Table 3.1 and Table 3.3, it is easy to see that though the k-robust 2-club model can be solved in large instances, the results show that for most of the instances considered, finding a k-connected structure with good reachability properties is very restricted as compared to finding a k-connected structure with no constraint on path length. This phenomenon was also observed by [64] when solving for maximum subgraphs with relative vertex connectivity. Also, from Table 3.2 and Table 3.4, it is easy to note that for most of the instances, as k is increased from $\omega(G)-1$, the run time increases whenever a non-trivial, feasible solution is found. This shows that both the problems can be solved very quickly for values of k for which there exists a clique of size $k+1$ in the graph.

From the optimal values in Table 3.1 and Table 3.3, it is easy to see that the k-block and the k-robust 2-club models are tightly-knit. Also, for many of the instances there is no k-robust 2-club of size greater than the maximum clique of that instance. This implies that the minimum k-robust 2-club emulates the clique structure and can be applied for modeling robust structures with good reachability properties. However, this does not imply that modeling cohesive structures as a k-robust 2-club is restrictive in nature. It rather gives an understanding of the kind of highly cohesive clusters one can find in these instances, which in this case are restricted to cliques, and that the minimum k-robust 2-club subgraphs in these instances are highly connected. On the other hand, from Table 3.1 it can be seen that for a number of instances the minimum k-block problem gives a feasible solution when $k \geq \omega(G)$. However, as the value of k increases from $\omega(G) - 1$, the size of the highly connected cluster found also increases. These observations show that the minimum k-robust 2-club and the minimum k-block problems can be used for detecting highly cohesive structures in networks, and also for getting a good upper bound on the clique number of a graph.

3.4 Conclusion

In this chapter, we have considered two concepts, k-block and k-robust 2-club, that model structurally cohesive and robust clusters in networks. We discussed some basic properties of k-blocks and proved that the minimum k-block and the minimum k-robust 2-club problems are APX-hard for $k \geq 3$ and $k \geq 4$ respectively, and also established inapproximability results for the augmentation version of the problems. Integer programming formulations are proposed, and a polyhedral study is conducted for both problems. Sample numerical experiments are reported for both problems. If $k < \omega(G)$, both minimum k-blocks and minimum robust 2-clubs are just cliques of size $k + 1$. We observed that in most of the cases considered there is no k-robust 2-club if $k \geq \omega(G)$ and that infeasibility is established quite quickly. However, for most of the instances, the minimum k-block problem produces a feasible, nontrivial solution for $k \geq \omega(G)$. The models are flexible in the sense that the level of cohesiveness required in the clusters found can actually be specified by choosing the appropriate value of k. The choice of the values of k that would be interesting from a practical perspective depends on availability of a good estimate

Table 3.3: The minimum k-robust 2-club number for DIMACS, Trick's coloring and random instances.

Instance	Vertices	Edges	Density %	$\omega(G)$	$\mu_{k,2}(G)$ for $k =$	
					$\omega(G)-1$	$\omega(G)$
mann_a9.clq	45	918	92.73	16	16	$*$
c-fat200-1.clq	200	1534	7.71	12	12	-
c-fat200-2.clq	200	3235	16.26	24	24	-
c-fat200-5.clq	200	8473	42.58	58	58	-
johnson8-2-4.clq	28	210	55.56	4	4	13
johnson8-4-4.clq	70	1855	76.81	14	14	$*$
johnson16-2-4.clq	120	5460	76.47	8	8	$*$
hamming6-2.clq	64	1824	90.48	32	32	$*$
hamming6-4.clq	64	704	34.92	4	4	10
anna.col	138	493	5.22	11	11	-
david.col	87	406	10.85	11	11	-
huck.col	74	301	11.14	11	11	-
jean.col	80	254	8.04	10	10	-
miles.col	128	387	4.76	8	8	-
myciel3.col	11	20	36.36	2	2	-
myciel4.col	23	71	28.06	2	2	-
myciel5.col	47	236	21.83	2	2	-
mug88_1.col	88	146	3.81	3	3	-
mug88_25.col	88	146	3.81	3	3	-
mug100_1.col	100	166	3.35	3	3	-
mug100_25.col	100	166	3.35	3	3	-
1-Fullins_3.col	30	100	22.99	3	3	-
1-Insertions_4.col	67	232	10.49	2	2	-
2-Fullins_3.col	52	201	15.16	4	4	-
2-Insertions_3.col	37	72	10.81	2	2	-
3-Insertions_3.col	56	110	7.14	2	2	-
queen5_5.col	25	160	53.33	5	5	-
queen6_6.col	36	290	46.03	6	6	-
adjnoun.graph	112	425	6.84	5	5	-
as_22july06.graph	22963	48436	0.02	17	17	-
astro-ph.graph	16706	121251	0.09	57	57	-
celegans_metabolic.graph	453	2025	1.98	9	9	-
chesapeake.graph	39	170	22.94	5	5	-
cnr-2000.graph	325557	2738969	0.01	84	84	-
coAuthorsCiteseer.graph	227320	814134	0.00	87	87	-
coAuthorsDBLP.graph	299067	977676	0.00	115	115	-
cond-mat-2005.graph	40421	175691	0.02	30	30	-
dolphins.graph	62	159	8.41	5	5	-
email.graph	1133	5451	0.85	12	12	-
football.graph	115	613	9.35	9	9	-
ieeebus.graph	118	179	2.59	4	4	-
jazz.graph	198	2742	14.06	30	30	-
karate.graph	34	78	13.90	5	5	-
kreb.graph	62	153	8.09	6	6	-
memplus.graph	17758	54196	0.03	97	97	-
PGPgiantcompo.graph	10680	24316	0.04	25	25	28
polblogs.graph	1490	16715	1.51	20	20	23
polbooks.graph	105	441	8.08	6	6	-
power.graph	4941	6594	0.05	6	6	-
rgg_n_2_17_s0.graph	131072	728753	0.008	15	15	-
rgg_n_2_19_s0.graph	524288	3269766	0.002	18	18	-
rgg_n_2_20_s0.graph	1048576	6891620	0.002	17	17	-
n50p10	50	143	11.67	4	4	-
n50p15	50	186	15.18	4	4	-
n50p20	50	252	20.57	4	4	-
n55p10	55	164	11.04	3	3	-
n55p15	55	209	14.07	3	3	-
n60p10	60	168	9.49	3	3	-
n60p15	60	277	15.65	4	4	-
n65p05	65	116	5.58	3	3	-
n65p10	65	228	10.96	3	3	-
n70p05	70	120	4.97	3	3	-
n70p10	70	243	10.06	3	3	-

Table 3.4: DIMACS, Trick's coloring and random instances running time (in sec) for the minimum k-robust 2-club problem.

Instance	Reduced			Preprocessing	Time for $k =$	
	Vertices	Edges	Density %	time	$\omega(G)-1$	$\omega(G)$
mann_a9.clq	n\r	n\r	n\r	n\r	1.45	>10800
c-fat200-1.clq	n\r	n\r	n\r	n\r	5.78	8.75
c-fat200-2.clq	n\r	n\r	n\r	n\r	1.84	2.36
c-fat200-5.clq	n\r	n\r	n\r	n\r	8.81	3.23
johnson8-2-4.clq	n\r	n\r	n\r	n\r	0.18	0.47
johnson8-4-4.clq	n\r	n\r	n\r	n\r	247.17	>10800
johnson16-2-4.clq	n\r	n\r	n\r	n\r	14.80	>10800
hamming6-2.clq	n\r	n\r	n\r	n\r	2434.85	>10800
hamming6-4.clq	n\r	n\r	n\r	n\r	0.55	0.86
anna.col	n\r	n\r	n\r	n\r	27.17	26.83
david.col	n\r	n\r	n\r	n\r	9.08	9.09
huck.col	n\r	n\r	n\r	n\r	5.08	5.05
jean.col	n\r	n\r	n\r	n\r	5.13	5.12
miles.col	n\r	n\r	n\r	n\r	18.35	20.07
myciel3.col	n\r	n\r	n\r	n\r	0.11	0.03
myciel4.col	n\r	n\r	n\r	n\r	0.22	0.15
myciel5.col	n\r	n\r	n\r	n\r	1.75	1.58
mug88_1.col	n\r	n\r	n\r	n\r	4.10	4.08
mug88_25.col	n\r	n\r	n\r	n\r	3.93	4.09
mug100_1.col	n\r	n\r	n\r	n\r	5.89	5.86
mug100_25.col	n\r	n\r	n\r	n\r	6.16	33.19
1-Fullins_3.col	n\r	n\r	n\r	n\r	0.32	0.31
1-Insertions_4.col	n\r	n\r	n\r	n\r	3.31	3.17
2-Fullins_3.col	n\r	n\r	n\r	n\r	1.69	1.69
2-Insertions_3.col	n\r	n\r	n\r	n\r	0.37	0.36
3-Insertions_3.col	n\r	n\r	n\r	n\r	1.23	1.24
queen5_5.col	n\r	n\r	n\r	n\r	0.44	0.46
queen6_6.col	n\r	n\r	n\r	n\r	1.15	1.11
adjnoun.graph	79	359	11.65	0.002	0.11	0.09
as_22july06.graph	144	2758	26.79	0.090	4.43	9.99
astro-ph.graph	57	1596	100.00	0.012	0.10	0.09
celegans_metabolic.graph	51	383	30.04	0.001	0.07	0.06
chesapeake.graph	33	152	28.79	0.001	0.52	0.53
cnr-2000.graph	86	3652	99.92	3.943	18.84	0.98
coAuthorsCiteseer.graph	87	3741	100.00	0.074	0.29	0.29
coAuthorsDBLP.graph	115	6555	100.00	0.115	0.65	0.69
cond-mat-2005.graph	30	435	100.00	0.019	0.03	0.03
dolphins.graph	36	109	17.30	0.001	0.06	0.03
email.graph	12	66	100.00	0.002	0.02	0.02
football.graph	114	606	9.41	0.001	0.21	0.16
ieeebus.graph	4	6	100.00	0.001	0.02	0.02
jazz.graph	30	435	100.00	0.001	0.04	0.03
karate.graph	10	25	55.56	0.001	0.02	0.02
kreb.graph	11	34	61.82	0.002	0.05	0.03
memplus.graph	97	4656	100.00	0.005	0.39	0.40
PGPgiantcompo.graph	126	2326	29.54	0.003	7.14	6.89
polblogs.graph	438	11495	12.01	0.002	84.24	126.98
polbooks.graph	65	300	14.42	0.001	0.08	0.07
power.graph	12	36	54.55	0.002	0.02	0.02
rgg_n_2_17_s0.graph	34	262	46.70	0.044	0.05	0.03
rgg_n_2_19_s0.graph	19	170	99.42	0.199	0.04	0.03
rgg_n_2_20_s0.graph	172	1624	11.04	0.440	1.05	2.23
n50p10	n\r	n\r	n\r	n\r	1.23	1.22
n50p15	n\r	n\r	n\r	n\r	1.51	1.55
n50p20	n\r	n\r	n\r	n\r	1.99	2.03
n55p10	n\r	n\r	n\r	n\r	1.65	1.59
n55p15	n\r	n\r	n\r	n\r	1.99	1.95
n60p10	n\r	n\r	n\r	n\r	2.09	2.02
n60p15	n\r	n\r	n\r	n\r	3.19	3.22
n65p05	n\r	n\r	n\r	n\r	1.83	1.79
n65p10	n\r	n\r	n\r	n\r	3.18	3.19
n70p05	n\r	n\r	n\r	n\r	2.21	2.14
n70p10	n\r	n\r	n\r	n\r	3.96	3.88

of $\omega(G)$, which can be obtained efficiently even for very large networks [65]. On the one hand, the values of k for which the considered problems are infeasible (which can typically be established quickly) yield an upper bound k on the clique number and on the other hand, an optimal solution of size $> k+1$ to one of our problems, whenever exists, not only provides the upper bound k on the clique number, but also yields a structurally cohesive cluster that is nontrivial. In the latter case, the problems appear to be harder to solve in practice. An interesting avenue for further research would be to improve the inapproximability results for the problems considered and develop a cutting plane or a branch and cut algorithm using the established valid inequalities.

4. THE MAXIMUM S-STABLE CLUSTER PROBLEM

This chapter will focus on the new clique relaxation model introduced in Chapter 2, namely, s-stable cluster, for describing stable clusters. Here, we will discuss some local optimality conditions for the maximum s-stable cluster problem, which is an extension of the work done on the independent set problem by Nemhauser and Trotter [45], study the s-stable cluster polytope in detail, and introduce two exact algorithms for solving the maximum s-stable cluster problem.

4.1 Computational Complexity

Given a simple graph $G = (V, E)$ and positive integers s, c, s-STABLE CLUSTER problem asks, does there exists an s-stable cluster of size at least c in G ?

The s-stable cluster property is non-trivial, interesting and hereditary on induced subgraphs, hence using the result by Yannakakis [67], we have the following result.

Theorem 4.1.1 s-STABLE CLUSTER *is NP-complete for any fixed* $s \geq 1$.

We show this result explicitly in a restricted class of graphs called claw-free graphs. A graph is claw-free if it does not contain the complete bipartite graph $K_{1,3}$ ("claw") as an induced subgraph. We show that s-STABLE CLUSTER is NP-complete on claw-free graphs using a reduction from CLIQUE, which is NP-complete on claw-free graphs [21].

Proposition 4.1.2 s-STABLE CLUSTER *is NP-complete on claw-free graphs.*

Proof Clearly, s-STABLE CLUSTER is in NP for constant s. For any fixed $s \geq 2$, we give a reduction from CLIQUE on claw-free graphs. Given a claw-free graph $G = (V, E)$ on $|V| = n$ vertices, we construct a claw-free graph $G' = (V', E')$, which is an instance of s-STABLE CLUSTER problem, as follows. Let $G_i = (V_i, E_i)$, $i = 1, \ldots, s-1$, be copies of the complete graph K_{n+1} on $n+1$ vertices, and let G' be the disjoint union of G and $G_1, \ldots, G_{s-1}$.

We now prove that the maximum s-stable clusters in G' correspond to the maximum cliques in G and vice-versa. Firstly, observe that the largest s-stable cluster H in G' must include the set $\cup_{i=1}^{s-1} V_i$, as G_i is complete for each $i = 1, \ldots, s-1$. This implies that if $S = H \cap V$ is not empty then S must be a clique in G, as otherwise the number of independent vertices in H would be greater than $s+1$. Using these two observations, it is easy to see that any maximum s-stable cluster in G' corresponds to a maximum clique in G and vice-versa. ∎

4.2 Local Optimality Conditions

Nemhauser and Trotter [45] studied the optimality conditions of the vertex packing problem, which is nothing else but the maximum clique (1-stable cluster) problem in the complement graph. In this section, we analyze whether these optimality conditions can be extended to s-stable clusters when $s \geq 2$.

First, consider the following notations which will be used in this section. Let $\mathcal{P}_G^s$ denote the family of all s-stable clusters in G, and let $\mathcal{I}_G^k$ denote the family of all stable sets of cardinality k in G. For any $P \in \mathcal{P}_G^s$ and $k = 1, 2, \ldots, s$, define $\bar{N}_k(P)$ as follows:

$$\bar{N}_k(P) = \{v \in U : U \in \mathcal{I}_G^k \text{ and } P \cup U \notin \mathcal{P}_G^s\},$$

and let $\bar{N}_0(P) = \emptyset$. Define $\bar{N}(P) = \cup_{r=1}^{s} \bar{N}_r(P)$. Let $c(P)$ denote the cardinality of set P in G. Next, we generalize the notion of augmenting subsets introduced in [45] for vertex packing to s-stable clusters. For $P \in \mathcal{P}_G^s$, let $\bar{P} = V \backslash P$.

Definition 18 *Given a set $P \in \mathcal{P}_G^s$, a subset $I \subseteq \bar{P}$ is called an augmenting subset to P, if $(P \cup I)\backslash R \in \mathcal{P}_G^s$ and $c((P \cup I)\backslash R)) > c(P)$ for some subset $R \subseteq P$.*

It is easy to see that P is an optimum s-stable cluster if and only if there exists no augmenting subset to P in G. The augmenting subset admits the following equivalent characterization.

