FINDING SECOND-ORDER CLUBS

By

YAJUN LU

Bachelor of Science in Industrial Engineering
Zhongyuan University of Technology
Zhengzhou, Henan, China
2008

Master of Science in Industrial Engineering
Huazhong University of Science and Technology
Wuhan, Hubei, China
2011

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2019

FINDING SECOND-ORDER CLUBS

Dissertation Approved:

Dr. Balabhaskar Balasundaram

Dissertation Advisor

Dr. Austin Buchanan

Dr. Chaoyue Zhao

Dr. Jay Schweig

Dedicated to my

beloved parents, Kekuan Lu and Chunling Zhan,

beloved wife, Qian Gao,

beloved son, Jackie Lu,

and beloved daughter, Cathy Lu.

The dedication reflects the views of the author and are not endorsed by committee members or Oklahoma State University.

ACKNOWLEDGMENTS

First and foremost, it is my honor to be Dr. Balabhaskar Balasundaram's Ph.D. student. I would like to express my sincere gratitude to Dr. Balasundaram; he was one of the biggest personal influences on my academic studies as well as my life. He used his own practical actions to demonstrate what a great professor should do and how to work with students. He has patience, motivation, and immense knowledge. My dissertation could not have been completed without his guidance. His guidance helped me throughout my Ph.D. life. During my studies, whenever I needed help, he not only helped me to find the answers but also taught me how to solve the problem. I really appreciate that he always used consideration and respect in telling me what I should do and gave me appropriate criticism and praise. These greatly helped me develop confidence. I worked with Dr. Balasundaram for only a few years, but his influence on me is permanent. I will bring his best qualities to my future work.

I am deeply indebted to my committee members, Dr. Austin Buchanan, Dr. Chaoyue Zhao and Dr. Jay Schweig for their guidance, insightful comments and encouragement throughout my doctoral research and coursework. Without their support and help, I could not have completed my dissertation. Additionally, I thank Dr. Austin Buchanan and Dr. Sunderesh Heragu for their generous help during my academic job search.

I am very grateful to the faculty and staff of Industrial Engineering and Management (IEM) department, especially, Dr. Sunderesh Heragu, Dr. Tieming Liu, Dr. Manjunath Kamath, Dr. Farzad Yousefian, Dr. Juan Borrero for their help throughout my Ph.D. study. In addition,

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

special thanks are due to Dr. Tieming Liu, Dr. Sunderesh Heragu and Dr. Manjunath Kamath who provided kind and insightful advice when I was faced with difficult choices. I would also like to thank Ms. Laura Brown for her endless support and help since I joined the department.

Stillwater Toastmasters Club left me with many wonderful memories. In this club, the seasoned members, Dr. Deke Johnson, and Dr. Marley Beem were enthusiastic and patient in helping me improve my public speaking and leadership skills. I express my warm and sincere thanks to both of them and other members in the club. Financial support for my Ph.D. study came from the IEM department, the National Science Foundation (Award Number: 1404971), and industry collaborative project with Bay Valley Foods, LLC. I would like to thank them for their generous support. The computational experiments reported in this dissertation were conducted at the Oklahoma State University High-Performance Computing Center.

I would like to thank my friends, just to name a few, Hao Pan, Akash Gupta, Sadra Babaei, Hosseinali Salemi, Hamidreza Validi, Ronny Pacheco, Devaraja Radha Krishnan, Hang Zeng, Dr. Xiao Sun, Tengfei Sun, Mostafa Amini, Dr. Zhuqi Miao, Dr. Shuzhen Sun, Dr. Juan Ma, Dr. Saeed Piri, Dr. Ali Bagheri, and Christina Anaya for making my life colorful at Stillwater.

Last but not least, I thank my parents Kekuan Lu and Chunling Zhan, my wife Qian Gao, two lovely kids, and two brothers. From my first day in school, my parents worked hard and tried their best to provide me better education opportunities. Without their support, encouragement and selfless dedication, I could not have become who I am. My wife did everything she could to support my Ph.D. study and always encouraged me. My two lovely kids gave me the passion and motivation to complete my graduate program. I am grateful to my family for their constant support throughout my program.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: YAJUN LU

Date of Degree: July, 2019

Title of Study: FINDING SECOND-ORDER CLUBS

Major Field: INDUSTRIAL ENGINEERING AND MANAGEMENT

Abstract: Modeling data entities and their pairwise relationships as a graph is a popular technique to visualizing and mining information from datasets in a variety of fields such as social networks, biological networks, web graphs, and document networks. A powerful technique in this setting involves the detection of clusters. Clique, a subset of pairwise adjacent vertices, is often viewed as an idealized representation of a cluster. However, in the presence of errors in the data on which the graph is based, clique requirement may be too restrictive, resulting in small clusters or clusters that miss key members. Consequently, graph-theoretic clique generalizations based on the principle of relaxing elementary structural properties of a clique have been proposed in diverse fields to describe clusters of interest. For example, an s-club is a distance-based clique relaxation originally introduced in social network analysis to model cohesive social subgroups. In this dissertation, we consider low-diameter clusters that require another property like robustness, heredity, or connectedness (parameterized by r) to hold, in addition to the diameter. Specifically, we study s-clubs with side-constraints to make them less "fragile", i.e., less susceptible to increase in the diameter if vertices (and edges) are deleted. The overall goal of this dissertation is to develop effective exact algorithms with an emphasis on $s = 2, 3, 4$ and low values of r to solve the maximum r-robust s-club and r-hereditary s-club problems on moderately large instances (around 10^4 vertices and less than 5% density). We analyze the complexity of the associated feasibility testing and optimization problems. Cut-like formulations are proposed for the maximum r-robust s-club problem with $r \geq 2$ and $s \in \{2, 3, 4\}$. We explore preprocessing techniques and develop a graph decomposition approach for solving such problems. The computational benefits of each of the algorithmic ideas are empirically evaluated through our computational studies. Our approach permits us to solve problems optimally on very large and sparse real-life networks.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

Introduction

Modeling data entities and their pairwise relationships as a graph is a popular approach to visualizing and mining information from datasets in a variety of fields [Cook and Holder, 2006, Porter and Howison, 2017]. Typically, the set of vertices represents individuals, organizations or entities, and the set of edges corresponds to the ties among them. A powerful approach in this setting involves the detection of clusters—either by finding those of the largest cardinality (or weight), finding those that cover or partition the graph, or by enumerating all inclusionwise maximal clusters. Before either of these combinatorial optimization problems could be solved, we need to formally define a cluster.

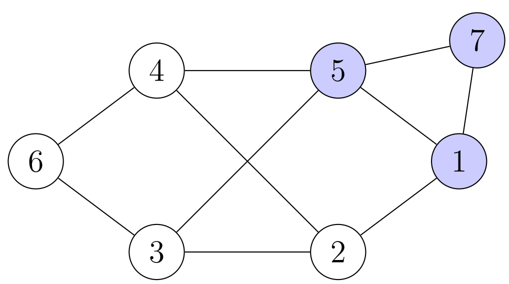

Figure 1.1: Vertices $\{1, 5, 7\}$ form a clique in the graph.

Cliques, a subset of pairwise adjacent vertices, are often viewed as an idealized representation of a cluster. Taking the graph in Figure 1.1 for instance, vertices $\{1, 5, 7\}$ form a clique in the graph. However, in the presence of errors in the data on which the graph is based, clique requirement may be too restrictive, resulting in small clusters or clusters that miss key members. Consequently, graph-theoretic clique generalizations based on the principle of relaxing elementary structural properties of a clique have been proposed in diverse fields to describe clusters of interest [Pattillo et al., 2013b]. For instance, a clique can be equivalently defined as a subset of vertices with pairwise distances at most one. Increasing the pairwise distance bound leads to a distance-based "clique relaxation." The more well-studied relaxations of cliques can be typically grouped into three categories.

(i) Degree based relaxations such as k-core and k-plex [Balasundaram et al., 2011, McClosky and Hicks, 2012, Guo et al., 2010];

(ii) Edge density based relaxations such as quasi-clique [Pattillo et al., 2013a, Pajouh et al., 2014];

(iii) Distanced based relaxations e.g. k-clique and k-club [Shahinpour and Butenko, 2013b, Bourjolly et al., 2002, Luce, 1950].

Such models are less sensitive than the clique model to edges that may be missing in the graph due to erroneous or incomplete underlying data. Before reviewing these ideas, we introduce the necessary notations next.

1.1 Notations and definitions

We list some of the notations used and recall some basic graph theory concepts (Refer [Diestel, 1997] or [Bondy and Murty, 2008] for additional details). We are given a graph $G = (V, E)$ of order n and size m with a vertex set $V := [n]$ and the edge set $E \subseteq \binom{V}{2}$, where $[n] := \{1, \ldots, n\}$ for any positive integer n and $\binom{V}{2} := \{\{u, v\} : u, v \in V, u \neq v\}$. When left unspecified, we use $V(G)$ and $E(G)$ to denote the vertex and edge sets of G, respectively. The complete graph on n vertices is denoted by K_n, the complete bipartite graph with partitions of size p and q by $K_{p,q}$, the n-vertex path is denoted P_n, and the n-cycle is denoted C_n. Given two disjoint graphs H and G, the graph join operation, denoted as $H * G$, produces the graph that "combines" G and H by joining every vertex from G and every vertex from H using a new edge. Formally,

$V(H*G)=V(H)\cup V(G)$ and $E(H*G)=E(H)\cup E(G)\cup\{\{u,v\}:u\in V(H),v\in V(G)\}$. If two graphs G and H are isomorphs, we denote that by $G\simeq H$. We denote by $G[S]$, the subgraph induced by a subset of vertices S.

The distance between a pair of vertices i and j in G, which is denoted by $d_G(i,j)$, refers to the length of the shortest path between vertices i and j. Let $\mathrm{diam}(G)$ denote the diameter of G representing the maximum distance between any pair of vertices in a connected graph G. For each vertex $v\in V$, we denote by $N_G(v)$, the *open* neighborhood of v, i.e., the set of all adjacent vertices of v. The *closed* neighborhood of v includes itself, denoted by $N_G[v]=N_G(v)\cup\{v\}$. When the graph under consideration is readily apparent, we drop the subscript used to identify the graph. Analogously, given a positive integer s, we denote by $N^s_G(v)$, the set of distance-s *open* neighbors of v, and formally, $N^s_G(v)=\{u\in V:1\leq d_G(u,v)\leq s\}$. The set of distance-$s$ *closed* neighbors of v is denoted by $N^s_G[v]=\{u\in V:d_G(u,v)\leq s\}$. Let $\deg_G(v)$ denote the degree of v corresponding to the cardinality of $N_G(v)$.

In addition, we denote by $\delta(G)=\min\{\deg_G(v):v\in V\}$, the minimum vertex degree of G. For convenience, we denote by $G-v$ and $G-e$, the deletion of a vertex $v\in V$ and an edge $e\in E$ from the graph, respectively. Note that when a vertex is deleted, all its incident edges are also deleted. Similarly, we denote the deletion of a subset S of vertices (or edges) by $G-S$. We also make use of the edge indicator function as follows: $\mathbb{1}_E(i,j)=1$ if $\{i,j\}\in E$ and $\mathbb{1}_E(i,j)=0$ otherwise.

A graph G of order at least $k+1$ is said to be k-connected for a positive integer k if $G-S$ is connected for every vertex-subset S containing at most $k-1$ vertices. Every non-empty graph is 0-connected and every non-trivial connected graph is 1-connected. The largest integer k for which G is k-connected is called the connectivity of G, denoted as $\kappa(G)$. Note that for any non-trivial G, $\kappa(G)$ is zero if and only if G is disconnected, and $\kappa(G)=n-1$ if and only if G is K_n. A *block* is a maximal biconnected (i.e., 2-connected) subgraph of a given graph G. Figure 1.2 shows a graph which decomposes into two blocks. A subset

$D \subseteq V \setminus \{u, v\}$ is called a u, v-vertex separator for distinct non-adjacent vertices u and v if u is disconnected from v in the subgraph $G - D$.

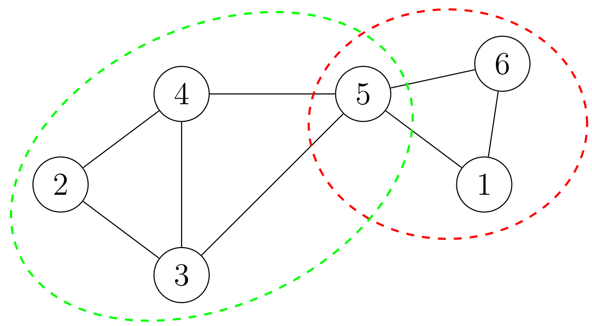

Figure 1.2: A graph that decomposes into two blocks

1.2 Motivation

Pattillo et al. [2013b] coined the term "elementary clique-defining property" for several graph properties that can equivalently define a clique. For instance, given a graph $G = (V, E)$, a subset of vertices C forms a clique if it satisfies any one of the following properties.

a) The pairwise distances between vertices in C is at most one, i.e., for every $u, v \in C$, we have $d_G(u, v) \leq 1$.

b) The diameter of the subgraph induced by C is at most one, i.e., $\mathrm{diam}(G[C]) \leq 1$.

c) The connectivity of the subgraph induced by C is at least $|C| - 1$, i.e., $\kappa(G[C]) \geq |C| - 1$.

Systematically relaxing the bounds involving the elementary clique-defining properties leads to various *clique relaxations*. Pattillo et al. [2013b] discussed the issues related to the development of such clique relaxations at great length and present a taxonomic system to classify them. One such approach is to classify them based on the *order* of the clique relaxation. The parameterized models obtained by relaxing any one elementary clique-defining property is called a first-order clique relaxation. Those that relax two such properties simultaneously are called second-order clique relaxations. Consider the following definitions.

Definition 1 (Luce [1950])**.** *Given a positive integer s, a subset of vertices S is called an s-clique if $d_G(u, v) \leq s$ for every pair of vertices $u, v \in S$.*

Definition 2 (Luce [1950], Mokken [1979])**.** *Given a positive integer s, a subset of vertices S is called an s-club if* $\text{diam}(G[S]) \leq s$*.*

Both models are distance-based first-order clique relaxations. That is, they generalize a clique by relaxing one elementary clique-defining property, and the clique is obtained as a special case when $s = 1$. Clearly, the fundamental difference between an s-clique and s-club is that the distance bound is applicable to the original graph in the former, and to the induced subgraph in the latter. Hence, every s-club is an s-clique, but not vice versa. The s-clubs are more cohesive because they guarantee that the bounded-length paths between pairs of vertices are completely contained within the cluster.

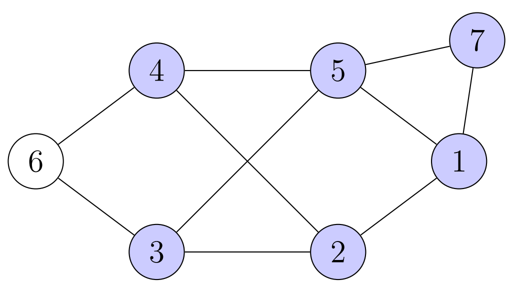

Figure 1.3: The subset $\{1, 2, 3, 4, 5, 7\}$ is the maximum 2-club in the graph.

Originally introduced to model cohesive subgroups in social networks [Luce, 1950, Mokken, 1979], s-clubs can be used to model low-diameter clusters in many different settings for smaller values of s such as social network analysis [Wasserman and Faust, 1994] and bioinformatics [Balasundaram et al., 2005]. Low value of parameter s implies that pairwise distances inside the group are short, and consequently, passing messages takes few hops/members in the s-club. In particular, the 2-club (see Figure 1.3) represents clusters in which every pair of vertices are either adjacent, or have a common neighbor inside the cluster. Hence, 2-clubs formalize the notion of a friend-of-a-friend social subgroup in which members may be directly acquainted or related through a mutual acquaintance inside the cluster. Another example is the protein interaction network (PIN) which models the proteins in an organism as vertices and their interactions as edges. Cluster detection in a PIN is used to identify protein complexes that may have important cellular functions. We refer the reader to surveys by Shahinpour and Butenko [2013a] and by Balasundaram and Pajouh [2013] for more information on s-clubs.