Proposition 4.2.1 *Given a set $P \in \mathcal{P}_G^s$, a subset $I \subseteq \bar{P}$ is an augmenting subset to P if and only if $I \in \mathcal{P}_G^s$, and there exists $R \subseteq (P \cap \bar{N}(I))$ such that $(P \cup I)\backslash R \in \mathcal{P}_G^s$ and $c(I) > c(R)$.*

Proof First, note that for any subset $S \subseteq V\backslash\bar{N}(I)$ such that $S \in \mathcal{P}_G^s$, we have $I \cup S \in \mathcal{P}_G^s$. Hence, if $R \subseteq P$ is such that $I \cup R \notin \mathcal{P}_G^s$, then $R \subseteq \bar{N}(I) \cap P$. Hence, if I is an augmenting subset to P, then by Definition 18, there exists $R \subseteq P$ such that $(P \cup I)\backslash R) \in \mathcal{P}_G^s$, and $c((P \cup I)\backslash R) > c(P)$, implying that $I \in \mathcal{P}_G^s$ and $c(I) > c(R)$ for some $R \subseteq \bar{N}(I) \cap P$. To show the other direction, it is easy to see that if there exists $R \subseteq (P \cap \bar{N}(I))$ such that $(P \cup I)\backslash R \in \mathcal{P}_G^s$ and $c(I) > c(R)$, then $c((P \cup I)\backslash R)) > c(P)$, and the result follows. ∎

Given $I \subseteq \bar{P}$, let $P(I)$ be the smallest subset of $P \cap \bar{N}(I)$ such that $(P \cup I)\backslash P(I) \in \mathcal{P}_G^s$. Then, $c(I) > c(R) \geq c(P(I))$.

When $s = 1$, given $P, I \in \mathcal{P}_G^s$ in G such that $P \cap I = \emptyset$, checking if I is an augmenting subset to P is easy, as $P(I) = P \cap \bar{N}(I)$ such that $(P \cup I)\backslash P(I) \in \mathcal{P}_G^s$, can be found in polynomial time. However, when $s \geq 2$, $P(I)$ is a subset of $P \cap \bar{N}(I)$, which gives way to two problems, namely, the hardness of finding $P(I)$ and checking if I is an augmenting subset to P.

Theorem 4.2.2 *Given a graph $G = (V, E)$ and sets $P, I \in \mathcal{P}_G^s$ such that $P \cap I = \emptyset$, finding the smallest set $P(I)$, such that $(P \cup I)\backslash P(I) \in \mathcal{P}_G^s$ is NP-hard.*

Proof For any $s \geq 2$, we give a reduction from an instance of CLIQUE on s-stable clusters, which can be proved NP-complete using a reduction from MAX 2-SAT, to an instance $G'_s = (V'_s, E'_s), P, I$ of the augmentation subset problem. The construction of $G'_s = (V'_s, E'_s)$ is as follows. Let $G_i = (V_i, E_i)$, $i = 1, \ldots, s-1$, be copies of the complete graph K_{n+1} on $n+1$ vertices, where $V_i = \{0^i, 1^i, \ldots, n^i\}$, $\forall i = 1, \ldots, (s-1)$. Let $V'_s = (\cup_{i=1}^{s-1} V_i) \cup V \cup \{u_0\}$ and $E'_s = (\cup_{i=1}^{s-1} E_i) \cup E \cup E_{u_0} \cup E'_{u_0}$, where $E_{u_0} = \{(0^i, u_0) : i = 1, \ldots, s-1\}$ and $E'_{u_0} = \{(v, u_0) : \forall v \in V\}$. This completes the construction of G'_s with $P = V$ and $I = V'_s \backslash V$. It is easy to see that $P, I \in P^s_{G'_s}$. We now prove that a clique of the largest cardinality in G correspond to the smallest set $P(I)$ such that $(P \cup I) \backslash P(I) \in \mathcal{P}^s_G$, and vice-versa.

First, note that the sets $\{i^1, i^2, \ldots, i^{s-1}\}$, for each $i = 0, 1, \ldots, n$ are stable sets of size $s-1$, that is $\alpha(I) = s-1$. Hence, any set $P' \subseteq P$ such that $P' \cup I \in \mathcal{P}^s_G$, must be a clique as otherwise, for any $u, v \in P'$ such that $(u, v) \notin E$, the set $\{u, v, i^1, i^2, \ldots, i^{s-1}\}$ will be a stable set of size $s+1$. This implies that a clique of maximum cardinality P' in G will imply the smallest set $P(I) = P \backslash P'$ such that $P' \cup I \in \mathcal{P}^s_G$. Similarly, if $P(I)$ is the smallest set such that $(P \cup I) \backslash P(I) \in \mathcal{P}^s_G$, then $P \backslash P(I)$ is a maximum clique in G. ∎

Theorem 4.2.3 *Given a graph $G = (V, E)$ and sets $P, I \in \mathcal{P}^s_G$ such that $P \cap I = \emptyset$, checking if I is an augmenting subset to P is NP-hard, for any fixed $s \geq 3$.*

Proof We give a reduction from an instance ϕ of 3-SAT to an instance $G = (V, E), P, I$ of the augmentation subset problem such that ϕ is satisfiable if and only if I is an augmenting subset to P. The construction of $G = (V, E)$ is as follows.

Let $\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be a 3-CNF with m clauses, and for $r = 1, \ldots, m$, C_r has exactly 3 distinct literals l^r_1, l^r_2, l^r_3. Let $G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_{s-1} = (V_{s-1}, E_{s-1})$ be such that for each clause $C_r = (l^r_1, l^r_2, l^r_3)$ in ϕ, we place three vertices

$v_1^{r(p)}, v_2^{r(p)}, v_3^{r(p)}$ into V_p, for $p = 1, 2, \ldots, (s-1)$, and we put an edge between 2 vertices $v_i^{r(p)}$ and $v_j^{s(p)}$ in V_p if the following conditions hold:

- the literals corresponding to $v_i^{r(p)}$ and $v_j^{s(p)}$ are in different clauses, that is $r \neq s$.
- the literals corresponding to $v_i^{r(p)}$ and $v_j^{s(p)}$ are consistent, that is l_i^r and l_j^s are not negations of each other.

Let $\tilde{E}$ be the edge set between the sets V_a and V_b, for all distinct pairs $a, b \in \{1, 2, \ldots, s-1\}$. Then, for any two vertices $v_i^{r(a)} \in V_a$, $v_j^{s(b)} \in V_b$, $(v_i^{r(a)}, v_j^{s(b)}) \in \tilde{E}$, if the following conditions hold:

- when the literals corresponding to $v_i^{r(a)}$ and $v_j^{s(b)}$ belong to the same clause, that is when $r = s$, the literals are the same, that is $i = j$.
- when the literals corresponding to $v_i^{r(a)}$ and $v_j^{s(b)}$ doesn't belong to the same clause, that is when $r \neq s$, they are not negations of each other.

Let the graph $G' = (V', E')$ be such that $V' = \cup_{i=1}^{s-1} V_i$, and $E' = (\cup_{i=1}^{s-1} E_i) \cup \tilde{E}$. Then, it is easy to see that for any $v_i^{r(a)}, v_j^{s(b)} \in V'$ such that $(v_i^{r(a)}, v_j^{s(b)}) \notin E'$, either the literals corresponding to both the vertices belong to the same clause when $a = b$, or they are not consistent. Hence, for any three vertices $v_i^{r(a)}, v_j^{s(b)}, v_k^{t(c)} \in V'$ we have, $N_{G'}[v_i^{r(a)}] \cup N_{G'}[v_j^{s(b)}] \cup N_{G'}[v_k^{t(c)}] = V'$, implying that $\alpha(G') = 3$. Let $G_1'', G_2'', \ldots, G_{s-1}''$ be $(s-1)$ identical copies of a complete graph on $2m$ vertices, where m is the number of clauses in the 3-sat instance ϕ. Denote the vertex set and edge set of the r^{th} such copy respectively by V_r'' and E_r'', where $V_r'' = \{0_r, 1_r, \ldots, (2m-1)_r\}$, and $E_r'' = \{(i_r, j_r) : j > i, i = 0, 1, \ldots, 2m-1\}$. Let the graph $\acute{G} = (\acute{V}, \acute{E})$ be such that $\acute{V} = (\cup_{i=1}^{s-1} V_i'') \cup \{u_0\}$, and $\acute{E} = (\cup_{i=1}^{s-1} E_i'') \cup E_{u_0}$, where $E_{u_0} = \{(u_0, 0_r) : r = 1, 2, \ldots s-1\}$. It is easy to see that $\alpha(\acute{G}) = s$, since the set $\{u_0, i_1, i_2, \ldots, i_{s-1}\}$, for $i = 0, 1, \ldots, 2m-1$ is an independent set of size s. Finally, we have $G = (V, E)$, with

$V = V' \cup \acute{V}$, and $E = E' \cup \acute{E} \cup \hat{E}$, where $\hat{E} = \{(u_0, v) : \forall v \in V'\}$. This completes the construction of G, with $P = V'$ and $I = \acute{V}$ such that $c(I) = 2m(s-1)+1$, and $c(P) = 3m(s-1)$. Then, we have $P, I \in \mathcal{P}^s_G$, and $P \cap I = \emptyset$.

First note that I is an augmenting subset to P if and only if, for any subset $P' \subseteq P$ such that $P' \cup I \in \mathcal{P}^s_G$, P' is a clique and $c(P \backslash P') \leq 2m(s-1)$.

Suppose ϕ has a satisfying assignment, then each clause C_r contains at least one literal l^r_i that is assigned 1, and each such literal corresponds to a vertex $v_i^{r(p)}$, for $p = 1, \ldots, s-1$. Picking one such true literal from each clause yields a set P' of $m(s-1)$ vertices. Then by the construction of G, we know that P' is a clique. Hence, $I \cup P' \in \mathcal{P}^s_G$, and $c(I) > c(P \backslash P')$, implying that I is an augmenting subset to P.

Conversely, suppose I is an augmenting subset to P, then we have the smallest set $P(I)$ such that $(P \cup I) \backslash P(I) \in \mathcal{P}^s_G$ and $c(P(I)) \leq 2m(s-1)$. This implies that, there exists a set $P' = P \backslash P(I)$, which must be a clique of size at least $m(s-1)$. Note that no edges in G' connect vertices that correspond to literals that are negations of each other, or to literals that are in the same clause, unless the vertices are in distinct sets $V_a, V_b \subseteq V$, $a \neq b$. Hence the size of the maximum clique in G' can be at most $m(s-1)$, with m vertices each from G_i, for $i = 1, \ldots, s-1$. Hence, since I is an augmenting subset to P, we have a clique P' of size $m(s-1)$. By taking a copy G_p for some $p \in \{1, 2, \ldots, s-1\}$, and by assigning 1 to literal l^r_i such that $v_i^{r(p)} \in V_p$ is in the clique, we ensure that each clause is satisfied in the given 3-SAT instance ϕ, and hence ϕ is satisfied. ∎

The characterization of an augmenting subset yielded a local sufficient condition for optimality for the vertex packing problem [45], which can be generalized as follows.

Theorem 4.2.4 *If P is an optimum s-stable cluster in $\hat{G}$ which is the graph induced by $P \cup \bar{N}(P)$, then $P \subseteq P^*$, where P^* is an optimum s-stable cluster in G.*

Proof Let P' be an optimum stable set in G', the graph induced by $V' = V \backslash (P \cup \bar{N}(P))$, and let $P^* = P \cup P'$, and $\bar{P}' = V' \backslash P'$. Define $\bar{P}^* = \bar{N}(P) \cup \bar{P}'$ and let $I \subseteq \bar{P}^*$ such that $I \in \mathcal{P}_G^s$. Define $I = I_1 \cup I_2$, where $I_1 = I \cap \bar{N}(P)$, $I_2 = I \cap \bar{P}'$, and $I_1 \cap I_2 = \emptyset$. Consider a smallest subset $P^*(I) \subseteq P^* \cap \bar{N}(I)$, such that $(P^* \cup I) \backslash P^*(I) \in \mathcal{P}_G^s$. Then, since $P \cap \bar{N}(I_2) = \emptyset$ we have,

$$
\begin{aligned}
P^* \cap \bar{N}(I) &= (P \cup P') \cap \bar{N}(I) \\
&= (P \cap \bar{N}(I)) \cup (P' \cap \bar{N}(I)).
\end{aligned}
$$

Note that, $P \cup I_2 \in \mathcal{P}_G^s$, and hence, $P \cap \bar{N}(I_2) = \emptyset$. Then, we claim that $P \cap \bar{N}(I) = P \cap \bar{N}(I_1)$. Suppose not, then there exists a set of vertices $R \subseteq P$ and $H \subseteq I$, such that $R \cup H \notin \mathcal{P}_G^s$, and $H \cap I_2 \neq \emptyset$. Then, $H \subseteq \bar{N}(P)$ implying that $I_2 \cap \bar{N}(P) \neq \emptyset$, and hence $P \cup I_2 \notin \mathcal{P}_G^s$ which is not true. Then,

$$
\begin{aligned}
P^* \cap \bar{N}(I) &= (P \cap \bar{N}(I_1)) \cup (P' \cap \bar{N}(I_1)) \cup (P' \cap \bar{N}(I_2)) \\
&\cup ((P' \cap \bar{N}(I)) \backslash ((P' \cap \bar{N}(I_1)) \cup (P' \cap \bar{N}(I_2))).
\end{aligned}
$$

Let $P^*(I) = A_1 \cup A_2 \cup A_3 \cup A_4$, where $A_1 \subseteq P \cap \bar{N}(I_1)$, $A_2 \subseteq P' \cap (\bar{N}(I_1) \backslash \bar{N}(I_2))$, $A_3 \subseteq P' \cap \bar{N}(I_2)$, and $A_4 \subseteq (P' \cap \bar{N}(I)) \backslash ((P' \cap \bar{N}(I_1)) \cup (P' \cap \bar{N}(I_2))$. Then, $A_2 \cap P = \emptyset, A_3 \cap P = \emptyset$. This implies that, since $(P^* \cup I) \backslash P^*(I) \in \mathcal{P}_G^s$, we have $(P \cup I_1) \backslash A_1 \subseteq (P^* \cup I) \backslash P^*(I)$, and $(P \cup I_1) \backslash A_1 \in \mathcal{P}_G^s$. Hence, due to optimality of P in $P \cup \bar{N}(P)$, and since $I_1 \subseteq \bar{N}(P)$, using Proposition 4.2.1 we have,

$$
c(I_1) \leq c(A_1) \leq c(A_1 \cup A_2). \tag{4.1}
$$

Similarly, since $A_1 \cap P' = \emptyset, A_2 \cap \bar{N}(I_2) = \emptyset$, and $(P' \cup I_2) \backslash A_3 \subseteq (P^* \cup I) \backslash P^*(I)$, we have $(P' \cup I_2) \backslash A_3 \in \mathcal{P}_G^s$. Hence, due to the optimality of P' in $P' \cup \bar{P}'$, and since $I_2 \subseteq \bar{P}'$, we have

$$c(I_2) \leq c(A_3) \leq c(A_3 \cup A_4). \tag{4.2}$$

Summing up inequalities (4.1) and (4.2), we have

$$c(I) = c(I_1) + c(I_2) \leq c(A_1 \cup A_2) + c(A_3 \cup A_4),$$

and since A_1, A_2, A_3 and A_4 are disjoint sets,

$$c(I) \leq c(A_1 \cup A_2 \cup A_3 \cup A_4) = c(P^*(I)).$$

Since $P^*(I)$ is the minimum possible set of vertices in $P^* \cap \bar{N}(I)$ such that $(P^* \cup I) \backslash P^*(I) \in \mathcal{P}_G^s$, no such augmenting set I of P^* exists, and hence, using proposition 4.2.1, P^* is an optimum s-stable cluster in G and the result follows. ∎

4.3 The s-Stable Cluster Polytope

Consider a graph $G = (V, E)$ with $|V| = n$. For any set $A \subseteq V$, let x_A denote the incidence vector of A. Let $\mathcal{I}_{s+1}$ denote the family of all independent sets of size $s+1$ in G and A be a $|\mathcal{I}_{s+1}| \times n$ matrix, whose rows correspond to independent sets of size $s+1$. Then, the maximum s-stable cluster can be formulated as:

$$\omega_s(G) = \max \quad \sum_{i \in V} x_i \tag{4.3}$$

$$\text{subject to} \quad Ax \leq s \tag{4.4}$$

$$x \in \{0, 1\}^n \tag{4.5}$$

Constraint (5.2) ensures that the subgraph induced by the subset of vertices $\{i : x_i = 1\}$ is an s-stable cluster. Let $LP_s(G) = \{x \in R^n : Ax \leq s, x \leq 1\}$ be the feasible region of the linear programming(LP) relaxation of the above formulation given by

$$\max_{x \in LP_G^s} \sum_{i \in V} x_i. \tag{4.6}$$

If x is an extreme point of LP_G^s, then the values of $x_j, \forall j \in V$ includes $0, 1$ or $\frac{s}{s+1}$. Then we have the following generalization of the result given in [45] for vertex packing.