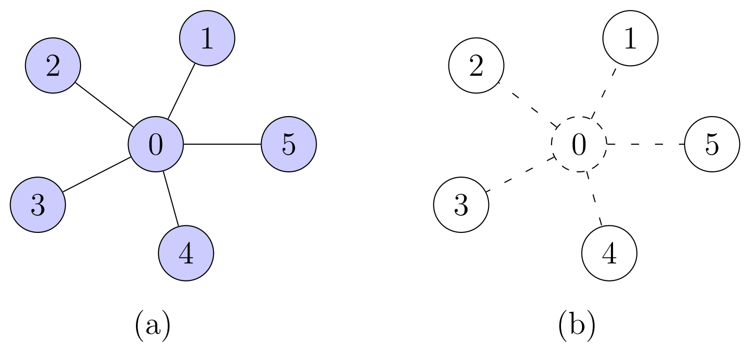

Figure 1.4: A star graph ($K_{1,5}$) whose vertex set is a 2-club. However, the subset $\{1, 2, 3, 4, 5\}$ is an independent set in the graph obtained by removing the center of the star $\{0\}$.

Although s-clubs ensure low pairwise distances inside the cluster, they may not be robust in the sense that deleting a single vertex could increase the distances, even disconnect the graph. An extreme example is a 2-club that induces a star graph ($K_{1,q}$), which results in an independent set when the center of the star is deleted from the 2-club (see Figure 1.4). In general, s-club property is not hereditary under vertex deletion for $s \geq 2$, i.e., the diameter bound may not be preserved even if the induced subgraph remains connected under vertex deletion. These observations motivated various authors to consider the following second-order clique relaxations.

Definition 3 (Veremyev and Boginski [2012]). *Given a graph G and positive integers r and s, a subset of vertices S is called an r-robust s-club if there are at least r (internally)*

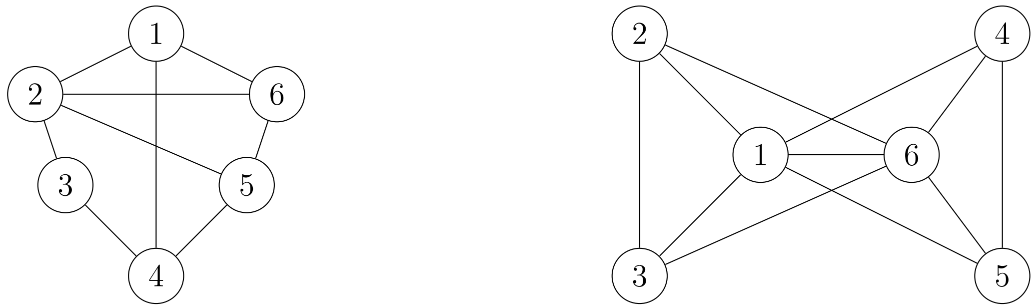

Figure 1.5: Examples about 2-club (left) and 2-robust 2-club (right)

vertex-disjoint paths of length at most s in $G[S]$ between every distinct pair of vertices in S.

Figure 1.5 illustrates the difference between a 2-club and a 2-robust 2-club. The graph on the left in Figure 1.5 is a 2-club but it is not a 2-robust 2-club, since there is only one vertex-disjoint path of length at most two between the vertices 1 and 4. The graph on the right in Figure 1.5 is a 2-robust 2-club, in which there are at least two vertex-disjoint paths of length at most two between every pair of vertices.

By Definition 3, every r-robust s-club S must contain at least $r+1$ vertices except when the definition is trivially satisfied, i.e., $|S| \leq 1$. Furthermore, the only r-robust s-clubs that contain *exactly* $r+1$ vertices are $(r+1)$-vertex cliques. As long as the r-robust s-club contains two vertices that are not adjacent, it must contain at least $r+2$ vertices.

Definition 4 (Pattillo et al. [2013b])**.** *Given a graph G and positive integers r and s, a subset of vertices S is called an r-hereditary s-club if* $\text{diam}(G[S \setminus T]) \leq s$ *for all $T \subseteq S$ with $|T| < r$.*

In Definition 4, we say S admits *deletion sets* with fewer than r vertices. Note that if $r = 1$, Definitions 3 and 4 coincide with Definition 2 for every positive integer s, i.e., every s-club is both 1-hereditary and 1-robust. We note that Definition 4 deviates slightly from the original definition of Pattillo et al. [2013b], which allowed deletion sets up to size r. Our redefinition is more convenient to use especially when working with both models simultaneously. When r is at least two, Definitions 3 and 4 diverge as we discuss next.

Lemma 1. *Consider a graph $G = (V, E)$, and integers $s \geq 2$ and $r \geq 2$. If $S \subseteq V$ is an r-robust s-club then S is r-hereditary.*

Proof. Suppose S is an r-robust s-club with $|S| \geq r + 2$, as the claim is trivial otherwise. Consider a pair of vertices $u, v \in S$, which by Definition 3 are connected by at least r internally vertex-disjoint paths in $G[S]$ of length at most s. Any deletion set T with less than r vertices can disconnect at most $r - 1$ of those paths. Hence, $d_{G[S \setminus T]}(u, v) = d_{G[S]}(u, v) \leq s$. $\square$

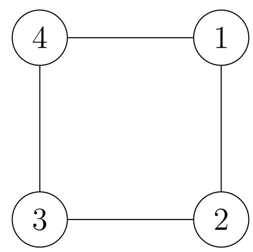

Figure 1.6: The vertex set of graph C_4 is a 2-hereditary 2-club, but it is not a 2-robust 2-club.

It is natural to consider a converse of Lemma 1 that says an r-hereditary s-club S with at least $r + 1$ vertices is also r-robust. However, that is not true. Consider a 4-cycle (see Figure 1.6), the vertex set of which is a 2-hereditary 2-club that contains more than 3 vertices, but it is not 2-robust. The distinction stems from the fact that adjacent pairs of vertices in an r-hereditary s-club are not required to satisfy any additional requirements—their pairwise distance remains the same under any number of vertex deletions, not including the end-points themselves. However, in an r-robust s-club, adjacent vertex pairs still need to be connected by at least $r - 1$ additional vertex-disjoint paths of length s or less.

Definition 5 (Yezerska et al. [2017])**.** *Given a graph G and positive integers r and s, a subset of vertices S is called an r-connected s-club if* $\mathrm{diam}(G[S]) \leq s$ *and* $\kappa(G[S]) \geq r$.

By Menger's Theorem [Menger, 1927], every non-trivial s-club is 1-connected, r-hereditary and r-robust s-clubs are r-connected. By contrast, although the vertices of a 5-cycle form a 2-connected 2-club, it is neither a 2-hereditary 2-club nor a 2-robust 2-club (see Figure 1.7).

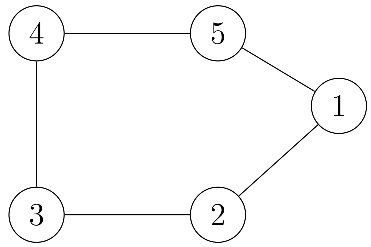

Figure 1.7: The vertex set of C_5 forms a 2-connected 2-club, but it is neither a 2-hereditary 2-club nor a 2-robust 2-club.

Note that deleting any vertex in a C_5 will increase the diameter, and every distinct pair of vertices have exactly one path of length at most two. In this dissertation, we focus on the r-robust s-club model and wherever possible extend our approaches to r-hereditary s-club and r-connected s-club.

CHAPTER II

Literature Review

The focus of this dissertation is on combinatorial optimization problems seeking a maximum cardinality s-club that also satisfies an additional property of robustness, heredity, or connectivity following Definitions 3, 4, and 5 respectively, the so-called second-order s-clubs. In this chapter we review relevant literature on computational complexity, integer programming (IP) formulations, preprocessing techniques, and exact algorithms for closely related problems. Based on the literature review, we identify some research gaps and thus develop our research statement.

2.1 The maximum s-club problem

The maximum s-club problem is to seek an s-club of maximum cardinality. First, we present the decision version of the maximum s-club problem.

Problem: s-Club (for integer constant s).

Input: Graph G and a positive integer c.

Question: Does G contain an s-club of size at least c?