Lemma 4.3.1 *If the optimal solution x to (4.6) is given by $x_j = \frac{s}{s+1}, \forall j \in V$, then it is the unique solution to (4.6) if and only if $\forall P \in P_G^s$,*

$$c(P) < \sum_{r=1}^{s} \frac{s-(r-1)}{r} c(\bar{N}_r(P) \setminus \cup_{k=0}^{r-1} \bar{N}_k(P)). \tag{4.7}$$

Proof Suppose there exists $P \in P_G^s$ such that,

$$\begin{aligned}
& c(P) \geq \sum_{r=1}^{s} \frac{s-(r-1)}{r} c(\bar{N}_r(P) \setminus \cup_{k=0}^{r-1} \bar{N}_k(P)) \\
\iff \quad & (1 - \frac{s}{s+1}) c(P) \geq \sum_{r=1}^{s} (\frac{s}{s+1} - \frac{r-1}{r}) c(\bar{N}_r(P) \setminus \cup_{k=0}^{r-1} \bar{N}_k(P)) \\
\iff \quad & c(P) + \sum_{r=1}^{s} \frac{r-1}{r} c(\bar{N}_r(P) \setminus \cup_{k=0}^{r-1} \bar{N}_k(P)) \geq \frac{s}{s+1} (c(P) + \sum_{r=1}^{s} c(\bar{N}_r(P) \setminus \cup_{k=0}^{r-1} \bar{N}_k(P))) \\
\iff \quad & c(P) + \sum_{r=1}^{s} \frac{r-1}{r} c(\bar{N}_r(P) \setminus \cup_{k=0}^{r-1} \bar{N}_k(P)) \geq \frac{s}{s+1} [c(V) - c(V \setminus (P \cup \bar{N}(P)))] \\
\iff \quad & c(P) + \sum_{r=1}^{s} \frac{r-1}{r} c(\bar{N}_r(P) \setminus \cup_{k=0}^{r-1} \bar{N}_k(P)) + \frac{s}{s+1} c(V \setminus (P \cup \bar{N}(P))) \geq \frac{s}{s+1} c(V) \\
\iff \quad & cx' \geq cx, \text{for some feasible solution } x',
\end{aligned}$$

where

$$x'_j = \begin{cases} 1 & j \in P \\ \frac{r-1}{r} & j \in \bar{N}_r(P) \setminus \cup_{k=0}^{r-1} \bar{N}_k(P), \forall r = 1, .., s \\ x_j & j \in V \setminus (P \cup (\cup_{r=1}^{s} \bar{N}_r(P))), \end{cases} \tag{4.8}$$

implying that x is not the unique solution. ∎

Note that, if P is maximal in Lemma 4.3.1, then $\bar{N}(P) = \bar{N}_1(P)$.

The s-stable cluster polytope of G, denoted by $P_s(G)$, is the convex hull of the incidence vectors of all the s-stable clusters in G. The following theorem establishes the basic properties of $P_s(G)$.

Theorem 4.3.2 (Full dimension) *Let $P_s(G)$ denote the s-stable cluster polytope of G. Then,*

1. *$dim(P_s(G)) = |V|$.*

2. *$x_v \geq 0$ induces a facet of $P_s(G)$ for every $v \in V$.*

3. *For $s \geq 2$, $x_v \leq 1$ induces a facet of $P_s(G)$ for every $v \in V$.*

Proof Let e_i be the unit vector with i^{th} component 1 and the rest zero.

1. The points $0, e_i, \forall i \in V$ are $|V| + 1$ affinely independent points in $P_s(G)$, and hence, $dim(P_s(G)) = |V|$.

2. Let $F_v^0 = \{x \in P_s(G) : x_v = 0\}$. Then $0, e_w, \forall w \in V - \{v\}$ are $|V|$ affinely independent points in F_v^0. Hence, $dim(F_v^0) = |V| - 1$ and it is a facet.

3. For $s \geq 2$, let $F_v^1 = \{x \in P_s(G) : x_v = 1\}$. Then $e_v, e_v + e_w, \forall w \in V - \{v\}$ are $|V|$ affinely independent points in F_v^1. Hence, $dim(F_v^1) = |V| - 1$ and it is a facet. ∎

Lemma 4.3.3 *Let I denote an independent set of size at least $s + 1$. Then the inequality $\sum_{v \in I} x_i \leq s$ induces a facet of $P_s(G)$ if and only if I is maximal.*

Proof Let I be any maximal independent set in G, and let $F_I = \{x \in P_s(G) : \sum_{v \in I} x_v = s\}$ denote the face induced. Let there be a valid inequality $\alpha x \leq \beta$ such that $F = \{x \in P_s(G) : \alpha x = \beta\} \supseteq F_I$. For every $S \subseteq I$ such that $|S| = s$, $x_S \in F$ and hence, $\alpha x_S = \beta$. This implies that $\alpha_v = \alpha_w = \alpha, \forall v, w \in I$, and $\beta = s\alpha$. Since I is maximal, for each $v \in V \backslash I$ let $S_v \subseteq I$ be such that $|S_v| = s$, and there exists a vertex $u \in S_v$ such that $(u, v) \in E$. Then, we have $x_{S_v \cup \{v\}}, x_{S_v}$ in F_I. This implies that $\alpha_v = 0, \forall v \in V \backslash I$, and $\sum_{v \in I} \alpha x_v = s\alpha$. Hence, F_I^3 is a maximal face, and hence a facet.

For the other direction, let I not be maximal and $\sum_{v \in I} x_v \leq s$ induces a facet of $P_s(G)$. Then, there exists an independent set I' such that $I \subset I'$, and $\sum_{v \in I'} x_v \leq s$ is valid for $P_s(G)$ and subsumes $\sum_{v \in I} x_v \leq s$ implying that it cannot be a maximal face. ∎

Let $\mathcal{I}$ denote the family of all maximal independent sets in G, and B be a $|\mathcal{I}| \times n$ matrix, whose rows correspond to maximal independent sets in G. Then, we can alternatively formulate the maximum s-stable cluster problem as:

$$\omega_s(G) = \max \quad \sum_{i \in V} x_i \tag{4.9}$$

$$\text{subject to} \quad Bx \leq s \tag{4.10}$$

$$x \in \{0, 1\}^{|V|} \tag{4.11}$$

Note that B can be restricted to contain maximal independent sets of size at least $s + 1$. In general, the number of maximal independent sets in a given graph may be exponential and hence, solving this formulation is difficult. In fact, the maximal independent set formulation for the maximum clique problem is proved to be hard to solve. Next, we prove some classes of facets for the s-stable cluster polytope.

4.3.1 Hereditary Property

Let Π be any hereditary property of graphs. Given a graph $G = (V, E)$, let the edge set E' be such that $G = (V, E')$ is a complete graph, and let $\theta_\Pi(G)$ denote the cardinality of the largest induced subgraph in G satisfying Π. Denote by $P_\Pi(G)$ the convex hull of all the incidence vectors satisfying property Π in G.

For any given hereditary property Π, an edge $e \in E' \backslash E$ is called an essential edge of G if $\theta_\Pi(G+e) = \theta_\Pi(G)+1$. The following theorem is a variant and a generalization of a result and proof given by Chvátal [11] for the stable set polytope, which was also proved to be true for the co-s-plex polytope [41].

Theorem 4.3.4 *Let $G = (V, E)$ be a graph and $E^* \subseteq E' \backslash E$ be the set of essential edges of G for a given property Π. If $G^* = (V, E^*)$ is connected then the inequality*

$$\sum_{v \in V} x_v \leq \theta_\Pi(G)$$

induces a facet of $P_\Pi(G)$.

Proof We show that the proof given in [41] works for all graphs satisfying the given conditions. Let G satisfy the given conditions and let $|V| = n$. Let $P_\Pi(G) = \{x \in R^n_+ | \sum_{v \in V} \alpha_{iv} x_v \leq b_i, i \in I\}$, where the inequalities are all the facets excluding the non-negativity constraints. We consider the dual programs.

$$\max\{\sum_{v\in V} x_v | x \geq 0, \sum_{v\in V} \alpha_{iv} x_v \leq b_i, i \in I\}$$
$$\min\{\sum_{i\in I} \lambda_i b_i | \lambda \geq 0, \sum_{i\in I} \alpha_{iv} \lambda_i \geq 1, v \in V\}.$$

An optimal solution λ^* to the minimization problem above satisfies $\sum_{i\in I} \lambda_i b_i = \theta_\Pi(G)$. Let there be a vertex $k \in V$ and by dual feasibility there exists $j \in I$ such that $\lambda_j^*, \alpha_{jk} > 0$. Let $(u, w) \in E^*$. Then there exists incidence vectors $y \in P_\Pi(G)$ and $z \in P_\Pi(G)$ such that,

$$\sum_{v\in V} y_v = \sum_{v\in V} z_v = \theta_\Pi(G) \tag{4.12}$$

with,

$$y_u = z_w = 1, y_w = z_u = 0, \text{and } y_v = z_v \forall v \in V \backslash \{u, w\}. \tag{4.13}$$

Then

$$\sum_{v\in V} \alpha_{jv} y_v = \sum_{v\in V} \alpha_{jv} z_v = b_j. \tag{4.14}$$

Suppose not, then WLOG let $\sum_{v\in V} \alpha_{jv} z_v < b_j$. Then,

$$\sum_{v\in V} z_v \leq \sum_{v\in V} (\sum_{i\in I} \alpha_{iv} \lambda_i) z_v = \sum_{i\in I} (\sum_{v\in V} \alpha_{iv} z_v) \lambda_i \leq \sum_{i\in I} \lambda_i b_i = \theta_\Pi(G),$$

contradicting (4.12). Now, equations (4.12) and (4.13) imply that $\alpha_{ju} = \alpha_{jw}$, and this holds for any $(u, w) \in E^*$. Now, since G^* is connected, we have $\mu = \alpha_{jv} = \alpha_{jk} >$

$0, \forall v \in V$, for some positive scalar μ. Hence, equations (4.12) and (4.14) imply that

$$b_j = \sum_{v \in V} \alpha_{jv} z_v = \mu \sum_{v \in V} z_v = \mu \theta_\Pi(G),$$

implying that the inequality $\sum_{v \in V} x_v \leq \theta_\Pi(G)$ induces a facet of $P_\Pi(G)$. ∎

An s-stable cluster is a hereditary property on graphs such that $e \in E' \backslash E$ is an essential edge if $\omega_s(G + e) = \omega_s(G) + 1$. Then, we have the following corollary.

Corollary 4.3.5 *Let $G = (V, E)$ be a graph and $E^* \subseteq E' \backslash E$ be the set of essential edges of G for an s-stable cluster. If $G^* = (V, E^*)$ is connected then the inequality $\sum_{v \in V} x_v \leq \omega_s(G)$ induces a facet of $P_s(G)$.*

4.3.2 Paths, Holes, & Wheels

In general, classes of facets induced by paths, holes and wheels have been developed for some clique relaxation models that include co-s-plex [41].

Let P^n denote the path on n vertices. If $n \leq 2s$, then P^n is an s-stable cluster as the size of any maximal independent set in P^n is s. For $n > 2s$, we have the following result.

Lemma 4.3.6 *$\omega_s(P^n) = 2s$, $\forall n > 2s, \forall s \geq 1$.*

Proof Given a path P^n, label the vertices with $\{1, .., n\}$ such that, vertex i is adjacent to vertices $i-1, i+1, \forall i \in \{2, .., n-1\}$. Then, any consecutive set of $2s$ vertices is an s-stable cluster. Hence, $\alpha_s(P^n) \geq 2s$. For the other inequality, note that the set of odd vertices and even vertices respectively form maximal independent sets in P^n. As the maximal independent set inequalities induce facets of $P_s(G)$, we have, $\omega_s(G) \leq \sum_{v \in P^n} x_v \leq 2s$.

Let H^n denote a hole (induced chordless cycle) on n vertices. If $n \leq 2s+1$, then H^n is an s-stable cluster. If $n \geq 2s+2$, then H^n is not an s-stable cluster as there exists an independent set of size at least $s+1$.

Lemma 4.3.7 $\omega_s(H^n) = 2s$, $\forall n \geq 2s+2, s \geq 1$.

Proof Label the vertices in H^n with $\{1, 2, .., n\}$ such that $N(1) = \{2, n\}, N(n) = \{1, n-1\}, N(i) = \{i-1, i+1\}, \forall i \in \{2, .., n-1\}$. Then, any set of $2s$ consecutive vertices is an s-stable cluster and hence, $\alpha_s(H^n) \geq 2s$. Consider the cases:

1. *n is even*: The set of odd vertices and even vertices form maximal independent sets in H^n. Hence, using the argument in Lemma 4.3.6 the result follows.

2. *n is odd*: Consider the following maximal independent sets of size $\frac{n-1}{2}$: $I_i = \{i, (i+2) \mod n, (i+4) \mod n, ..., (n+(i-3)) \mod n\}, \forall i \in \{1, .., n\}$. The number of such sets is n, and each vertex occurs exactly in $\frac{n-1}{2}$ inequalities. Since they are maximal independent sets, we have $\sum_{v \in I_i} x_v \leq s, \forall i \in \{1, 2, .., n\}$. Summing the n inequalities,

$$\sum_{v \in H^n} x_v \leq 2s\frac{n}{n-1} \implies \sum_{v \in H^n} x_v \leq \lfloor 2s + \frac{2s}{n-1} \rfloor.$$

   Since $n \geq 2s+2$, we have, $\omega_s(H^n) \leq \sum_{v \in H^n} x_v \leq 2s$, and the result follows.

Theorem 4.3.8 *Let $G = (V, E)$ be an odd hole on n vertices with $n > 2s+1$. Then the inequality $\sum_{i \in V} x_i \leq 2s$ induces a facet of $P_s(G)$.*

Proof We will prove this result using Corollary 4.3.5. In an odd hole, any consecutive set of $2s-2$ vertices $S \subset V$ form an $(s-1)$-stable cluster. Then, for any $i \in \{1, 2, \ldots, n\}$, there exists an edge $(i, (i \mod n)+1) \in E$ such that $(i, v), ((i \mod n)+1, v) \notin E$ for any $v \in S$, and the set $S' = S \cup \{i, (i \mod n)+1\}$ is an

s-stable cluster of size $2s$. If we add an edge $(i, (i \mod n) + 2)$ to G, then the set $\{i, (i \mod n)+1, (i \mod n)+2\}$ forms a triangle, and the set $S' \cup \{(i \mod n)+2\}$ forms an s-stable cluster of size $s+1$. Hence the edge $(i, (i \mod n) + 2)$ is an essential edge of G. This is true for any pair of vertices $i, (i \mod n) + 2 \in V$, for $i \in \{1, 2, \ldots, n\}$, and if E' is the edge set such that $G = (V, E')$ is a complete graph, then the set of edges $E^* = \{(i, (i+2) \mod n) \in E' : \forall i = 1, 2, \ldots, n\}$ is an essential edge set of G for an s-stable cluster. Then $G^* = (V, E^*)$ is a Hamilton cycle, and by Corollary 4.3.5, the inequality $\sum_{v \in H^n} x_v \leq 2s$ induces a facet of $P_s(H^n)$. ∎

A wheel W^{n+1} on $n+1$ vertices is a graph containing a hole H^n and an extra vertex u, such that u is adjacent to each $v \in H^n$. It is easy to see that $\omega_s(W^{n+1}) = 2s+1$. Then, $\sum_{v \in W^{n+1}} x_v \leq 2s+1$ is valid for $P_s(W^{n+1})$.

Corollary 4.3.9 *For n odd, and $n > 2s+1$, the inequality $\sum_{v \in H^n} x_v \leq 2s$ is a facet of $P_S(W^{n+1})$.*

Proof We prove the result by lifting the odd hole inequality given in Theorem 4.3.8. By Theorem 4.3.8, $\sum_{v \in H^n} x_v \leq 2s$ induces a facet of $P_s(H^n)$. Let u be the additional vertex in W^{n+1} such that u is adjacaent to all $v \in H^n$, and let $\alpha_u = 2s - \max\{\sum_{v \in H^n} x_v : x \in P_s(W^{n+1}), x_u = 1\}$. Then, $\alpha_u x_u + \sum_{v \in H^n} x_v \leq 2s$ induces a facet of $P_s(W^{n+1})$, and we know that $\alpha_u = 0$. Hence, $\sum_{v \in H^n} x_v \leq 2s$ induces a facet of $P_s(W^{n+1})$. ∎

Now, we provide a class of graphs, for which there exists a complete description for the s-stable cluster polytope. Let $B'x \leq s$ be the restricted set of inequalities given by (4.10), that only includes the maximal independent sets of size at least $s+1$.

Theorem 4.3.10 *Let $G = (V, E)$ be a union of paths $P_1, P_2, \ldots P_r$, $r \geq 1$ such that $|V| \geq s+1$. Then $P_s(G) = \{x \in R^n_+ : B'x \leq s, x \leq 1\}, s \geq 2$.*

Proof Suppose $P_s(G) \neq \{x \in R^n_+ : B'x \leq s, x \leq 1\}$. Then, there exists a facet $F_\alpha = \{x \in P_s(G) : \alpha x = \beta\}$, where the inequality $\alpha x \leq \beta$ is neither induced by any maximal independent set nor is a positive scalar multiple of the inequality $x \leq 1$. First, note that G can be written as a disjoint union of at most two maximal independent sets, as G is a union of connected components $P_1, P_2, ..P_r$, $r \geq 1$ where each C_i denotes a path. Then, $V = H_1 \cup H_2$, where H_1, H_2 are disjoint maximal independent sets of G. To see this, consider a component P_p, such that $|P_p|$ is odd. Then, we can split this into two independent sets P_{p_1} and P_{p_2} of cardinality $|\frac{P_p+1}{2}|$ and $|\frac{P_p-1}{2}|$, respectively. Similarly, if $|P_p|$ is even, we get two independent sets P_{p_1} and P_{p_2}, each of size $|\frac{P_p}{2}|$. Then, $H_1 = \cup_{p \in \{1,..,r\}} P_{p_1}$, and $H_2 = \cup_{p \in \{1,..,r\}} P_{p_2}$ are two disjoint maximal independent sets of G, such that H_1 is the largest independent set in G.

Given that $F_\alpha = \{x \in P_s(G) : \alpha x = \beta\}$ is a facet of $P_s(G)$, we have $\beta = \max\{\alpha x : x \in P_s(G)\}$. If $|H_1| < s+1$, then $\omega_s(G) = |V|$, and $F_\alpha \subseteq \{x \in P_s(G) : x_i = 1\}$. Then, by Theorem 4.3.2, F_α is not a facet. If $|H_1| \geq s+1$, we claim that $F_\alpha \subseteq \{x \in P_s(G) : \sum_{i \in H_1} x_i = s\}$. Suppose not. Then, for each $x \in F_\alpha$, there is a maximal independent set I that is satisfied at equality, as otherwise $\beta \neq \max\{\alpha x : x \in P_s(G)\}$. Then, for each $i \in I$ such that $x_i = 1$, either i is an isolated vertex in G, in which case $i \in H_1$, or, if i is not an isolated vertex and $i \notin H_1$, then there exists $j \in V$ such that $j \in N(i)$ and $x_j = 1$, in which case $j \in H_1$. Note that the maximum number of such vertices possible is s, and hence $\sum_{i \in H_1} x_i = s$. This implies that $\forall x \in F_\alpha$, $\sum_{i \in H_1} x_i \leq s$ is always satisfied at equality, and $F_\alpha \subseteq \{x \in P_s(G) : \sum_{i \in H_1} x_i = s\}$. Hence, for any $\alpha > 0, s \geq 2$, $\alpha x \leq \beta$ does not induce a facet of $P_s(G)$. ∎

Corollary 4.3.11 $P_s(P^n) = \{x \in R^n_+ : B'x \leq s, x \leq 1\}, s \geq 2.$

Corollary 4.3.12 $P_s(H^n) = \{x \in R^n_+ | B'x \leq s, \sum_{i \in V(H^n)} x_i \leq 2s, x \leq 1\}$, *for n* *odd, $n \geq 2s+2$.*

Proof Suppose not. Then, there exists a facet $F = \{x \in P_s(H^n) | \alpha x = \beta\}$, where the inequality $\alpha x \leq \beta$ is not induced by the vertex set $V(H^n)$, any maximal independent set, or is a positive scalar multiple of the inequality $x \leq 1$. Define $V_\alpha = \{v \in V : \alpha_v > 0\}$ and consider the graph $G[V_\alpha]$. $G[V_\alpha]$ is either a connected path or a set of disconnected paths, and F induces a facet of $P_s(G[V_\alpha])$, which is contradiction to Theorem 4.3.10.

Corollary 4.3.13 $P_s(H^n) = \{x \in R^n_+ | B'x \leq s, x \leq 1\}$*, for n even, $n \geq 2s+2$.*

4.4 Exact Algorithms

In this section we describe two exact algorithms for the maximum s-stable cluster problem, and discuss their performance on various graphs, when $s = 2$. In particular, we implement a branch and cut framework, and adapt a combinatorial branch and bound algorithm for hereditary structures for the maximum s-stable cluster problem.

4.4.1 Branch & Cut Algorithm

Here, we discuss the implementation of the branch and cut algorithm for the maximum s-stable cluster problem, when $s = 2$. The implementation is done using ILOG CPLEX 11.0 framework with the CPLEX callback functions. This framework is very advantageous as it has effective default settings for branching process, node selection and allows the user to customize the branch and cut algorithm in CPLEX that includes the separation and addition of user-defined cuts.

In our implementation, we only test the performance of the maximal independent set inequalities for the maximum s-stable cluster problem. Preliminary computational experiments shows that, the upper bound obtained by solving the LP relaxation given by (4.3)-(4.5) is very weak. Hence, we strengthen this relaxation by adding $O(n^2)$ maximal independent set inequalities (4.10), generated using Heuristic 3, and use it in the branch and cut algorithm for finding a strong upper bound in the root node. Ideally, it is enough to solve the LP relaxation given by maximal independent set inequalities that subsumes the inequalities given by (5.2). However, generating all these inequalities is tedious, and hence this approach is not used. Given an LP solution, we apply the greedy heuristic used by Balasundaram et al [5] for the k-plex problem, to identify the violated maximal independent set inequalities.

Separation of MIS: Given an LP solution x for the maximum s-stable cluster problem, we remove all the vertices in V for which $x_v = 0$ to obtain V'. For generating a maximal independent set I, we initialize I with every vertex from V'. We find a vertex with minimum degree from $G[V' \backslash N[I]]$ to update I, and repeat this step until $V \backslash N[I]$ is empty.

In addition, we turned off all the CPLEX generated cuts, as we found that adding these cuts increased the computation time, but we used the default settings in CPLEX for the branching process. We set a time limit of 3 hours using the CPLEX parameter *TiLim*. The algorithm terminates if the problem is infeasible or it reports the best integer feasible solution and the upper bound on the optimal solution, if optimality is not reached within the time limit.

4.4.2 Algorithms for Hereditary Structures

Given a graph $G = (V, E)$ and a hereditary property Π, Tukhanov et el. [60] gave an exact algorithmic framework for detecting optimal structure satisfying Π. This is an exact combinatorial branch and bound algorithm that chooses a candidate set C from V, and finds the optimal set of vertices S satisfying property Π from C. This procedure is iterated by adding a vertex from $V \backslash C$ to the candidate set C, and updating the optimal solution, until $V \backslash C = \emptyset$. In each iteration, given

Heuristic 1 Greedy Heuristic to Generate $O(n^2)$ MIS.

```
for i ∈ V do
    for j ∈ V\N[i] do
        I = {i, j}
        for v = j + 1, ..., |V| + j − 1 do
            u = v  mod |V|
            if u ∉ N[I] then
                I = I ∪ {u}
            end if
        end for
    end for
end for
```

a candidate set C and the current feasible solution S, an updated candidate set $C' \subseteq C \backslash S$ is obtained, which satisfies the property that $S \cup \{v\}$ is a feasible solution, for every $v \in C'$. This updated candidate set C' is obtained using a simple verification procedure based on property Π that, given a candidate set C and a feasible solution $S \subseteq C$, checks if $S \cup \{v\}$ is a feasible solution, for every $v \in C \backslash S$. The complete details of this algorithm can found in [60].

We know that the s-stable cluster is a hereditary property on induced subgraphs, and hence we can adapt their algorithm by giving a suitable verification procedure. Given a candidate set C and an s-stable cluster $S \subseteq C$, it should be noted that the most straight forward way to verify if $S' = S \cup \{v\}$ is also an s-stable cluster, for any $v \in C \backslash S$, is to ensure that there exists no stable set of cardinality $s + 1$ in S'. This involves checking every $s + 1$ vertices in set S, and hence is a very tedious process for large values of s. However, for small values of s, this procedure works well and is given by Algorithm 2. Given a feasible solution S and a vertex $v \in C \backslash S$, the algorithm checks if $S \cup \{v\}$ is an s-stable cluster.

Algorithm 2 Simple s-stable set verification procedure.

function IS2STABLESET(S,s,v)
  $NN_i \longleftarrow$ non-neighbors of vertex i in S
  for $u \in S$ **do**
    if $(u, v) \notin E$ **then**
      $NN_v \longleftarrow NN_v \cup \{u\}$
    end if
  end for
  if $\alpha(G[NN_v]) = s$ **then**
    return $false$
  end if
  return $true$
end function

4.5 Computational Experiments

In this section, we discuss the computational results of the exact algorithms implemented for s-stable clusters, when $s = 2$. All numerical experiments were run on Dell Computer with Intel® Xeon® E5620 2.40 GHz processor and 12GB of RAM, and CPLEX solver was used for the branch & cut implementation. The test bed for the experiments consists of instances from the second and the tenth DIMACS implementation challenges [16, 17], and some random instances [19]. The random instances are based on $G(n, p)$ model [19], where n represents the number of vertices and p denotes the probability that two vertices are adjacent. We considered p values of $0.75, 0.8, 0.85, 0.9, 0.95$ and $0.25, 0.35, 0.45, 0.55, 0.65$ to test the performance of the algorithms, respectively on dense graphs and graphs of sparse and moderate density. We also restricted the value of n to 200 as the branch and cut algorithm ran out of memory for sparse graphs with more than 200 vertices, and as mentioned earlier, we imposed a time limit of 3 hours on both algorithms. In all the tables given, the branch and cut algorithm is denoted by "BC", and the combinatorial branch and bound algorithm is denoted by "A2".

Tables 4.1 & 4.2 show the running times in seconds for the algorithms, respectively for random graphs with moderate and high density. The 2-stable cluster numbers for graphs with moderate and high density obtained respectively by the combinatorial branch and bound, and the branch and cut algorithms are given in Tables 4.3 & 4.4. The computational results for the DIMACS and coloring instances are given in Table 4.5, which contains the instance information, including graph order, size, density, the 2-stable clusters numbers found by the algorithms, and their run times in seconds. In the tables, a run time of ">10800" implies that the algorithm could not find an optimal solution within the time limit, in which case, the size of the 2-stable cluster found by the combinatorial branch and bound algorithm is denoted by $\geq$, and for the branch and cut algorithm, the best integer solution and the bound on the optimal solution found are reported.

From Tables 4.1 & 4.2, it is clear that both the algorithms perform well for moderate and very dense graphs of size < 100. However, it can be noted that, for graphs of cardinality ≥ 100, the branch and cut algorithm, in general, performs well for dense graphs, which is evident from the running times that gradually increases as the density of the graph decreases. This may be due to the fact that, in dense graphs the number of maximum independent set inequalities violated is less, and adding the $O(n^2)$ maximal independent set inequalities at the root node produces a strong relaxation. Similarly, it can be observed that the combinatorial branch and bound algorithm performs well for graphs of sparse and moderate density, which was also evident from the computational results presented in [60]. The same phenomenon is reiterated in the results presented for the DIMACS instances in Table 4.5. The observation, that the branch and cut algorithm runs out of memory for large and sparse graphs may be due to the separation heuristic employed for generating the

Table 4.1: Comparison of run times (in sec) on random graphs ($n > 50, p \leq .65$) for $s = 2$.

n	$p = .25$		$p = .35$		$p = .45$		$p = .55$		$p = .65$	
	A2	BC	A2	BC	A2	BC	A2	BC	A2	BC
50	0.02	4.06	0.02	9.32	0.04	4.33	0.07	6.04	0.3	0.88
60	0.07	23.8	0.11	22.24	0.22	33.51	0.99	20.22	3.44	7.35
70	0.07	96.13	0.27	155.74	1.22	105.6	3.09	91.8	11.32	24.97
80	0.13	305.75	0.57	407.4	2.06	503.97	33.95	454	67.26	59.3
90	0.22	1237.88	1.08	728.11	6.12	1859.44	87.33	1211.92	959.88	1574.02
100	0.27	2347.27	1.31	4114.34	25.31	7932.4	252.01	4139.52	6751.2	4072.31
110	0.85	4815.74	4.5	>10800	32.09	>10800	823.32	>10800	>10800	>10800
120	0.59	7479.3	9.89	>10800	224.09	>10800	6756.67	>10800	>10800	>10800
130	0.82	>10800	23.97	>10800	256.1	>10800	4790.84	>10800	>10800	>10800
140	3.96	>10800	43.08	>10800	840.55	>10800	>10800	>10800	>10800	>10800
150	3.98	>10800	121.04	>10800	2111.3	>10800	>10800	>10800	>10800	>10800
160	8.73	>10800	103.11	>10800	7148.36	>10800	>10800	>10800	>10800	>10800
170	16.26	>10800	332.86	>10800	6784.55	>10800	>10800	>10800	>10800	>10800
180	39.51	>10800	254.62	>10800	>10800	>10800	>10800	>10800	>10800	>10800
190	23.93	>10800	578.27	>10800	>10800	>10800	>10800	>10800	>10800	>10800
200	59.18	>10800	1109.59	>10800	>10800	>10800	>10800	>10800	>10800	>10800

Table 4.2: Comparison of run times (in sec) on random graphs ($n > 50, p \geq .75$) for $s = 2$.

n	$p = .75$		$p = .80$		$p = .85$		$p = .90$		$p = .95$	
	A2	BC	A2	BC	A2	BC	A2	BC	A2	BC
50	1.41	0.25	1.05	0.16	0.32	0.05	0.35	0.02	0.02	0.02
60	9.11	0.46	7.77	0.29	1.38	0.04	0.16	0.02	0.02	0.04
70	156.57	6.09	112.26	1.00	189.94	0.16	0.32	0.02	0.04	0.02
80	1150.87	85.29	631.09	4.84	1216.89	2.27	45.42	0.07	0.04	0.05
90	7981.70	130.84	>10800	38.97	9121.37	0.82	1425.53	0.05	0.15	0.00
100	>10800	311.65	>10800	249.90	>10800	11.53	>10800	0.18	40.82	0.08
110	>10800	>10800	>10800	4223.68	>10800	42.61	>10800	0.44	41.72	0.04
120	>10800	>10800	>10800	3388.53	>10800	104.01	>10800	0.29	>10800	0.04
130	>10800	>10800	>10800	>10800	>10800	1582.26	>10800	0.58	>10800	0.10
140	>10800	>10800	>10800	>10800	>10800	9204.79	>10800	6.49	>10800	0.08
150	>10800	>10800	>10800	>10800	>10800	>10800	>10800	23.02	>10800	0.04
160	>10800	>10800	>10800	>10800	>10800	>10800	>10800	31.41	>10800	0.02
170	>10800	>10800	>10800	>10800	>10800	>10800	>10800	1509.59	>10800	0.05
180	>10800	>10800	>10800	>10800	>10800	>10800	>10800	>10800	>10800	0.05
190	>10800	>10800	>10800	>10800	>10800	>10800	>10800	>10800	>10800	0.04
200	>10800	>10800	>10800	>10800	>10800	>10800	>10800	>10800	>10800	0.16

Table 4.3: 2-stable clusters found by A2 on random graphs.

n	$p = .25$	$p = .35$	$p = .45$	$p = .55$	$p = .65$
50	9	11	13	16	21
60	10	12	14	18	23
70	10	13	15	20	25
80	11	13	16	20	28
90	11	14	17	23	26
100	11	14	18	22	29
110	12	14	18	23	$\geq$28
120	13	15	19	23	$\geq$29
130	12	15	19	25	$\geq$29
140	12	15	19	$\geq$24	$\geq$29
150	13	15	19	$\geq$24	$\geq$28
160	12	16	19	$\geq$23	$\geq$29
170	13	16	20	$\geq$25	$\geq$30
180	13	16	$\geq$20	$\geq$23	$\geq$30
190	14	16	$\geq$20	$\geq$23	$\geq$27
200	13	16	$\geq$21	$\geq$23	$\geq$29

violated maximal independent set inequalities, which is quite expensive. Overall, our computational results suggest that the branch and cut algorithm performs well for dense graphs, while the combinatorial branch and bound algorithm works well for sparse and moderately dense graphs.