Bourjolly et al. [2002] showed that s-Club is NP-complete for any constant positive integer; they remain NP-complete even when restricted to graphs of diameter $s+1$ [Balasundaram et al., 2005]. Testing inclusionwise maximality of s-clubs is also NP-hard [Pajouh and Balasundaram, 2012]. The s-Club remains NP-hard on 4-chordal graphs for every positive integer s [Golovach et al., 2014], bipartite graphs for every fixed $s \geq 3$, and on chordal graphs for every *even* fixed $s \geq 2$ [Asahiro et al., 2010]. However, s-Club is polynomial-time

Table 2.1: Complexity results related to s-Club and 2-Club

Model	Key results
s-Club	**NP-complete** for every positive integer s [Bourjolly et al., 2002], even when restricted to graphs of diameter $s+1$ [Balasundaram et al., 2005] **NP-hard** on 4-chordal graphs for every positive integer s [Golovach et al., 2014], bipartite graphs for every fixed $s \geq 3$, and on chordal graphs for every even fixed $s \geq 2$ [Asahiro et al., 2010]. Testing inclusionwise maximality of s-clubs is also NP-hard [Pajouh and Balasundaram, 2012] **Polynomial-time** solvable on trees, interval graphs, and graphs with bounded tree- or cliquewidth for every fixed $s \geq 1$ [Schäfer, 2009], chordal bipartite, strongly chordal and distance hereditary graphs for every fixed $s \geq 1$, AT-free graphs for all positive fixed $s \geq 2$, and weakly chordal graphs for every fixed odd s [Golovach et al., 2014] **Inapproximable** within a factor of $n^{\frac{1}{2}-\epsilon}$ in general graphs for any $\epsilon > 0$ and a fixed $s \geq 2$, and a factor of $n^{\frac{1}{3}-\epsilon}$ for chordal and split graphs with even s [Asahiro et al., 2010] **Fixed-parameter tractable** when parameterized by both solution size k and dual parameter $d = \|V\| - k$ [Schäfer et al., 2012]
2-Club	**NP-hard** on split graphs [Asahiro et al., 2010], graphs with clique cover number three and diameter three, graphs with domination number two and diameter three, graphs with distance one to bipartite graphs, and connected graphs with average degree at most α for any constant $\alpha > 2$ [Hartung et al., 2015] **Polynomial-time** solvable on bipartite graphs in $O(n^5)$ [Schäfer, 2009] **Fixed-parameter tractable** when parameterized by distance to cographs [Hartung et al., 2015]

solvable on trees, interval graphs, and graphs with bounded tree- or cliquewidth for every fixed $s \geq 1$ [Schäfer, 2009], chordal bipartite, strongly chordal and distance hereditary graphs for every fixed $s \geq 1$, AT-free graphs for all positive fixed $s \geq 2$, and weakly chordal graphs for every fixed odd s [Golovach et al., 2014]. These and other complexity results related to s-Club as well as the special case 2-Club are summarized in Table 2.1.

2.1.1 IP formulations

An IP formulation for the maximum s-club problem was first proposed by Bourjolly et al. [2002] and this so-called "chain formulation" is presented next.

$$\max \sum_{i \in V} x_i \tag{2.1}$$

$$\text{s.t. } x_i + x_j \leq \sum_{P_{ij}^t \in C_{ij}^s} y_{ij}^t + 1 \qquad \forall \{i,j\} \in \binom{V}{2} \setminus E : C_{ij}^s \neq \emptyset \tag{2.2}$$

$$y_{ij}^t \leq x_r \qquad \forall r \in V(P_{ij}^t), P_{ij}^t \in C \tag{2.3}$$

$$x_i + x_j \leq 1 \qquad \forall \{i,j\} \in \binom{V}{2} \setminus E : C_{ij}^s = \emptyset \tag{2.4}$$

$$x_i \in \{0,1\} \qquad \forall i \in V \tag{2.5}$$

$$y_{ij}^t \in \{0,1\} \qquad \forall P_{ij}^t \in C \tag{2.6}$$

In formulation (2.1)–(2.6), C_{ij}^s is the collection of all paths of length at most s linking vertices i and j in graph G and $C := \cup_{i,j \in V} C_{ij}^s$. Buchanan and Salemi [2017] developed a path-like formulation which dominates this "chain formulation".

Veremyev and Boginski [2012] proposed a compact IP formulation for the maximum s-club problem with $O(sn^2)$ variables and constraints. Compared with the $O(n^{s+1})$ worst-case size of the chain formulation (2.1)–(2.6), this model represents a significant improvement in size and is presented next.

$$\max \sum_{i \in V} x_i \tag{2.7}$$

$$\text{s.t.} \quad x_i + x_j - 1 \leq \mathbb{1}_E(i,j) + \sum_{\ell=2}^{s} z_{ij}^{\ell} \qquad i,j \in V : i < j \tag{2.8}$$

$$z_{ij}^2 \leq x_i, \quad z_{ij}^2 \leq x_j, \quad z_{ij}^2 \leq \sum_{t \in N(i) \cap N(j)} x_t \qquad i,j \in V : i < j \tag{2.9}$$

$$z_{ij}^2 \geq \frac{1}{n}\Big(\sum_{t \in N(i) \cap N(j)} x_t \Big) + (x_i + x_j - 2) \qquad i, j \in V : i < j \tag{2.10}$$

$$z_{ij}^\ell \leq x_i \quad z_{ij}^\ell \leq \sum_{t \in N(i)} z_{tj}^{\ell-1} \qquad i, j \in V : i < j, \ell = 3, \dots, s \tag{2.11}$$

$$z_{ij}^\ell \geq \frac{1}{n}\Big(\sum_{t \in N(i)} z_{tj}^{\ell-1} \Big) + (x_i - 1) \qquad i, j \in V : i < j, \ell = 3, \dots, s \tag{2.12}$$

$$z_{ij}^\ell \in \{0, 1\} \qquad \forall i, j \in V : i < j, \ell = 2, \dots, s \tag{2.13}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{2.14}$$

In formulation (2.7)–(2.14), the $z_{ij}^\ell = 1$ if and only if the internal vertices on some path of length exactly ℓ between i and j are all included in the solution containing i and j. Using binary variables u_{ij}^ℓ that take the value one if all the internal vertices on some path of length *at most* ℓ between i and j are included in the solution, Veremyev et al. [2015] reformulated the problem as follows.

$$\max \sum_{i \in V} x_i \tag{2.15}$$

$$\text{s.t.} \quad x_i + x_j - 1 \leq u_{ij}^s \qquad \forall \{i, j\} \in \binom{V}{2} \tag{2.16}$$

$$u_{ij}^1 = 0 \qquad \forall \{i, j\} \in \binom{V}{2} \setminus E \tag{2.17}$$

$$u_{ij}^1 = u_{ij}^\ell \qquad \forall \{i, j\} \in E, \ell = 2, \dots, s \tag{2.18}$$

$$u_{ij}^\ell \leq \sum_{t \in N(i)} u_{tj}^{\ell-1} \qquad \forall \{i, j\} \in \binom{V}{2} \setminus E, \ell = 2, \dots, s \tag{2.19}$$

$$u_{ij}^\ell \leq x_i, \quad u_{ij}^\ell \leq x_j \qquad \forall \{i, j\} \in \binom{V}{2}, \ell = 1, \dots, s \tag{2.20}$$

$$u_{ij}^\ell = u_{ji}^\ell \qquad \forall \{i, j\} \in \binom{V}{2}, \ell = 1, \dots, s \tag{2.21}$$

$$u_{ij}^\ell \in [0, 1] \qquad \forall \{i, j\} \in \binom{V}{2}, \ell = 1, \dots, s \tag{2.22}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{2.23}$$

Recently, a cut-like formulation for the maximum s-club problem based on length-bounded vertex separators was presented by Buchanan and Salemi [2017], and the reported computational results showed that their algorithm using this formulation outperforms directly solving other IP formulations in the literature. Before introducing this formulation, we need to define a length-s u,v-separator (see for instance Lovász et al. [1978], Buchanan and Salemi [2017]).