4.6 Conclusion

In this work, we introduced a new clique relaxation, namely s-stable cluster, for modeling clusters with the independence number bounded by s. We studied some basic properties associated with this model, and established the NP-completeness of the maximum s-stable cluster problem, for any fixed positive integer s, on claw-free graphs. In addition, we studied some optimality conditions established for stable set problem, and analyzed their relevance for s-stable clusters, for $s \geq 2$. Two

Table 4.4: 2-stable clusters found by B&C on random graphs.

n	$p = .75$	$p = .80$	$p = .85$	$p = .90$	$p = .95$
50	26	31	35	40	48
60	31	36	40	50	58
70	33	39	46	57	67
80	36	41	47	60	76
90	38	44	54	67	84
100	41	46	56	70	89
110	[39,41]	47	61	73	99
120	[42,45]	51	64	80	102
130	[41,48]	[51,56]	66	86	113
140	[41,51]	[52,59]	68	92	116
150	[42,56]	[54,64]	[70,74]	91	123
160	[46,60]	[54,67]	[69,78]	96	127
170	[43,63]	[57,72]	[73,84]	98	138
180	[44,65]	[56,74]	[74,86]	[99,102]	142
190	[43,68]	[54,77]	[70,90]	[102,107]	150
200	[43,72]	[55,80]	[73,93]	[110,113]	157

Table 4.5: Compuational results for DIMACS Instances for $s = 2$.

Instance				BC		A2	
	Vertices	Edges	Density %	$\omega_2(G)$	time (sec)	$\omega_2(G)$	time (sec)
adjnoun	112	425	6.84	9	111.40	9	0.03
anna	138	493	5.22	18	10.35	18	0.02
c-fat200-1.clq	200	1534	7.71	24	85.45	24	0.08
c-fat200-2.clq	200	3235	16.26	46	97.80	46	0.06
c-fat200-5.clq	200	8473	42.58	116	6894.89	≥99	>10800
david	87	406	10.86	19	1.16	19	0.02
dolphins	62	159	8.41	10	0.47	10	0.02
football	115	613	9.36	18	5.26	18	0.16
hamming6-2.clq	64	1824	90.48	64	0.02	64	0.02
hamming6-4.clq	64	704	34.93	10	3.55	10	0.01
hamming8-2.clq	256	31616	96.87	256	0.05	256	1.44
hamming8-4.clq	256	20864	63.93	32	76.80	32	1852.57
huck	74	301	11.15	20	0.55	20	0.01
ieeebus	118	179	2.60	7	104.38	7	0.02
jazz	198	2742	14.06	47	1300.38	47	0.07
johnson16-2-4.clq	120	5460	76.48	16	0.15	≥16	>10800
johnson8-2-4.clq	28	210	55.56	8	0.04	8	0.01
johnson8-4-4.clq	70	1855	76.82	28	0.07	28	0.21
karate	34	78	13.91	9	0.08	9	0.03
kreb	62	153	8.10	11	0.52	11	0.01
MANN_a27.clq	378	70551	99.02	261	0.07	261	2.28
MANN_a9.clq	45	918	92.73	33	0.00	33	0.02
san200_0.7_1	200	13930	70.00	[48,53]	>10800	≥28	>10800
san200_0.7_2	200	13930	70.00	[35,36]	>10800	≥16	>10800
san200_0.9_1	200	17910	90.00	125	8.54	≥75	>10800
san200_0.9_2	200	17910	90.00	114	30.74	≥65	>10800
san200_0.9_3	200	17910	90.00	88	0.11	≥46	>10800
sanr200_0.7	200	13868	69.69	[34,63]	>10800	≥33	>10800
sanr200_0.9	200	17863	89.77	[105,111]	>10800	≥65	>10800

binary programming formulations are presented for the maximum s-stable cluster problem, and different classes of facets are introduced for the associated polytope. A branch and cut algorithm is implemented using a family of facets introduced, and a combinatorial branch and bound algorithm for hereditary structures is adapted. Computational experiments are performed for a number of random and standard graph instances using both algorithms, and their results are compared.

This chapter presents more of a basic study of the s-stable cluster model, which gives way to a number of research questions that needs to be answered. In view of the fact that the s-stable cluster has a similar polyhedral structure as a clique, it is imperative to analyze, if all the properties that hold for a clique polytope, still hold true for s-stable clusters, when $s \geq 2$. Another important question is, can a relaxation be obtained for the maximum s-stable cluster problem, that finds as good an upper bound as the Lovász [38] theta number is for the maximum clique problem?

5. STRONG UPPER BOUNDS FOR THE MAXIMUM CLIQUE PROBLEM

In Chapters 2, 3, & 4, we discussed clique relaxation models for modeling cohesive, robust and stable clusters, which were introduced to overcome the restrictive nature of cliques, and are more practical and useful in various applications. This however has not reduced the popularity that the maximum clique problem enjoys, and it is still one of the well-studied problems in theoretical computer science. The fact that the problem is NP-hard [27] and is hard to approximate [32], has made it all the more challenging and interesting to study.

Motivated by these factors, many exact algorithms that use branch and bound technique [46, 49, 51, 58, 66], enumerative algorithms [8, 59], and a number of heuristics [9, 28] that approximate $\omega(G)$, have been developed. To this end, many convex relaxations [50] that determine a good upper bound on $\omega(G)$ have also been proposed. Perhaps the strongest results in this direction are related to the notion of Lovász theta [38], denoted by $\theta(G)$, which satisfies the inequalities $\omega(G) \leq \theta(G) \leq \chi(G)$, known as the sandwich theorem [35]. Here $\chi(G)$ is the chromatic number, which is the minimum number of colors required for a proper coloring of G. Both $\omega(G)$ and $\chi(G)$ are hard to compute, whereas $\theta(G)$ can be computed in polynomial time. The Lovász theta and its stronger variants have been used to obtain tight upper bounds on the clique number in several works [7, 31, 33, 37, 56]. However, computing such bounds involves lifting to spaces of very high dimensions, making them applicable to only small graphs in practice.

In this work, we take an alternative approach based on clique relaxation models obtained by enforcing elementary clique-defining properties [52]. In such models, we require that a subset of vertices satisfies a property that must hold for any clique of a fixed size $k + 1$. Then solving the problem of minimizing the size of such a clique relaxation structure in G will yield a clique of size $k+1$ whenever one exists, and an empty set or a set of size greater than $k + 1$ if there is no clique of size $k + 1$ in the graph. Hence, if a heuristic is available that produces a maximum clique on some graph G, this approach can be used to verify the optimality of the heuristic solution by either detecting infeasibility or establishing a nontrivial (that is, better than $k + 1$) lower bound on the objective of the considered minimization problem for $k = \omega(G)$. Hence, our objective in this work is to develop nontrivial, easily computable lower bounds for the minimization problem in order to get a tight upper bound on $\omega(G)$. More specifically, linear programming (LP) based bounds for two variants of the minimum k-core problem (defined below) are investigated and compared to the standard LP relaxations for the maximum clique problem.

Given $G = (V, E)$ and positive integer k, a k-core S is such that $\delta(G[S]) \geq k$, and the degeneracy of G is given by the largest k for which G has a nonempty k-core. Obviously, any clique of size $k + 1$ is a k-core. Hence, given a heuristic lower bound k for $\omega(G)$, the k-core obtained by recursively removing all vertices of degree less than k from the graph ("peeling") can be used for scale-reduction in very large and sparse graphs for employing exact algorithms to solve the maximum clique problem [1, 18, 65]. However, the peeling procedure yields the maximum-size k-core in the graph that may be much larger than the clique number. In this work, we use *minimum* k-cores to get a tighter upper bound on the clique number of a graph. Consider the following problem:

Definition 19 (Minimum k-core) *Given a graph $G = (V, E)$, and a positive integer k such that G has degeneracy at least k, find a smallest non-empty k-core in G. The size of a minimum k-core is denoted by $m_k(G)$.*

It is easy to see that if there exists a non-empty clique of size $k+1$, then $m_k(G) = k+1$, and for a given k, if there exists no k-core in G or $m_k(G) > k+1$, then $\omega(G) \leq k$. This property makes this problem very suitable for developing a good upper bound for the maximum clique problem. However, using exact algorithms for finding the smallest such k satisfying this property may not be a good approach from a computational standpoint as, unlike the maximum k-core which is polynomially solvable, the minimum k-core problem does not admit a constant factor approximation for $k \geq 3$, unless P = NP [4]. Hence, as a starting point, we aim to show that simple linear programming relaxations of the minimum k-core problem and its stronger versions can be used to find easily computable, tight upper bounds for the maximum clique problem.

The organization of this chapter is as follows. Section 5.1 discusses the minimum k-core problem and its stronger version, which can be exploited to develop continuous relaxations to find a good upper bound for the clique number. The linear programming relaxations and its comparison to the standard relaxations for the clique polytope is discussed in Section 5.2, and Section 5.3 reports the results of computational experiments. Finally, Section 5.4 concludes this chapter.

5.1 Minimum k-Core and k-Core/2-Club problems

A subset S is called a *minimal* k-core if $G[S \backslash S']$ is not a k-core for any proper subset $S' \subset S$. By definition, a minimum k-core is minimal, and may not be unique. Observe that, any minimal k-core is connected.

For a given positive integer s, a set $S \subseteq V$ is an *s-club* if $diam(G[S]) \leq s$. An s-club S is a k-core/s-club if $\delta(G[S]) \geq k$. A minimal k-core/s-club can be defined in a similar manner as a minimal k-core. Then, we consider the minimum k-core/s-club problem that is defined as follows.

Definition 20 (Minimum k-core/s-club) *Given a graph $G = (V, E)$ and a positive integer k, find a smallest non-empty k-core/s-club in G. The size of a minimum k-core/s-club is denoted by $m_{k,s}(G)$.*

Note that any k-core/s-club will always be a k-core, but the converse need not be true, and hence, $m_{k,s}(G) \geq m_k(G)$. Thus, the property of a minimal k-core that makes it suitable for developing a good upper bound on the clique number of a graph also holds true for a k-core/2-club, that is, for a given k, if there exists no k-core/s-club in G or $m_{k,s}(G) > k + 1$, then $\omega(G) \leq k$ and the smallest k satisfying this property is a tight upper bound on the clique number.

A small value of s will inadvertently help in reducing the search space, especially in sparse graphs and help in achieving the goal of a tighter structure that is closer to clique. Thus, we consider the special case of the k-core/s-club model with $s = 2$, namely the k-core/2-club model, which has the following relation with a k-core.

Property 1 *A k-core $G = (V, E)$ of cardinality at most $2k + 1$, is connected with $diam(G) \leq 2$.*

In fact a k-core of size $k + 1$ is a clique, and hence its diameter is one.

As we pointed out above, the inequality $m_{k,s}(G) \geq m_k(G)$ always holds. An interesting question to answer is, when is $m_k(G)$ strictly less than $m_{k,s}(G)$? Next we show that this question is NP-hard to answer. We start with the following lemma.

Lemma 5.1.1 *(a) Given $k \geq 3, j > k + 1$, a graph $G = (V, E)$ can be constructed in polynomial time such that $|V| = j$ and V is a minimal k-core that is also a minimal k-core/2-club.*

(b) Given $k \geq 3, j > 2k + 1$, a graph $G = (V, E)$ can be constructed in polynomial

time such that $|V| = j$ and V is a minimal k-core that is not a minimal k-core/2-club.

Proof (a) Let A and W be vertex sets of size $k-3$ and $j-k+3$, respectively. Let $G = (V, E)$ be a graph with $V = A \cup W$ and E such that $G[A]$ is complete, $G[W]$ is a wheel, and the remaining edges are given by $E' = \{(a, w) : a \in A, w \in W\}$. It is easy to see that V is a minimal k-core and a minimal k-core/2-club.

(b) Consider two vertex sets A_1, A_2 of size $k-2$ and a set $C = \{v_1, \ldots, v_{j-2k+4}\}$ of size $j - 2k + 4$. Let $G = (V, E)$ be a graph with $V = A_1 \cup A_2 \cup C$ and E is such that $G[A_1], G[A_2]$ is complete, $G[C]$ is a chordless cycle such that $N(v_1) = \{v_2, v_{j-2k+4}\}, N(v_{j-2k+4}) = \{v_1, v_{j-2k+3}\}$, and $N(i) = \{v_{i-1}, v_{i+1}\}, \forall i \in \{2, \ldots, j-2k+3\}$. The remaining edges in E are given by $E_1 \cup E_2$, where $E_1 = \{(a_1, v_i) : a_1 \in A_1, i \in \{1, \ldots \lfloor \frac{j-2k+4}{2} \rfloor\}$ and $E_2 = \{(a_2, v_i) : a_2 \in A_2, i \in \{\lfloor \frac{j-2k+4}{2} \rfloor + 1, \ldots, j - 2k + 4\}$. It is easy to see that V is a minimal k-core, and $diam(G) > 2$. ∎

Proposition 5.1.2 *Given a graph $G = (V, E)$, it is NP-hard to check if $m_k(G) = m_{k,2}(G)$ for a fixed $k \geq 3$.*

Proof Suppose there is a polynomial time algorithm $\mathcal{A}$ that, given a graph $G = (V, E)$, correctly answers if $m_k(G) = m_{k,2}(G)$ with "yes" or "no".

1. Suppose the answer is "no", i.e., $m_k(G) < m_{k,2}(G)$. Then $2k + 1 < m_k(G) < m_{k,2}(G)$ Then we prove that $m_k(G)$ can be computed using a polynomial time algorithm $\mathcal{A}_1$, which proceeds as follows. Let $H_{j,k}$ be the graph constructed using Lemma 5.1.1(a). For $j = 2k + 2, 2k + 3, \ldots,$ we incrementally update $G' = G \cup H_{j,k}$ while $m_k(G') = m_{k,2}(G')$. We terminate as soon as $m_k(G') < m_{k,2}(G')$ for some step j', implying that $m_k(G) = j' - 1$.

2. Suppose the answer is "yes", i.e., $m_k(G) = m_{k,2}(G)$. We check if there is a k-core of size $k+1, \ldots, 2k+1$ in G. This can be done in $O(n^{2k+1})$. If there is no k-core of size at most $2k+1$, then we can compute $m_k(G)$ using the following polynomial time algorithm $\mathcal{A}_2$. Let $G_{j,k}$ be the graph constructed using Lemma 5.1.1(b).

   For $j = 2k+2, 2k+3, \ldots$, we consider $G' = G \cup G_{j,k}$ while $m_k(G') < m_{k,2}(G')$. As the value of j increases, at some value j' we will have $m_k(G') = m_{k,2}(G') = j'$, implying that $m_k(G) = m_{k,2}(G) = j'$.

This proves that if such an algorithm $\mathcal{A}$ exists, we can compute $m_k(G)$ in polynomial time. The result follows from the fact that the minimum k-core problem is hard to approximate [4]. ∎

5.2 The Proposed Bounds

In this section, we develop linear programming relaxations for the minimum k-core and k-core/2-club problems, and strengthen by introducing some cutting planes. In addition, we show that the upper bounds found by solving these relaxations are better than those found by solving he standard linear programming relaxations of the clique problem.

5.2.1 Minimum k-Core

Given a graph $G = (V, E)$ and a positive integer k, let x_S denote the characteristic vector of a set $S \subseteq V$. Then, the minimum k-core problem can be formulated as a binary program:

$$m_k(G) = \min \quad \sum_{i \in V} x_i \tag{5.1}$$

$$\text{subject to} \quad \sum_{v \in N(i)} x_v \geq k x_i, \forall i \in V \tag{5.2}$$

$$\sum_{i \in V} x_i \geq k + 1 \tag{5.3}$$

$$x \in \{0, 1\}^{|V|}. \tag{5.4}$$

Constraint (5.2) ensures that for any vertex $i \in V$ such that $x_i = 1$, the degree of i in the subgraph induced by the subset of vertices $\{i : x_i = 1\}$ is at least k, and constraint (5.3) ensures that only a non-empty k-core is included as a feasible solution.

Denote by $P_k(G)$ the k-core polytope of G given by the convex hull of all non-empty k-cores in G, and let $LP_k(G)$ denote the feasible region of the LP relaxation of the above formulation, obtained by relaxing the binary constraint (5.4) to

$$0 \leq x_i \leq 1, \forall i \in V.$$

Next we provide some valid inequalities that are facet inducing for $P_k(G)$, when G is a $(k+1)$-core.

Lemma 5.2.1 *Given a graph $G = (V, E)$, where V is a $(k+1)$-core, we have:*

1. *$dim(P_k(G)) = |V|$.*

2. *$x_v \geq 0$ induces a facet of $P_k(G)$ if $|N(j)| \geq k + 2, \forall j \in V$ such that $v \in N(j)$.*

3. *$x_v \leq 1$ induces a facet of $P_k(G)$ for every $v \in V$.*

Proof Let e be the vector of all ones, and e_i be the unit vector with 1 as the i^{th} component and rest 0.

1. The points $e, e - e_i, \forall i \in V$ are $|V| + 1$ affinely independent points in $P_k(G)$, and hence, $dim(P_k(G)) = |V|$.

2. Let $F_v^0 = \{x \in P_k(G) : x_v = 0\}$. Then $e - e_v, e - e_v - e_w, \forall w \in V \backslash \{v\}$ are $|V|$ affinely independent points in F_v^0. Hence $dim(F_v^0) = |V| - 1$, and F_v^0 is a facet.

3. Let $F_v^1 = \{x \in P_k(G) : x_v = 1\}$. Then $e, e - e_i, \forall i \in V \backslash \{v\}$ are $|V|$ affinely independent points in F_v^1. Hence $dim(F_v^1) = |V| - 1$, and F_v^1 is a facet. ∎

For any $u, v \in V$ such that $(u, v) \notin E$, if u, v are included in the k-core, then the total number of vertices included in the k-core must be at least $k + 2$. Hence, the inequality given by $\sum_{i \in V \backslash \{u,v\}} x_i \geq k$ is valid for $P_k(G)$. This can be generalized for any maximal independent set in the following manner.

Theorem 5.2.2 *(*MIS *Inequality) For a graph $G = (V, E)$, let V be a $(k + 1)$-core. Then for any maximal independent set I of size $k + (k \mod 2)$ or more in G, the inequality*

$$\sum_{v \in V \backslash I} x_v \geq k \tag{5.5}$$

induces a facet of $P_k(G)$ if $(i, j) \in E$ for each $i \in I$, $j \in V \backslash I$.

Proof The validity of the inequality follows from the fact that if a vertex $v \in I$ is in the k-core, then inequality (5.2) implies that $\sum_{v \in V \backslash I} x_v \geq k$, and if I is not included in the k-core, then inequality (5.3) implies that $\sum_{v \in V \backslash I} x_v \geq k + 1$. Let $F_I = \{x \in P_k(G) : \sum_{v \in V \backslash I} x_v = k\}$. To prove that the inequality induces a facet, let there be a valid inequality $\alpha^T x \leq \beta$ such that $F = \{x \in P_k(G) : \alpha^T x = \beta\} \supseteq F_I$.

Let $S = S' \cup I$, where $S' \subseteq V \backslash I$ and $|S'| = k$. Then $x_S \in F_I$. Consider a vertex $v \in S'$ and $w \in V \backslash S$ and let $R = (S \cup \{w\}) \backslash \{v\}$. Then $x_R \in F_I$ and $\alpha^T x_R = \beta$, $\alpha^T x_S = \beta$ implies $\alpha_v = \alpha_w$. Since S, v and w are arbitrary $\alpha_v = \mu, \forall v \in V \backslash I$, for some scalar μ. Consider a subset P' of V such that $|P'| = k$, and if k is even then P' contains $\frac{k}{2}$ pairs of adjacent vertices. For each $i \in I$, let $P_i = P' \cup (I \backslash \{i\})$ and $P = P' \cup I$, then $x_{P_i}, x_P \in F$. Then $\alpha^T x_{P_i} = \beta$, $\alpha^T x_P = \beta$ implies $\alpha_i = 0, \forall i \in I$ and $\beta = k\mu$, proving that F_I is a maximal face and hence a facet. ∎

The result above implies that, the addition of the maximal independent set inequalities may strengthen the considered LP relaxation. Let $\tilde{I}(G)$ denote the family of all maximal independent sets in G. Then, the strengthened relaxation is given by,

$$
\begin{aligned}
Z_{LP}^k = \min \quad & \sum_{i \in V} x_i && (5.6)\\
\text{subject to} \quad & \sum_{v \in N(i)} x_v \geq k x_i, \forall i \in V && (5.7)\\
& \sum_{i \in V} x_i \geq k + 1 && (5.8)\\
& \sum_{i \in V \backslash I} x_i \geq k, \forall I \in \tilde{I}(G) && (5.9)\\
& 0 \leq x_i \leq 1, \forall i \in V. && (5.10)
\end{aligned}
$$

Let K_1^* be such that,

$$K_1^* = \min_{k \in Z_+} k \quad \text{s.t} \quad Z_{LP}^k > k + 1 \quad \text{or} \quad LP_k(G) = \emptyset$$

Then, since $Z_{LP}^k \leq m_k(G)$, we have $\omega(G) \leq K_1^*$. Hence K_1^* is the upper bound on the clique number obtained by solving the linear programming relaxation (5.6)-(5.10) of the minimum k-core problem. Denote this upper bound by UB_1, that is, $UB_1 = K_1^*$.

5.2.2 Minimum k-Core/2-Club

The binary program for the minimum k-core/2-club problem can be formulated by adding the diameter constraints to the formulation of the minimum k-core problem:

$$
\begin{aligned}
m_{k,2}(G) = \min \quad & \sum_{i \in V} x_i && (5.11)\\
\text{subject to} \quad & \sum_{v \in N(i)} x_v \geq k x_i, \forall i \in V && (5.12)\\
& \sum_{v \in N^{\cap}(i,j)} x_v \geq x_i + x_j - 1, \quad \forall (i,j) \notin E, i \neq j && (5.13)\\
& \sum_{i \in V} x_i \geq k+1 && (5.14)\\
& x \in \{0,1\}^{|V|}. && (5.15)
\end{aligned}
$$

where, $N^{\cap}(i,j)$ denotes the common neighborhood of vertices i, j in G, that is, $N^{\cap}(i,j) = N(i) \cap N(j)$. Constraint (5.13) ensures that for any pair of vertices in the subgraph induced by the subset of vertices $\{i : x_i = 1\}$, there exists a path of length at most 2. Denote by $P_{k,2}(G)$ the k-core/2-club polytope that does not include the zero vector. Since any k-core/2-club is also a k-core, the results of the k-core polytope can be extended easily to the k-core/2-club polytope. Hence, inequality (5.5) is valid for $P_{k,2}(G)$.

Let the feasible region of the LP relaxation of the above formulation for the minimum k-core/2-club problem be denoted by $LP_{k,2}$. The corresponding tightened linear programming relaxation obtained after adding the maximal independent inequalities (5.9), that are valid for $P_{k,2}(G)$, is given by,

$$Z_{LP}^{k,2} = \min\{\sum_{i \in V} x_i : (5.9), (5.12), (5.13), (5.14), 0 \leq x_i \leq 1, \forall i \in V\} \qquad (5.16)$$

Let K_2^* be such that,

$$K_2^* = \min_{k \in Z_+} k \quad \text{s.t} \quad Z_{LP}^{k,2} > k+1 \quad \text{or} \quad LP_{k,2} = \emptyset$$

Then, since $m_{k,2}(G) \geq LP_{k,2}(G)$, we have $\omega(G) \leq K_2^*$. Hence K_2^* is the upper bound to the maximum clique problem found by solving the LP relaxation (5.16) of the minimum k-core/2-club problem. Denote this upper bound by UB_2, that is, $UB_2 = K_2^*$.

5.2.3 Comparison with the Fractional Clique Polytope

In this subsection, we review the standard LP relaxations for the maximum clique problem, and compare the corresponding upper bounds with the bounds from the proposed relaxations.

The clique polytope of a graph $G = (V, E)$ denoted by $P_{clique}(G)$, is the convex hull of the incidence vectors of cliques and is given by,

$$P_{clique}(G) = conv\{x \in \{0,1\}^{|V|} : x_i + x_j \leq 1, \forall (i,j) \notin E\},$$

and the edge formulation for the maximum clique problem is given by:

$$\omega(G) = \max\{\sum_{i \in V} x_i : x \in P_{clique}(G)\}.$$

Let $\tilde{I}(G)$ denote the family of all the maximal independent sets in G. Then for any $I \in \tilde{I}(G)$, the inequality given by

$$\sum_{i \in I} x_i \leq 1 \tag{5.17}$$

is a facet of $P_{clique}(G)$. Hence, an alternative formulation based on the maximal independent set inequalities is given by,

$$\omega(G) = \max\{\sum_{i\in V} x_i : \sum_{i\in I} x_i \leq 1, \forall I \in \tilde{I}(G), x \in \{0,1\}^{|V|}\}.$$

Let the fractional clique polytopes based on the edge inequalities and the maximal independent set inequalities be given by,

$$F_e(G) = \{x \in R^n : x_i + x_j \leq 1, \forall (i,j) \notin E, 0 \leq x_i \leq 1, \forall i \in V\}$$
$$F_I(G) = \{x \in R^n : \sum_{i\in I} x_i \leq 1, \forall I \in \tilde{I}(G), 0 \leq x_i \leq 1, \forall i \in V\},$$

and their corresponding linear programming relaxations be given by,

$$Z^e_{LP} = \max_{x\in F_e(G)} \sum_{i\in V} x_i \tag{5.18}$$
$$Z^I_{LP} = \max_{x\in F_I(G)} \sum_{i\in V} x_i \tag{5.19}$$

Then, since $F_I(G) \subseteq F_e(G)$, we have $\omega(G) \leq \lfloor Z^I_{LP} \rfloor \leq \lfloor Z^e_{LP} \rfloor$. Let UB_e, UB_I denote the upper bounds on the clique number of a given graph found by solving the linear programming relaxations (5.18) & (5.19) respectively, where, $UB_I = \lfloor Z^I_{LP} \rfloor$ and $UB_e = \lfloor Z^e_{LP} \rfloor$.

We now show that the upper bounds UB_1, UB_2, found respectively by solving the LP relaxations of the minimum k-core and k-core/2-club problems, are better than the bounds UB_I ans UB_e .

Lemma 5.2.3 $\omega(G) \leq UB_2 \leq UB_1 \leq UB_I \leq UB_e$

Proof We need to show that $UB_1 \leq UB_I$. Assume the contrary, that is, $UB_1 > UB_I$, and let $UB_1 = K^*_1 = s$. Then, $Z^s_{LP} > s+1$ or $LP_s(G) = \emptyset$, and $Z^I_{LP} < s$.

Let $k = s-1$, then by the definition of K_1^*, $Z_{LP}^k = s$. Let $x^{s-1} \in R^n$ be the optimal solution to the linear programming relaxation (5.6) - (5.10) with $k = s-1$, and $\sum_{i\in V} x_i^{s-1} = s$. Then using the MIS inequality $\sum_{i\in V\setminus I} x_i^{s-1} \geq s-1$ we have,

$$\begin{aligned} s = \quad & \sum_{i\in V} x_i^{s-1} \geq s - 1 + \sum_{i\in I} x_i^{s-1}, \forall I \in \tilde{I}(G) \\ \Longrightarrow \quad & \sum_{i\in I} x_i^{s-1} \leq 1, \forall I \in \tilde{I}(G) \end{aligned}$$

This implies that $x^{s-1} \in F_I$, and $Z_{LP}^I \geq \sum_{i\in V} x_i^{s-1} = s$. Then, $UB_I = \lfloor Z_{LP}^I \rfloor \geq s$ which is a contradiction. Hence, $UB_1 \leq UB_I$.

Since $LP_{k,2} \subseteq LP_k$, we have $Z_{LP}^k \leq Z_{LP}^{k,2}$, and hence, $K_2^* \leq K_1^*$. ∎

5.3 Computational Experiments

In this section, we present computational results to evaluate the quality of the upper bounds for the maximum clique problem found by the proposed relaxations, as well as a comparison with some existing bounds found in the literature. Note that the LP relaxations $LP_k, LP_{k,2}$ can be made tighter by adding the maximal independent set inequalities (5.9). However, the number of maximal independent sets in a given instance may be exponential. Hence, we use Heuristic 3 to generate $O(n^2)$ maximal independent sets, and computationally show that adding the inequalities given by (5.9) just for the generated sets make the relaxations very strong. It should be noted that the heuristic duplicates some of the sets, and hence the total number of inequalities generated will be less than n^2. However, computationally it is seen that the cardinality of the maximal independent sets chosen using this method, in general, is quite large, and this makes it very effective in cutting out many fractional solutions.

From Lemma 5.2.3, it is clear that the LP relaxation $LP_{k,2}$ in theory, may give a

Heuristic 3 Greedy Heuristic to Generate $O(n^2)$ MIS.

```
for i ∈ V do
    for j ∈ V\N[i] do
        I = ∅ ∪ {i, j}
        for v = j + 1, ..., |V| + j − 1 do
            u = v mod |V|
            if u ∉ N[I] then
                I = I ∪ {u}
            end if
        end for
    end for
end for
```

better upper bound than LP_k due to the addition of diameter constraints. However, in practice, addition of these constraints does not have much effect, especially in dense graphs due to Property 1. Hence, we exploit the diameter constraints to make the model more tight, by solving the LP relaxation (5.16) in the 2-neighborhood $N[N[v]]$ of a given vertex, where $N[v]$ is the closed neighborhood of v given by $N(v) \cup \{v\}$. This procedure was originally introduced in [6] to solve the maximum 2-club problem and is adapted to serve our purpose as follows. Given a value of k, solve for $Z_{LP}^{k,2}$ using (5.16) in G, however if the value of $Z_{LP}^{k,2}$ obtained is equal to $k+1$, then pick a vertex $v \in V$, and fix $x_v = 1$. Then, solve (5.16) in the 2-neighborhood $N[N[v]]$ of v, and update V by removing v. This process is iterated for each vertex $v \in V$, until $V = \emptyset$, and the minimum solution obtained is reported.

We solve the LP relaxations (5.6)-(5.10) of the minimum k-core problem given in Section 5.2.1 and (5.16) of the minimum k-core/2-club problem given in Section 5.2.2 to get the upper bounds UB_1 and UB_2, respectively. The maximal independent sets used in inequality (5.9) are generated by Heuristic 3, and for a given k, $Z_{LP}^{k,2}$ is obtained using the 2-neighborhood procedure explained above. All numerical experiments were run on Dell Computer with Intel® Xeon® E5620 2.40 GHz processor and 12GB of RAM, and CPLEX 11.0 solver was used. We consider instances from the second and the tenth DIMACS implementation challenges [16, 17] and Trick's coloring instances [13] to perform the experiments. We solve the relaxations on the maximum k-core found in these instances, as it is easy to show that a minimum k-core must be a part of the maximum k-core.