Definition 6. *A subset $C \subseteq V \backslash \{u, v\}$ of vertices is called a length-s u,v-separator in graph $G = (V, E)$ if $d_{G-C}(u, v) > s$.*

The cut-like formulation has $|V|$ variables and is presented next. In constraints (2.25), $\forall(u, v, C)$ is a shorthand for any pair of non-adjacent vertices u, v and all length-s u,v-separators C in graph G.

$$\max \sum_{i \in V} x_i \tag{2.24}$$

$$x_u + x_v \leq 1 + \sum_{i \in C} x_i \qquad \forall(u, v, C) \tag{2.25}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{2.26}$$

2.1.2 Exact algorithms

Bourjolly et al. [2002] developed a branch-and-bound (BB) algorithm for the maximum s-club problem. This BB algorithm obtained the upper bounds based on the computation of maximal stable sets in the power graph, and the results showed that the problem difficulty directly depended on the graph density. Pajouh and Balasundaram [2012] and Chang et al. [2013] also proposed BB algorithms for the maximum s-club problem. Recently, Moradi and Balasundaram [2018] (see also [Lu et al., 2018]) developed a decomposition and branch-and-cut (BC) algorithm by using the canonical hypercube cut (CHC) and graph decomposition techniques. Buchanan and Salemi [2017] devised a BC algorithm by employing a delayed generation of length-s u, v-separator constraints. A fixed-parameter tractable algorithm was developed by Hartung et al. [2015] for finding 2-clubs.

2.2 Second order s-clubs

Like the maximum s-club problem, we formally define decision versions of the maximum r-robust s-club problem (r, s-MRCP), maximum r-hereditary s-club problem (r, s-MHCP), and maximum r-connected s-club problem (r, s-MCCP) as follows.

Problem: r-Robust s-Club/r-Hereditary s-Club/r-Connected s-Club (for integer constants $r, s \geq 2$).

Input: Graph G and positive integer c.

Question: Does G contain an r-robust s-club/r-hereditary s-club/r-connected s-club of size at least c?

Komusiewicz et al. [2019] showed that r-Hereditary 2-Club and r-Robust 2-Club are NP-complete for every fixed positive integer $r \geq 2$. The hardness of both problems was proved by reduction from 2-Club. Yezerska et al. [2017] proved that 2-Connected 2-Club is NP-complete. Interestingly, Komusiewicz et al. [2019] also utilized a similar construction and further strengthened the result by showing that r-Connected 2-Club is NP-complete, even on split graphs, for any fixed positive integer $r \geq 1$. Table 2.2 summarizes the known complexity results related to second-order s-clubs.

2.2.1 IP formulations

Veremyev and Boginski [2012] first presented an IP formulation for a relaxation of the r, s-MRCP, in which the r paths of length at most s are only required to be distinct, and not necessarily vertex-disjoint. However, when $s = 2$, distinct paths must also be vertex disjoint,

Table 2.2: Complexity results related to second order s-clubs

Model	**Key results**
r-Robust 2-Club	**NP-complete** for every fixed positive integer $r \geq 1$, and it remains NP-complete on graphs with both diameter two and domination number one, graphs with degeneracy $6 + r$, split graphs, and connected graphs with average vertex degree at most α, for all $\alpha > 2$ [Komusiewicz et al., 2019] **Fixed-parameter tractable** when parameterized by dual parameter $\ell = \lvert V\rvert - k$ where k is solution size [Komusiewicz et al., 2019] **does not admit** a $(2-\epsilon)^{\ell}.nO^{(1)}$-time algorithm for any $\epsilon > 0$ [Komusiewicz et al., 2019]
r-Hereditary 2-Club	**NP-complete** for every fixed positive integer $r \geq 1$, and it remains NP-complete on graphs with both diameter two and domination number one, graphs with degeneracy $6 + r$, split graphs, and connected graphs with average vertex degree at most α, for all $\alpha > 2$ [Komusiewicz et al., 2019] **Fixed-parameter tractable** when parameterized by dual parameter $\ell = \lvert V\rvert - k$ where k is solution size [Komusiewicz et al., 2019] **does not admit** a $(2-\epsilon)^{\ell}.nO^{(1)}$-time algorithm for any $\epsilon > 0$ [Komusiewicz et al., 2019]
r-Connected 2-Club	**NP-complete** for any fixed positive integer $k \geq 1$, even on split graphs [Komusiewicz et al., 2019] **Fixed-parameter tractable** when parameterized by dual parameter $\ell = \lvert V\rvert - k$ where k is solution size [Komusiewicz et al., 2019] **does not admit** a $(2-\epsilon)^{\ell}.nO^{(1)}$-time algorithm for any $\epsilon > 0$ [Komusiewicz et al., 2019]

and therefore their formulation, which correctly models the special case, is presented next.

$$\max \sum_{i \in V} x_i \tag{2.27}$$

$$\text{s.t. } r(x_i + x_j - 1) \leq \mathbb{1}_E(i,j) + \sum_{k \in N(i) \cap N(j)} x_k \qquad \forall \{i,j\} \in \binom{V}{2} \tag{2.28}$$

$$x_i \in \{0,1\} \qquad \forall i \in V \tag{2.29}$$

Constraints (2.28) ensure that there exist r vertex-disjoint paths of length at most two between every pair of vertices.

For the special case $s = 3$, Almeida and Carvalho [2014] presented an IP formulation for the $r, 3$-MRCP by enumerating all vertex-disjoint paths of length at most three, which is presented next.

$$\max \sum_{i \in V} x_i \tag{2.30}$$

$$\text{s.t. } y_{pq}^{ij} \leq x_i, \quad y_{pq}^{ij} \leq x_j \qquad \forall \{i, j\} \in \mathcal{N}^3, \{p, q\} \in E_{ij} \tag{2.31}$$

$$x_i + x_j \leq 1 \qquad \forall \{i, j\} \in \mathcal{P}^3 \tag{2.32}$$

$$\sum_{q:\{p,q\} \in E_{ij}} y_{pq}^{ij} \leq x_p \qquad \forall \{i, j\} \in \mathcal{N}^3, p \in F_{ij} \tag{2.33}$$

$$\sum_{p:\{p,q\} \in E_{ij}} y_{pq}^{ij} \leq x_q \qquad \forall \{i, j\} \in \mathcal{N}^3, q \in B_{ij} \tag{2.34}$$

$$\sum_{t \in N(i) \cap N(j)} x_t + \sum_{(p,q) \in E_{ij}} y_{pq}^{ij} \geq r(x_i + x_j - 1) \qquad \forall \{i, j\} \in \mathcal{N}^3 \tag{2.35}$$

$$\sum_{t \in N(i) \cap N(j)} x_t + \sum_{\{p,q\} \in E_{ij}^1} y_{pq}^{ij} \geq (r - 1)(x_i + x_j - 1) \qquad \forall \{i, j\} \in E \tag{2.36}$$

$$y_{pq}^{ij} \leq x_i, \quad y_{pq}^{ij} \leq x_j \qquad \forall \{i, j\} \in E, \{p, q\} \in E_{ij}^1 \tag{2.37}$$

$$\sum_{q:\{p,q\} \in E_{ij}^1} y_{pq}^{ij} \leq x_p \qquad \forall \{i, j\} \in E, p \in N(i) \backslash (N(j) \cup \{j\}) \tag{2.38}$$

$$\sum_{p:\{p,q\} \in E_{ij}^1} y_{pq}^{ij} \leq x_q \qquad \forall \{i, j\} \in E, q \in N(j) \backslash (N(i) \cup \{i\}) \tag{2.39}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{2.40}$$

$$y_{pq}^{ij} \in \{0, 1\} \qquad \forall \{i, j\} \in E, \{p, q\} \in E_{ij}^1 \tag{2.41}$$

$$y_{pq}^{ij} \in \{0, 1\} \qquad \forall \{i, j\} \in \mathcal{N}^3, \{p, q\} \in E_{ij} \tag{2.42}$$

In formulation (2.30)–(2.42), $\mathcal{N}^3$ denotes the set of all pairs of non-adjacent vertices whose distance does not exceed three in graph G, $\mathcal{P}^3$ corresponds to the set of all pair of vertices that cannot be simultaneously included in a 3-club, $E_{ij} = \{\{p, q\} \in E : p \in N(i)\backslash N(j), q \in N(j)\backslash N(i)\}$ represents the set of inner edges of three-edge chains that link vertices i and j, and V_{ij} denotes the set of their end vertices which can be partitioned into subsets $F_{ij} = \{v \in V_{ij} : v \in N(i) \setminus N(j)\}$ and $B_{ij} = \{v \in V_{ij} : v \in N(j) \setminus N(i)\}$. Let $E^1_{ij} = \{\{p, q\} \in E : p \in N(i)\backslash(N(j) \cup \{j\}), q \in N(j)\backslash(N(i) \cup \{i\})\}$ and each variable y^{ij}_{pq} is associated with one edge in $E^1_{i,j}$ for any pair of adjacent vertices i, j in G.