Table 5.1 shows the parameters of the graph instances: graph order, size, density, clique number, the upper bounds UB_1, UB_2 found by the relaxations $LP_k, LP_{k,2}$ respectively, the objective function values $Z^k_{LP}, Z^{k,2}_{LP}$, and the running times in CPU seconds for the proposed relaxations. In the computational results presented in the table, '$*$' denotes that the upper bound found was not the optimal bound, '$-$' denotes that the clique number of the instance is not known, 'inf' denotes that the relaxation was infeasible, and '†' denotes that UB^2 was found after solving the relaxation in the 2-neighborhood of each vertex as described earlier.

From the table, it is clear that the proposed relaxations give very tight upper bounds, mostly optimal bounds, for the considered instances. The relaxations especially give optimal bounds for all but two of the DIMACS instances, and it is also clear that the computational time is in general very low for both the relaxations. However, the running time tends to increase when UB_2 is found by solving $LP_{k,2}$ in the 2-neighborhood of each vertex, but this procedure is only used for eleven of the instances considered. It should also be noted that, for all the instances for which the relaxations yielded an infeasible solution, the largest k-core for $k = \omega(G)$ was empty. This shows that the pre-processing technique works very well given a heuristic lower bound, and hence, the method proposed gives a strong upper bound for $\omega(G)$.

Table 5.1: Upper bounds obtained for DIMACS and Trick's coloring instances.

Instance	Vertices	Edges	Density %	$\omega(G)$	UpperBound		LPR		time(sec)	
					UB_1	UB_2	Z^k_{LP}	$Z^{k,2}_{LP}$	LP_k	$LP_{k,2}$
brock200_1.clq†	200	14834	74.55	21	38*	33*	39.03	34.02	68.38	333.97
brock200_2.clq†	200	9876	49.63	12	22*	16*	23.05	17.04	88.02	493.58
brock200_3.clq†	200	12048	60.55	15	27*	21*	28.02	22.01	80.64	417.91
brock200_4.clq†	200	13089	65.78	17	30*	24*	31.01	25.01	79.82	394.33
c125.9.clq†	125	6963	89.85	34	43*	40*	44.03	41.01	3.22	15.40
c250.9.clq†	250	27984	89.91	44	71*	68*	72.01	69.02	102.17	572.26
c-fat200-1.clq	200	1534	7.71	12	12	12	13.1	13.1	10	4.33
c-fat200-2.clq	200	3235	16.26	24	24	24	25.06	25.06	6.82	4.19
c-fat200-5.clq†	200	8473	42.58	58	66*	58	67.01	59.04	5.56	236.67
DSJC125.1.clq†	125	736	9.50	4	5*	4.00	6.06	5.33	2.83	61.49
DSJC125.5.clq†	125	6961	89.82	10	16*	13*	17.06	14.06	10.43	56.10
DSJC125.9.clq†	125	3891	50.21	34	42*	41*	43.01	42.03	3.14	16.03
hamming6-2.clq	64	1824	90.48	32	32	32	33.04	33.04	0.08	0.08
hamming6-4.clq†	64	704	34.92	4	5*	4	6.16	5.34	0.18	3.2
hamming8-2.clq	256	31616	96.86	128	128	128	129.01	129.01	1.42	1.89
hamming8-4.clq	256	20864	63.92	16	16	16	17.07	17.07	6.83	8.55
johnson16-2-4.clq	120	5460	76.47	8	8	8	9.15	9.15	0.32	0.32
johnson32-2-4.clq	496	107880	87.88	16	16	16	17.06	17.06	14.67	14.49
johnson8-2-4.clq	28	210	55.56	4	4	4	5.33	5.33	0.05	0.05
johnson8-4-4.clq	70	1855	76.81	14	14	14	15.08	15.08	0.07	0.07
keller4.clq†	171	9435	64.92	11	15*	14*	16.06	15.04	22.94	154.26
mann_a27.clq†	378	70551	99.01	126	135*	131*	136.01	132.01	1.44	144.65
mann_a45.clq†	1035	533115	99.63	343	360*	354*	361.003	355.006	17.58	3600.51
mann_a9.clq†	45	918	92.73	16	18*	17*	19.06	18.07	0.02	0.22
san200_0.7_2.clq†	200	13930	70.00	18	20*	18.00	21.09	19.03	19.72	247.97
san200_0.9_1.clq	200	17910	90.00	70	70	70	71.02	71.02	2	2.95
san200_0.9_2.clq	200	17910	90.00	60	60	60	61.02	61.02	2.58	3.52
san200_0.9_3.clq	200	17910	90.00	44	44	44	45.03	45.03	1.75	2.16
sanr200_0.7.clq†	200	13868	69.69	18	33*	28*	34.02	29.01	79.32	356.64
sanr200_0.9.clq†	200	17863	89.76	–	59*	57*	60.01	58.03	2.82	138.17
1-FullIns_3.col	30	100	22.99	3	3	3	4.29	4.29	0.08	0.04
1-FullIns_4.col†	93	593	13.86	3	4*	3	5.37	4.5	0.57	9.29
1-Insertions_4.col	67	232	10.49	2	2	2	3.05	3.05	0.35	0.19
1-Insertions_5.col†	202	1227	6.04	2	3*	2	4.36	4	4.72	239.36
2-FullIns_3.col	52	201	15.16	4	4	4	5.23	5.24	0.21	0.11
2-FullIns_4.col†	212	1621	7.25	4	5*	4	6.34	5.34	7.6	312.52
2-Insertions_3.col	37	72	10.81	2	2	2	3.37	3.64	0.07	0.07
2-Insertions_4.col	149	541	4.91	2	2	2	3.13	3.13	2.53	1.08
anna.col	138	493	5.22	11	11	11	inf	inf	0.02	0.01
david.col	87	406	10.85	11	11	11	inf	inf	0.01	0.02
huck.col	74	301	11.14	11	11	11	inf	inf	0	0.02
jean.col	80	254	8.04	10	10	10	inf	inf	0.02	0.01
miles.col	128	387	4.76	8	8	8	inf	inf	0.04	0.05
mug100_1.col	100	166	3.35	3	3	3	4.25	4.25	0.91	0.35
mug100_25.col	100	166	3.35	3	3	3	4.27	4.27	0.8	0.44
mug88_1.col	88	146	3.81	3	3	3	4.25	4.25	0.69	0.03
mug88_25.col	88	146	3.81	3	3	3	4.28	4.28	0.61	0.27
myciel3.col	11	20	36.36	2	2	2	3.05	3.05	0.02	0.02
myciel4.col†	23	71	28.06	2	3*	2	4.34	4	0.04	0.1
myciel5.col†	47	236	21.83	2	3*	2	4.14	3.5	0.1	0.69
queen5_5.col	25	160	53.33	5	5	5	6.25	6.25	0.05	0.05
queen6_6.col†	36	290	46.03	6	7*	6	8.17	7.23	0.05	0.25
adjnoun.graph	112	425	6.84	5	5	5	6.16	6.16	0.35	0.24
as_22july06.graph	22963	48436	0.02	17	17	17	18.08	18.08	2	1.21
astro-ph.graph	16706	121251	0.09	57	57	57	inf	inf	0.03	0.09
celegans_metabolic.graph	453	2025	1.98	9	9	9	10.09	10.09	0.12	0.13
chesapeake.graph	39	170	22.94	5	5	5	6.25	6.25	0.05	0.04
cnr-2000.graph	325557	2738969	0.01	84	84	84	inf	inf	4.02	7.84
coAuthorsCiteseer.graph	227320	814134	0.00	87	87	87	inf	inf	0.09	0.21
coAuthorsDBLP.graph	299067	977676	0.00	115	115	115	inf	inf	0.12	0.39
cond-mat-2005.graph	40421	175691	0.02	30	30	30	inf	inf	0.02	0.04
dolphins.graph	62	159	8.41	5	5	5	inf	inf	0.08	0.08
email.graph	1133	5451	0.85	12	12	12	inf	inf	0.03	0.03
football.graph	115	613	9.35	9	9	9	inf	inf	1.21	0.69
ieeebus.graph	118	179	2.59	4	4	4	inf	inf	0.01	0.02
jazz.graph	198	2742	14.06	30	30	30	inf	inf	0.03	0.04
karate.graph	34	78	13.90	5	5	5	inf	inf	0.02	0.03
kreb.graph	62	153	8.09	6	6	6	inf	inf	0.03	0.02
memplus.graph	17758	54196	0.03	97	97	97	inf	inf	0.01	0.1
PGPgiantcompo.graph	10680	24316	0.04	25	25	25	26.07	26.07	1.63	1.01
polbooks.graph	105	441	8.08	6	6	6	7.2	7.2	0.26	0.13
power.graph	4941	6594	0.05	6	6	6	inf	inf	0.03	0.01
rgg_n_2_17_s0.graph	131072	728753	0.01	15	15	15	inf	inf	0.05	0.1
rgg_n_2_19_s0.graph	524288	3269766	0.00	18	18	18	inf	inf	0.53	0.42
rgg_n_2_20_s0.graph	1048576	6891620	0.00	17	17	17	18.04	18.04	1.83	3.45

Table 5.2: Comparison with upper bounds using the LP relaxation of the edge formulation.

Instance	Vertices	$\omega(G)$	UpperBound		
			UB_1	UB_2	UB_e
mann_a9.clq	45	16	18*	17*	18*
c-fat200-1.clq	200	12	12	12	12
c-fat200-2.clq	200	24	24	24	24
c-fat200-5.clq	200	58	66*	58	66*
johnson8-2-4.clq	28	4	4	4	4
johnson8-4-4.clq	70	14	14	14	14
johnson16-2-4.clq	120	8	8	8	8
hamming6-2.clq	64	32	32	32	32
hamming6-4.clq	64	4	5*	4	5
hamming8-2.clq	256	128	1128	128	128
hamming8-4.clq	256	16	16	16	16
san200_0.9_1.clq	200	70	70	70	70
san200_0.9_2.clq	200	60	60	60	60
san200_0.9_3.clq	200	44	44	44	44
anna.col	138	11	11	11	11
david.col	87	11	11	11	11
huck.col	74	11	11	11	11
jean.col	80	10	10	10	10
miles.col	128	8	8	8	8
myciel3.col	11	2	2	2	2
myciel4.col	23	2	3*	2	3*
myciel5.col	47	2	3*	2	3*
mug88_1.col	88	3	3	3	12*
mug88_25.col	88	3	3	3	12*
mug100_1.col	100	3	3	3	14*
mug100_25.col	100	3	3	3	14*
1-FullIns_3.col	30	3	3	3	3
1-FullIns_4.col	93	3	4*	3	5*
1-FullIns_5.col	282	3	4*	3	6*
1-Insertions_4.col	67	2	2	2	2
1-Insertions_5.col	202	2	3*	2	3*
2-FullIns_3.col	52	4	4	4	5*
2-FullIns_4.col	212	4	5*	4	6*
2-Insertions_3.col	37	2	2	2	2
2-Insertions_4.col	149	2	2	2	2
queen5_5.col	25	5	5	5	5
queen6_6.col	36	6	7*	6	7*

Table 5.3: Comparison with SDP-based bounds [31].

Instance	Vertices	$\omega(G)$	UB_1	UB_2	SDP
mann_a9.clq	45	16	18	17**	17.475**
mann_a27.clq	378	126	135	131**	132.7629
mann_a45.clq	1035	343	360	354**	356
johnson8-2-4.clq	28	4	4	4	4
johnson8-4-4.clq	70	14	14	14	14
johnson16-2-4.clq	120	8	8	8	8
san200_0.9_1.clq	200	70	70	70	70
san200_0.9_2.clq	200	60	60	60	60
san200_0.9_3.clq	200	44	44	44	44
sanr200_0.9.clq	200	-	59	57	49.2735**

Table 5.4: Comparison with Ellipsoidal [37] and SDP [29] based bounds.

Instance	Vertices	$\omega(G)$	UpperBound		
			UB_2	E_1	N-VI
brock200_1.clq	200	21	33	27.79**	-
brock200_2.clq	200	12	16	14.32**	-
brock200_3.clq	200	15	21	19**	-
brock200_4.clq	200	17	24	21.52**	-
c125.9.clq	125	34	40	38.05	36.23**
c250.9.clq	250	44	68	57.41**	-
c-fa200-5.clq	200	58	58**	60.36	-
DSJC125.1.clq	125	4	4**	-	4.04**
DSJC125.5.clq	125	10	13	11.48**	11.46**
DSJC125.9.clq	125	34	41	38.44	36.01**
keller4.clq	171	11	14	14.09	13.15**
mann_a27	378	126	131**	132.88	-
mann_a9.clq	45	16	17**	-	17.29**
p_hat300-3.clq	300	36	52	41.66**	-
san200_0.7_2.clq	200	18	18**	18.1**	-
sanr200_0.7.clq	200	18	28	24**	-
sanr200_0.9.clq	200	42	57	49.77**	-

5.3.1 Comparison with Existing Approaches

We compare the upper bounds UB_1, UB_2 with upper bound UB_e found by solving the LP relaxation of the edge formulation for the maximum clique problem given by (5.18). We strengthen the fractional edge polytope by adding $O(n^2)$ maximal independent set inequalities, given by (5.17). To make the comparison fair, we use the same $O(n^2)$ maximal independent sets generated by Heurisic 3.

Table 5.2 shows the bounds UB_1, UB_2 found by the proposed relaxations and UB_e found by solving the relaxation (5.18) of the edge formulation after adding $O(n^2)$ maximal independent set inequalities (5.17). In all the cases presented, the bounds found by the proposed relaxations are as good or better than UB_e. Also, the proposed relaxations can solve very large sparse instances by employing the peeling-based preprocessing.

Gruber and Rendl [31] presented a semidefinite programming (SDP) based approach for approximating the independence number. The approach solves SDP relaxations to improve the initial bound obtained by solving the stable set relaxation introduced by Lovász [38]. We compare their bounds with the bounds UB_1, UB_2 proposed by our relaxation on a subset of the DIMACS instances for which the results were reported in [31]. Table 5.3 provides the bounds, where the column 'SDP' refer to the bounds from their SDP relaxation. In this table, '$**$' denotes that the bound is best among all the bounds, but still is not the optimal one. From the table, it is clear that for all but one of the instances compared, the proposed relaxation is as good or better than the SDP relaxation.

Finally, we compare the bounds presented here with the results from [37] and [29]. Locatelli [37], improved Lovász theta number and its variants by adding non-valid inequalities, however they were able to solve instances of size less than 200 due to memory issues, and only reported the average computation time. Giandomenico et al. [29] developed a branch and cut algorithm, that uses cutting planes derived by constructing an ellipsoid containing the stable set polytope, and reported some preliminary experimental results with a time limit of about 3 hours. Table 5.4 shows the comparison of bounds for the instances mentioned in [37] and [29] mentioned respectively by, N-VI and E_1, with UB_2. Since UB_2 was better than $UB1$, we omit UB_1 from this table. In the table, '-' denotes that bound for the instance was not reported, and '$**$' denotes that the bound is best among all. From Table 5.4, it can be observed that for most of the instances, the gap between the bounds found by the proposed relaxation for the minimum k-core/2-club model and the bounds found by the ellipsoid method is not very large, and in a few cases our bounds out perform them. It was observed that though on an average the computation times of the ellipsoid method and the proposed relaxations looked similar, for the instances when the gap between the bounds is large, their computation time was significantly larger than the time reported by us. In addition, the proposed relaxations can solve large graphs of size greater than 1000 as evident from Table 5.1. The inferior bounds found by the proposed methods may be attributed to the fact that, these are simple linear relaxations, strengthened with a fixed number of maximal independent set inequalities. The bounds obtained may be improved by the addition of more valid inequalities, but then the computational time may also increase. Hence, there is a trade off between the quality of bounds produced and computation time.