Buchanan and Salemi [2017] suggested that a cut-like s-club formulation can be slightly modified for the r, s-MHCP presented next.

$$\max \sum_{i \in V} x_i \tag{2.43}$$

$$r(x_u + x_v - 1) \leq \sum_{i \in C} x_i \qquad \forall (u, v, C) \tag{2.44}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{2.45}$$

In formulation (2.43)–(2.45), $\forall(u, v, C)$ is again a shorthand for any pair of non-adjacent vertices u, v and all length-s u, v-separators C in graph G. Komusiewicz et al. [2019] studied the special case r-hereditary 2-club and presented an IP formulation based on the maximum 2-club problem formulation [Bourjolly et al., 2002]. Recalling that the maximum r-hereditary 2-club problem only requires additional constraints for non-adjacent pairs of vertices, consider the following formulation introduced by Komusiewicz et al. [2019].

$$\max \sum_{i \in V} x_i \tag{2.46}$$

$$\text{s.t. } r(x_u + x_v - 1) \leq \sum_{t \in N(u) \cap N(v)} x_t \qquad \forall \{u, v\} \in \binom{V}{2} \backslash E \tag{2.47}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{2.48}$$

Constraints (2.46) enforce that at least r common neighbors must be selected if a non-adjacent pair of vertices $u, v \in V$ are included in the solution.

An IP formulation for the maximum biconnected 2-club problem was proposed by Yezerska et al. [2017].

$$\max \sum_{i \in V} x_i \tag{2.49}$$

$$\text{s.t. } x_u + x_v - \sum_{t \in N(u) \cap N(v)} x_t \leq 1 \qquad \forall \{u, v\} \in \binom{V}{2} \setminus E \tag{2.50}$$

$$2(x_u + x_v - 1) \leq \sum_{t \in D} x_t \qquad \forall (u, v, D) \tag{2.51}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{2.52}$$

In formulation (2.49)–(2.51), $\forall (u, v, D)$ is a short hand for any pair of non-adjacent vertices u, v such that $d_G(u, v) = 2$ and all minimal (u, v)-vertex separators D in graph G. Constraints (2.50) ensure that the solution is a 2-club, and constraints (2.51) enforce that the solution is biconnected [Yezerska et al., 2017].

2.2.2 Preprocessing techniques and heuristics

To speed up algorithms for solving special cases of these problems, some preprocessing techniques and heuristic ideas have been considered in the literature, designed to reduce the size of the instance. The main ideas are briefly reviewed next.

For the maximum r-robust 2-club problem, Veremyev and Boginski [2012] and Komusiewicz et al. [2019] pointed out that we can ignore any vertex with degree less than r, as any vertex in an r-robust 2-club must have degree at least r. Similarly, all vertices with degree less than r can be removed for both maximum r-hereditary 2-club and maximum r-connected 2-club problems [Komusiewicz et al., 2019].

Yezerska et al. [2017] employed lower bound heuristics called VDEGREE and DROP for the maximum biconnected 2-club problem. For VDEGREE, the vertices were sorted in non-increasing order of their degree and then the heuristic identified connected components of the subgraphs induced by the *open* neighborhood of each vertex. The heuristic terminated when its size was larger than the degree of next vertex to be explored. The largest connected component along with its corresponding vertex formed a biconnected 2-club. The other heuristic technique DROP, used in this work was originally introduced by Bourjolly et al. [2000], which recursively deleted vertices with the smallest number of distance-2 neighbors until a 2-club was detected. If such 2-club was not biconnected, they constructed a biconnected 2-club from the subgraph induced by the 2-club found by DROP.

2.3 Research statement

In this dissertation, we study s-clubs with side-constraints that make them less "fragile", i.e., less susceptible to increase in the diameter if vertices (and edges) are deleted. Given a graph $G = (V, E)$, find a subset S of vertices of maximum cardinality such that:

i) $\text{diam}(G[S]) \leq s$, and

ii) S satisfies a graph property Π that reduces the fragility of $G[S]$ under vertex deletion.

In this sense, we focus on Π being r-robustness and wherever possible extend our results to r-heredity and r-connectedness.

Based on the literature review, theoretical and algorithmic results addressing both r, s-MRCP and r, s-MHCP are limited. Though Komusiewicz et al. [2019] established complexity results for the r-Robust s-Club and r-Hereditary s-Club when $s = 2$, computational complexities of both the robust and hereditary variants are still open for general positive integers $r \geq 2$ and $s \geq 3$. Almeida and Carvalho [2014] developed an IP formulation for the maximum r-robust 3-club problem by enumerating chains with one edge, two edges, and three edges, but no numerical experiments were reported in that work. For $r \geq 2$ and $s \geq 4$, there are no IP formulations and computational results for the r, s-MRCP. Furthermore, no computational experiments are available for r, s-MHCP when $r \geq 2$, and $s \geq 3$ in the literature.

These gaps motivate us to develop effective exact algorithms to solve r, s-MRCP and r, s-MHCP with an emphasis on $s = 2, 3, 4$ and low values of r on moderately large instances (around 10^4 vertices and less than 5% density). To achieve these goals, we address the following research objectives in this dissertation.

(i) Establish the computational complexity of both r-ROBUST s-CLUB and r-HEREDITARY s-CLUB on arbitrary and restricted graph classes for every pair of fixed positive integers $r \geq 2$ and $s \geq 3$.

(ii) Investigate IP formulations that permit row-generation schemes for r, s-MRCP for positive integers $r \geq 2$ and $s = 2, 3$, and 4. Extend these formulations to r, s-MHCP if possible.

(iii) Devise BC algorithms based on a delayed constraint generation scheme for both r, s-MRCP and r, s-MHCP.

(iv) Exploit preprocessing techniques including vertex and edge peeling and develop block based graph decomposition approaches for solving both r, s-MRCP and r, s-MHCP.

2.4 Organization of the dissertation

The remainder of this dissertation is organized as follows. In Chapter III, we establish complexity results for both r-ROBUST s-CLUB and r-HEREDITARY s-CLUB on arbitrary and restricted graph classes. Both problems are shown to be NP-complete, and we also prove the NP-hardness of feasibility testing when r or s is part of the input. Preprocessing techniques including vertex and edge peeling are discussed in Chapter IV. We also develop a block based decomposition algorithm in this chapter. The special case $r, 2$-MRCP is considered and we present a strengthened formulation in Chapter V. A computational study is conducted for assessing the performance of these approaches. We also extend these ideas to $r, 2$-MHCP and the maximum biconnected 2-club problem in this chapter. The focus of Chapter VI is on extending approaches for solving the $r, 2$-MRCP to larger s values. A cut-like IP formulation is presented for r, s-MRCP when $s = 2, 3$ and 4, which is the first IP formulation for the maximum r-robust 4-club problem in the literature. Preprocessing techniques and a block based graph decomposition algorithm are utilized for solving r, s-MRCP and r, s-MHCP. Numerical results are also reported to demonstrate the performance of these approaches in this chapter. Finally, we conclude our work in Chapter VII, which summarizes our contributions and also identifies some future research directions.

CHAPTER III

Computational Complexity

From the literature review in Chapter II, we have known that the computational complexities of both r-Robust s-Club and r-Hereditary s-Club are still open for every constant integers $r \geq 2$ and $s \geq 2$. In this chapter, we will not only prove results pertaining to the NP-hardness of optimization, but also the NP-hardness of feasibility testing. The computational complexity on some special graph classes is also established.

3.1 NP-hardness of optimization

Recall that Bourjolly et al. [2002] showed that s-Club is NP-complete for every constant integer $s \geq 2$ and it is NP-complete even when restricted to graphs of diameter $s+1$ [Balasundaram et al., 2005]. Theorems 1 and 2 that follow establish that r-Hereditary s-Club and r-Robust s-Club are NP-complete using reductions from s-Club. Note that the problems are trivially NP-hard when parameters r and s are not fixed in the problem definition, and are part of the input as they all include Clique as a special case when $s = r = 1$.

Theorem 1. r-Hereditary s-Club *is NP-complete for every constant integer* $s \geq 2$ *and* $r \geq 2$.

Proof. We show a polynomial-time reduction from s-Club. Given an instance $\langle G, c\rangle$ of s-Club, construct the instance $\langle G', c'\rangle$ of r-Hereditary s-Club as follows: let $G' := G * K_{r-1}$ and $c' := c + r - 1$. Suppose $S \subseteq V(G)$ is an s-club in G of size at least c. We claim that $S \cup V(K_{r-1})$ is an r-Hereditary s-Club in G' (of size at least c'). Consider a deletion set

$T \subseteq S \cup V(K_{r-1})$ such that $|T| \leq r-1$ and a pair of vertices $u, v \in S \cup V(K_{r-1}) \setminus T$. If $T = V(K_{r-1})$, then $G'[S \cup V(K_{r-1}) \setminus T] \simeq G[S]$, an s-club. Otherwise, $S \cup V(K_{r-1}) \setminus T$, which contains some vertex from K_{r-1} that dominates $G'[S \cup V(K_{r-1}) \setminus T]$, is a 2-club. Conversely, suppose $S' \subseteq V(G')$ is an r-hereditary s-club in G' of size at least c'. Then, $S' \setminus V(K_{r-1})$ of size at least c is an s-club in G' by Definition 4. Since $G'[S' \setminus V(K_{r-1})] \simeq G[S' \setminus V(K_{r-1})]$, it is an s-club in G as well. Since r is a fixed constant, verification can be completed in polynomial time by enumerating all possible deletion sets of size at most $r-1$ and computing the diameter. □

Corollary 1. *r-HEREDITARY s-CLUB is NP-complete for every constant integer $s \geq 2$ and $r \geq 2$, on graphs with diameter two, and on graphs with domination number one.*

Proof. Follows from the fact that every vertex in $V(K_{r-1}) \neq \emptyset$ dominates G'. □

Theorem 2. *r-ROBUST s-CLUB is NP-complete for every constant integer $s \geq 2$ and $r \geq 2$.*

Proof. We show a polynomial-time reduction from s-CLUB. Given an instance $\langle G, c \rangle$ of s-CLUB, construct the instance $\langle G', c' \rangle$ of r-ROBUST s-CLUB where $G' := G * K_{r-1}$ and $c' := c + r - 1$. Suppose $S \subseteq V(G)$ is an s-club in G of size at least c. We claim that $S \cup V(K_{r-1})$ is an r-robust s-club of size at least $c + r - 1$ in G' for the nontrivial case when $c \geq 2$.

For any two vertices $u, v \in S$, there exists a u, v-path of length at most s between them in $G[S]$ and there exist $r-1$ u, v-paths of length at most two through their common neighbors in $V(K_{r-1})$; these constitute r internally vertex-disjoint paths of length at most s between u and v in $G'[S \cup V(K_{r-1})]$.

For any two vertices $u, v \in V(K_{r-1})$ (if $r \geq 3$), given that $c \geq 2$ there are at least two paths of length two via their common neighbors in S, and $r-3$ u, v-paths of length two via their common neighbors in $V(K_{r-1})$. These paths, along with the edge $\{u, v\} \in E(G')$ constitute r internally vertex-disjoint paths of length at most s between u and v in $G'[S \cup V(K_{r-1})]$.

Finally, consider $u \in S$ and $v \in V(K_{r-1})$. There are $r-2$ u,v-paths of length two in G' via vertices in $V(K_{r-1}) \setminus \{v\}$. Since S is an s-club containing at least two vertices, there exists a vertex $w \in S \setminus \{u\}$ that is a common neighbor of u and v. Since $\{u,v\} \in E(G')$, there are r vertex disjoint paths between every pair of vertices in $G'[S \cup V(K_{r-1})]$.

Conversely, suppose $S' \subseteq V(G')$ is an r-robust s-club of size at least $c+r-1$ in G'. Since $|S \cap V(K_{r-1})| \leq r-1$, after deleting all vertices in $S \cap V(K_{r-1})$, a path of length at most s between any two vertices in subgraph $G'[S \setminus V(K_{r-1})]$ still exists. Hence, $S \setminus V(K_{r-1})$ is an s-club of size at least c in G, since $G'[S \setminus V(K_{r-1})] \simeq G[S \setminus V(K_{r-1})]$. Hence, r-Robust s-Club is NP-hard.

Golovach and Thilikos [2011] showed that verifying whether or not a graph of order n and size m contains r vertex-disjoint (u,v)-paths of length at most s between distinct vertices u and v can be answered in $O(2^{O(rs)} m \log n)$ time. Using their algorithm we can verify if $S' \subseteq V'$ is an r-robust s-club in G' in polynomial time for constant r and s. Hence, r-Robust s-Club belongs to class NP. □

Corollary 2. r-Robust s-Club *is NP-complete for every constant integer* $s \geq 2$ *and* $r \geq 2$*, on graphs with diameter two, and on graphs with domination number one.*

Proof. Follows from the fact that every vertex in $V(K_{r-1}) \neq \emptyset$ dominates G'. □

Chordal graphs, which contain no chordless cycles of length four or more, are a subclass of perfect graphs with interesting and desirable properties for clique detection [Rose et al., 1976]. For every non-negative integer t, a t-chordal graph contains no chordless cycles of length greater than t. So, 3-chordal graphs are precisely the classical chordal graphs. Golovach et al. [2014] proved that s-Club is NP-complete on 4-chordal graphs for every constant integer $s \geq 1$. Asahiro et al. [2010] proved NP-completeness on chordal graphs for every constant *even* integer $s \geq 2$, .

Corollary 3. *For every constant integer* $r \geq 2$*,* r-Hereditary s-Club *and* r-Robust s-Club *remain NP-complete,*

1. *on* 4*-chordal graphs for every constant integer* $s \geq 1$*, and*

2. *on chordal graphs for every constant* <u>*even*</u> *integer* $s \geq 2$*.*

Proof. It suffices to show that $H := G * K_\ell$ is t-chordal if G is t-chordal for $\ell \geq 1$. Suppose that G is t-chordal and $C \subseteq V(H)$ is a cycle of length strictly greater than $t \geq 3$. If C contains no vertices from K_ℓ, then C contains a chord between two vertices in $V(G)$ since G is t-chordal by assumption. If it contains at least one vertex from K_ℓ, then that vertex is adjacent to every other vertex in the C, and at least one such adjacent vertex in C creates a chord in C. □

Corollary 4. *For every constant integer* $r \geq 2$*,* r-Hereditary 2-Club *and* r-Robust 2-Club *problems remain NP-complete on graphs with clique cover number three.*

Proof. Trivially follows from the result of Golovach et al. [2014] who proved that 2-Club is NP-hard on graphs with clique cover number three, and our constructions. □

3.2 NP-hardness of feasibility testing

Recall from the proof of Theorem 2 that verifying if $S \subseteq V$ is an r-robust s-club can be completed in polynomial time by a parameterized algorithm when r and s are constants fixed in the problem definition. However, if one of these parameters is arbitrary, the feasibility problem is itself NP-hard. We establish these complexity results in this section.

When considering the problem of verifying if $S \subseteq V$ is an r-robust s-club, the following result of Lovász et al. [1978] that relates the size of vertex separators of bounded-length paths to the number of disjoint bounded-length paths is pertinent. For a pair of vertices u, v in G, let $\rho_s(G; u, v)$ denote the maximum number of internally vertex disjoint u, v-paths of length at most s in G, and for a non-adjacent pair of vertices u, v, let $\kappa_s(G; u, v)$ denote the minimum number of vertices in $V \setminus \{u, v\}$ whose deletion disconnects all u, v-paths of length at most s in G. Clearly, $\rho_s(G; u, v) \leq \kappa_s(G; u, v)$.