5.4 Conclusion

This work proposes LP-based methods that take advantage of k-core and k-core/2-club models to find a tight upper bound for the maximum clique problem. We discuss some basic properties of the two models and prove that the upper bounds found by the proposed methods are better than the standard linear programming relaxations for the maximum clique problem. We computationally show that the upper bounds found by the proposed relaxations are of good quality, and provide a comparison with some existing bounds from the literature. It is seen that our upper bounds are as good as some of the existing bounds, and that they are computationally easy to find. The fact that the proposed method relies on the existence of a good lower bound for the maximum clique problem may not be an issue, as there are many heuristics that provide good lower bounds, and our method can verify the optimality of the clique number proposed by many algorithms. An interesting avenue for future work is to see if a tighter convex relaxation can be developed for the minimum k-core and k-core/2-club models, that can enhance the bounds found by the proposed LP relaxations.

6. CONCLUSION AND FUTURE WORK

Graph clustering has gained immense popularity over the recent years, and a number of techniques have been developed that has made its applicability widen across many domains. This dissertation considers graph theoretic relaxations of clique for characterizing structurally cohesive and robust clusters, and modeling stable clusters. While other models like s-plex and s-clubs have been studied in the literature as models of clusters with cohesivness and reachability properties, the cluster measure used was that of familiarity in the former and diameter in the latter. Here, we consider models that encapsulate structures that are highly connected, robust and stable. In particular, we study the k-block, k-robust s-club, and s-stable cluster models, and consider optimization problems that identify different structural properties of the models, thus interpreting and detecting clusters that helps in understanding the overall network properties.

In this work, we study some basic properties exhibited by the k-block, k-robust s-club, and s-stable cluster models, and show their relationship with other cluster models. It is seen that the k-block and k-robust s-club models also inherit the property of familiarity from clique, and that the s-stable cluster model has good reachability properties. This in turn helps us to define the optimization problem for each model suitably, that is, we select minimization problems for the k-block and k-robust s-clubs, and maximization for s-stable clusters, that enables us in modeling both closely-knit clusters that are very well connected with good reachability properties, and large stable clusters, whose reachability is based on its connectivity.

When studying models for clustering, it is always desirable to have polynomial time algorithms that detect the clusters of interest easily. In this regard, we study the computational complexities of finding minimum k-blocks, minimum k-robust s-clubs and maximum s-stable clusters in graphs. We establish that minimization problems of k-block, for $k \geq 3$ and k-robust 2-club for $k \geq 4$ are very hard to approximate, even when the given graph itself is a k-block and a k-robust 2-club respectively. In addition, we see that, the augmentation problems which are generalizations of these problems, are also hard to approximate, and establish that the domination version of both these problems are hard to approximate as well. The decision version of the s-stable cluster problem is shown to be NP-complete trivially for any arbitrary s, and we establish that the maximum s-stable cluster problem is NP-hard on claw-free graphs.

The complexity results on arbitrary graphs give rise to some questions which need to be addressed in future research. Firstly, the hardness of approximation results established for the minimum k-block and k-robust 2-club problems, does not rule out a constant factor approximation for these problems, and it would be interesting to find a characterization of graphs for which the maximal k-blocks are always minimum. In addition, models which have the properties of connectivity and robustness are very applicable in the field of information and communication networks, and these networks are in general modeled using unit-disk graphs. Hence, it would be very interesting to establish complexity results of the minimization problems considered on unit-disk graphs.

Our study shows the various structural and polyhderal similarities that the s-stable cluster model enjoys with clique. It would be interesting to find the classes of graphs in which the maximum s-stable cluster problem is easy to solve, which will aid in understanding the relationship between the two problems further. For instance, the maximum clique problem can be solved polynomially on unit-disk graphs [12] and perfect graphs [30], and it would be interesting to see if the maximum s-stable cluster problem exhibit the same property.

As a part of our solution approaches towards the optimization problems of interest, we propose mixed integer formulations for the minimum k-block, k-robust 2-club problems, and the maximum s-stable cluster problem. Formulations based on cutset and multi-commodity flow are established for the minimum k-block problem, and a simple diameter based formulation is presented for the minimum k-robust 2-club problem. Valid inequalities and conditions under which they will be facet inducing are discussed for both the associated polytopes. The formulations are extended to their respective augmentation problems. Though, a detailed polyhedral study is conducted for both the problems, there are still certain gaps which needs to be filled in both cases. For the k-block and k-robust 2-club models, we present a class of graphs for which the polytope is full dimensional, it would be interesting to see if there are any other classes of graphs for which this is true, and this might lead to a whole new set of facets, which can then be used for developing a branch and cut algorithm.

Two alternative formulations, similar to the maximum clique problem, are presented for the maximum s-stable cluster problem, and some of the results presented for the vertex packing problem are analyzed to be extended. In particular, local optimality conditions are generalized for the maximum s-stable cluster problem. We introduce different classes of facets, in particular the rank inequality introduced by Chvátal [11], and provide a complete description of the s-stable cluster polytope for certain classes of graphs. A very interesting point to note in our polyhedral study is that, the maximal independent set inequality seem to be facet defining for all the clique relaxation models presented here, albeit under certain conditions. As mentioned earlier in this chapter, it would be interesting to see if the same properties that holds true for the classical maximum clique problem, is also true for the s-stable cluster model. In particular, it would be interesting to check other classes of graphs for which a complete description of the clique polytope is possible. It was also observed earlier in Chapter 4 that the LP relaxation of the maximum s-stable cluster problem is very weak, unless the maximal independent set inequalities are added to it. An interesting question is, can a standard LP model be generated with a fixed number of maximal independent set inequalities for any arbitrary s, that will be a strong relaxation when used in the branch and cut algorithms.

We present two exact algorithms, a branch and cut and a combinatorial branch and bound, for the maximum s-stable cluster problem. We analyze the performance of the maximal independent set inequalities in the branch and cut algorithm, and adapt a combinatorial branch and bound algorithm meant for detecting optimal hereditary structures. Results show the performance of the algorithm on dense and sparse graphs. For the combinatorial branch and bound, we provide a simpler verification procedure when $s = 2$. However, it should be noted that the complexity of the verification procedure is $O(n^s)$, which is not very good when s is large. Hence, there is a need to develop a much simple verification procedure that will be suitable and easy to compute for large values of s. The branch and cut algorithm presented, uses the solver options for node selection, which can be a direction of exploration. The algorithm can be enhanced by choosing better node selection methods, as this was the case with the maximum clique problem. Also, the performance of the algorithm after the addition of all facets, namely, cycles and holes, needs to be evaluated. This involves choosing the order of adding violated cuts, which may lead to interesting results.

We solve the multi-commodity flow formulation for the minimum k-block problem, and the diameter formulation for minimum k-robust 2-club problem directly after some preprocessing. It should be noted that branch and cut algorithm, if developed using the described facets, will perform well only for certain classes of graphs.

Hence, we propose a more heuristic based approach for these problems. The lack of hereditary property in these structures should be made note of when developing good heuristics for the problems, and the suggested preprocessing techniques such as local 2-neighborhood search can also be used for developing these heuristics.

The final part of our research work focuses on developing good upper bounds for the maximum clique problem. We propose a duality-like approach, where we solve a minimization problem to get an upper bound for a maximization problem. We use the minimum k-core and k-core/2-club problems for this purpose, and present LP relaxations for the same. We show that the proposed methods give better bounds than the standard relaxations for the maximum clique problem. It is also computationally shown that our bounds are better than some existing bounds. It would be interesting to see if we can develop tighter convex relaxations for these two problems, and analyze if other clique relaxation models can be used for developing good upper bounds for the maximum clique problem.

REFERENCES

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

[2] J. Abello, M.G.C. Resende, and S. Sudarsky. Massive quasi-clique detection. In S. Rajsbaum, editor, *LATIN 2002: Theoretical Informatics*, pages 598–612. Springer-Verlag, London, 2002.

[3] P. Alimonti and V. Kann. Some apx-completeness results for cubic graphs. *Theoretical Computer Science*, 237:123–134, 2000.

[4] O. Amini, D. Peleg, S. Pérennes, I. Sau, and S. Saurabh. On the approximability of some degree-constrained subgraph problems. *Discrete Applied Mathematics*, 160:1661–1679, 2012.

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

[6] B. Balasundaram, S. Butenko, and S. Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10:23–39, 2005.

[7] I.M. Bomze, F. Frommlet, and M. Locatelli. Copositivity cuts for improving sdp bounds on the clique number. *Mathematical Programming*, 124:13–32, 2010.

[8] C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

[9] S. Busygin, S. Butenko, and P.M. Pardalos. A heuristic for the maximum

independent set problem based on optimization of a quadratic over a sphere. *Journal of Combinatorial Optimization*, 6(3):287–297, 2002.

[10] G. Chartrand, A. Kaugars, and D. R. Lick. Critically n-connected graphs. *Proceedings of the American Mathematical Society*, 32:63–68, 1972.

[11] V. Chvátal. On certain polytopes associated with graphs. *Journal of Combinatorial Theory (B)*, 18:138–154, 1975.

[12] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

[13] COLOR02/03/04. Graph Coloring and its Generalizaions, 2014. Online: http://mat.gsia.cmu.edu/COLOR03/. Accessed July 2015.

[14] W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver. *Combinatorial Optimization.* John Wiley and Sons, New York, 1998.

[15] R. Diestel. *Graph Theory.* Springer-Verlag, 2000.

[16] Dimacs. Cliques, Coloring, and Satisfiability: Second Dimacs Implementation Challenge, 1995. Online: http://dimacs.rutgers.edu/Challenges/. Accessed July 2015.

[17] Dimacs. Graph partitioning and graph clustering: Tenth Dimacs Implementation Challenge, 2011. Online: http://www.cc.gatech.edu/dimacs10/index.shtml. Accessed July 2015.

[18] D. Eppstein and D. Strash. Listing all maximal cliques in large sparse real-world graphs. In P.M. Pardalos and S. Rebennack, editors, *Experimental Algorithms*, volume 6630 of *Lecture Notes in Computer Science*, pages 364–375. Springer Berlin Heidelberg, 2011.

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

[20] S Fajtlowicz and C.E Larson. Graph-theoretic independence as a predictor of

fullerene stability. *Chemical Physics Letters*, 377(56):485–490, 2003.

[21] R. Faudree, E. Flandrin, and Z. Ryjek. Claw-free graphs–a survey. *Discrete Mathematics*, 164(1):87–147, 1997.

[22] R.J. Faudree, R.J. Gould, and J.S. Powell. Property $p_{d,m}$ and efficient design of reliable networks. *Networks*, 60(3):167–178, 2012.

[23] U. Feige. Vertex cover is hardest to approximate on regular graphs. Technical Report MCS 03-15, Weizmann Institute, 2003.

[24] FICO. FICO Xpress Optimization Suite 7.7, 2014. Online: http://www.fico.com/en/products/fico-xpress-optimization-suite/. Accessed July 2015.

[25] P.W. Fowler, S. Daugherty, and W. Myrvold. Independence number and fullerene stability. *Chemical Physics Letters*, 448(13):75–82, 2007.

[26] N.E. Friedkin. Structural cohesion and equivalence explanations of social homogeneity. *Sociological Methods & Research*, 12:235–261, 1984.

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

[28] M. Gendreau, P. Soriano, and L. Salvail. Solving the maximum clique problem using a tabu search approach. *Annals of Operations Research*, 41:385–403, 1993.

[29] M. Giandomenico, A.N. Letchford, F. Rossi, and S. Smriglio. A new approach to the stable set problem based on ellipsoids. In *Integer Programming and Combinatoral Optimization*, volume 6655 of *Lecture Notes in Computer Science*, pages 223–234. Springer Berlin Heidelberg, 2011.

[30] M. Grötschel, L. Lovász, and A. Schrijver. Polynomial algorithms for perfect graphs. *North-Holland mathematics studies*, 88:325–356, 1984.

[31] G. Gruber and F. Rendl. Computational experience with stable set relaxations. *SIAM Journal on Optimization*, 13(4):1014–1028, 2003.

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

[33] J. Peña, J. Vera, and L.F. Zuluaga. Computing the stability number of a graph via linear and semidefinite programming. *SIAM Journal on Optimization*, 18:87–105, 2007.

[34] L. M. Kirousis, M. Serna, and P. Spirakis. Parallel complexity of the connected subgraph problem. *SIAM Journal on Computing*, 22:573–586, 1993.

[35] D.E. Knuth. The sandwich theorem. *Electronic Journal of Combinatorics*, 1:A1, 1994.

[36] S-C Liaw and G.J. Chang. Generalized diameters and rabin numbers of networks. *Journal of Combinatorial Optimization*, 2(4):371–384, 1998.

[37] M. Locatelli. Improving upper bounds for the clique number by non-valid inequalities. *Mathematical Programming*, 150:511–525, 2015.

[38] L. Lovász. On the shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25(1):1–7, 2006.

[39] C. Ma, D. Kim, Y. Wang, W. Wang, N. Sohaee, and W. Wu. Hardness of k-vertex-connected subgraph augmentation problem. *Journal of Combinatorial Optimization*, 20(3):249–258, 2010.

[40] D. W. Matula. k-blocks and ultrablocks in graphs. *Journal of Combinatorial Theory, Series B*, 24:1–13, 1978.

[41] B. McClosky and I.V. Hicks. The co-2-plex polytope and integral systems. *SIAM Journal on Discrete Mathematics*, 23(3):1135–1148, 2009.

[42] R.J. Mokken. Cliques, clubs and clans. *Quality and Quantity*, 13:161–173, 1979.

[43] J. Moody and D. R. White. Structural cohesion and embeddedness: A hierarchical concept of social groups. *American Sociological Review*, 68:103–127, 2003.

[44] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, 1999.

[45] G.L. Nemhauser and Jr. Trotter, L.E. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8(1):232–248, 1975.

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

[47] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.

[48] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[49] P. M. Pardalos and G. P. Rodgers. A branch and bound algorithm for the maximum clique problem. *Computers and Operations Research*, 19:363–375, 1992.

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

[51] B. Pattabiraman, M.M.A. Patwary, A.H. Gebremedhin, W.K. Liao, and A. Choudhary. Fast algorithms for the maximum clique problem on massive sparse graphs. In *Algorithms and Models for the Web Graph*, volume 8305 of *Lecture Notes in Computer Science*, pages 156–169. Springer International Publishing, 2013.

[52] J. Pattillo, N. Youssef, and S. Butenko. On clique relaxation models in network analysis. *European Journal of Operational Research*, 226:9–18, 2013.

[53] S. Raghavan. *Formulations and Algorithms for Network Design Problems with Connectivity Requirements*. PhD thesis, Massachusetts Institute of Technology, 1995.

[54] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of NP. In *Proceedings of the tewnty-ninth annual ACM Symposium on Theory of Computing*, STOC '97, pages 475–484. ACM, 1997.

[55] SE Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27 – 64, 2007.

[56] A. Schrijver. A comparison of the Delsarte and Lovasz bounds. *IEEE Transactions on Information Theory*, 25:425–429, 1979.

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

[58] E. Tomita and T. Seki. An efficient branch-and-bound algorithm for finding a maximum clique. In *Proceedings of the fourth International Conference on Discrete Mathematics and Theoretical Computer Science*, DMTCS'03, pages 278–289. Springer-Verlag, 2003.

[59] E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28–42, 2006.

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

[61] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, New York, NY, USA, 2001.

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

[63] A. Veremyev and V. Boginski. Robustness and strong attack tolerance of low-

diameter networks. In A. Sorokin, R. Murphey, M.T. Thai, and P.M. Pardalos, editors, *Dynamics of Information Systems: Mathematical Foundations*, volume 20 of *Springer Proceedings in Mathematics & Statistics*, pages 137–156. Springer New York, 2012.

[64] A. Veremyev, O.A. Prokopyev, V. Boginski, and E. L. Pasiliao. Finding maximum subgraphs with relatively large vertex connectivity. *European Journal of Operational Research*, 239(2):349–362, 2014.

[65] A. Verma, A. Buchanan, and S. Butenko. Solving the maximum clique and vertex coloring problems on very large sparse networks. *INFORMS Journal on Computing*, 27(1):164–177, 2015.

[66] D. R. Wood. An algorithm for finding a maximum clique in a graph. *Operations Research Letters*, 21(5):211–217, 1997.

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

[68] J-H. Yin, J-S. Li, G-L. Chen, and C. Zhong. On the fault-tolerant diameter and wide diameter of omega-connected graphs. *Networks*, 45(2):88–94, 2005.