Proposition 1 (Lovász et al. [1978])**.** *Consider a graph G of order n, non-adjacent vertices u, v, and a positive integer s. Then,*

$$\kappa_s(G; u, v) \leq \left\lfloor \frac{n}{2} \right\rfloor \rho_s(G; u, v).$$

Furthermore, for $s \in \{2, 3, 4\}, \rho_s(G; u, v) = \kappa_s(G; u, v)$.

The problem of verifying if S is an r-robust 2-club for any $r \geq 2$ amounts to checking if every adjacent pair of vertices in S have at least $r - 1$ common neighbors inside S, and every non-adjacent pair have at least r common neighbors inside S. In general, verifying if S is an r-robust s-club for $r \geq 2$ and $s \geq 3$ is to verify if $\rho_s(G[S]; u, v) \geq r$ for every pair of vertices in S. By Proposition 1, for $s \in \{3, 4\}$, this amounts to verifying if $\kappa_s(G[S]; u, v) \geq r$ for every $\{u, v\} \in \binom{S}{2} \setminus E$ and $\kappa_s(G[S] - e; u, v) \geq r - 1$ for every $e = \{u, v\} \in \binom{S}{2} \cap E$. For a fixed constant r, this can be completed in polynomial time by enumerating every pair of vertices $u, v \in S$ and every deletion set $T_{uv} \subseteq S \setminus \{u, v\}$ of cardinality $r - 2$ or $r - 1$ depending on whether u and v are adjacent or not, respectively; and then verifying if $\mathrm{diam}(G[S \setminus T_{uv}]) \leq s$. If the diameter bound is not satisfied for any T_{uv}, then S is not an r-robust s-club, and otherwise, it is.

Answering whether or not $S \subseteq V$ is an r-robust 3-club or an r-robust 4-club, when r is arbitrary and specified as part of the input, does not follow as a polynomially verifiable question from any of the foregoing arguments. However, Itai et al. [1982] showed that for $s \in \{3, 4\}$, we can compute $\rho_s(G; u, v)$ in $O(|E|\sqrt{|V|})$ time, which can be used to answer the verification question for r-robust 3-clubs and r-robust 4-clubs for arbitrary r. Itai et al. [1982] also established the following hardness result that allows us to show that the verification problem for r-robust s-clubs is NP-hard for arbitrary r and constant integer $s \geq 5$.

Problem: VERTEX-DISJOINT s-PATHS (for constant integer $s \geq 2$).

Input: Graph $G = (V, E)$, a pair of vertices $a, b \in V$, and a positive integer r.

Question: Does G contain at least r vertex disjoint a, b-paths of length at most s?

Proposition 2 (Itai et al. [1982]). VERTEX-DISJOINT s-PATHS *is NP-complete for every constant integer* $s \geq 5$.

Now consider the following feasibility testing problem for every constant integer $s \geq 2$, specified in the problem, while r can be arbitrary, specified in the input.

Problem: IS-ROBUST s-CLUB (for constant positive integer s).

Input: Graph $G = (V, E), S \subseteq V$, and a positive integer r.

Question: Is S an r-robust s-club in G?

Theorem 3. IS-ROBUST s-CLUB *is NP-hard for every constant integer* $s \geq 5$.

Proof. We show a polynomial-time reduction from an instance $\langle G = (V, E); a, b, r\rangle$ of VERTEX-DISJOINT s-PATHS to an instance $\langle G' = (V', E'), S', r'\rangle$ of IS-ROBUST s-CLUB next. The reduction is based on a similar construction employed by Buchanan and Validi [2018].

Case (i) Suppose s is odd. We define the following sets.

$$\begin{aligned}
T &\coloneqq \{t_i : i \in [r]\} \\
A &\coloneqq \left\{a_i^j : i \in [r], j \in \left[\frac{s-1}{2}\right]\right\} \\
B &\coloneqq \left\{b_i^j : i \in [r], j \in \left[\frac{s-1}{2}\right]\right\} \\
W &\coloneqq \left\{v_i^j : i \in [r], j \in \left[\frac{s-3}{2}\right], v \in N_G^{s-1}[a] \cup N_G^{s-1}[b] \setminus \{a, b\}\right\} \\
V' &\coloneqq V \cup T \cup A \cup B \cup W
\end{aligned}$$

In the definition the set W, recall from Chapter I that $N_G^{s-1}[u]$ denotes the *closed*

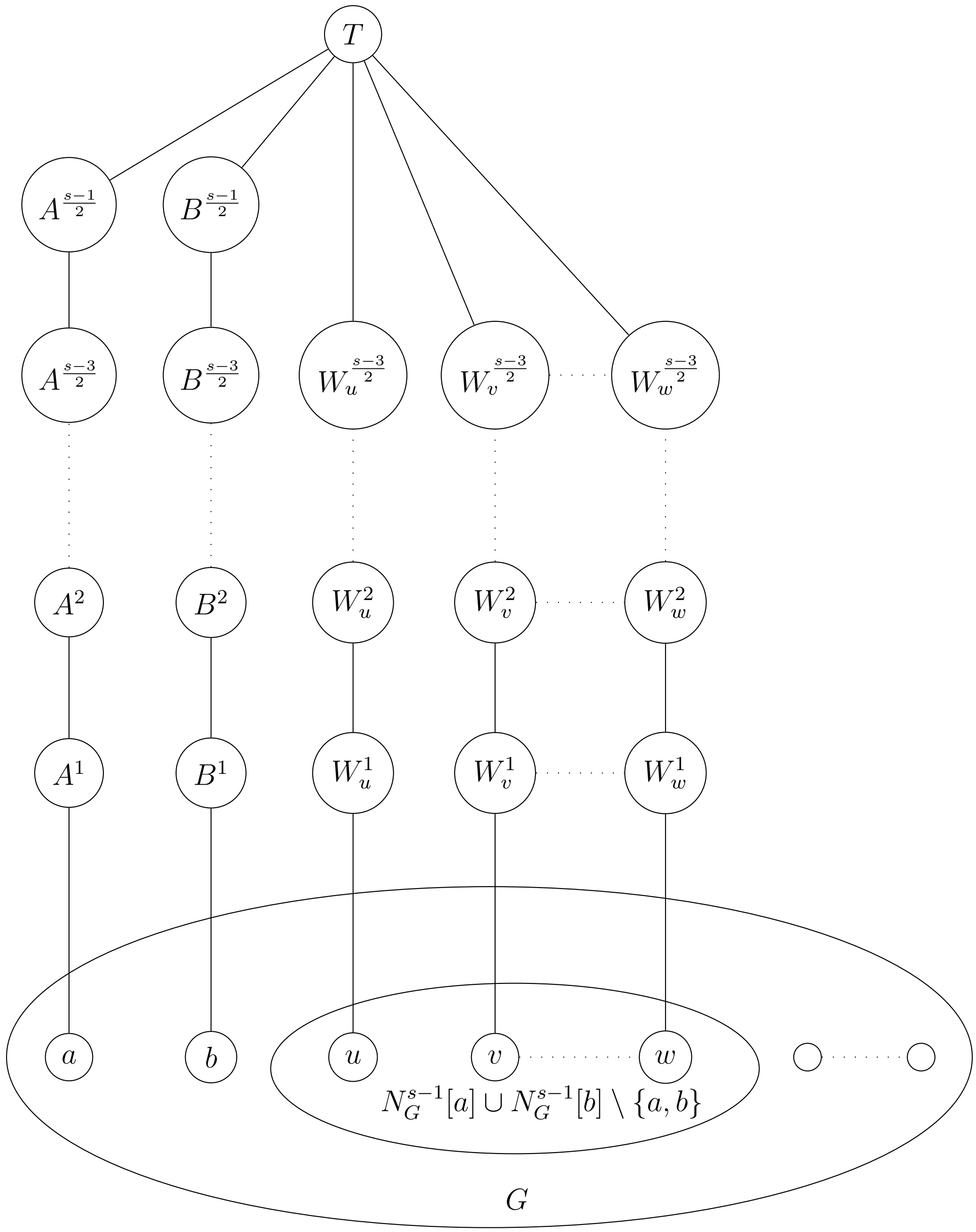

Figure 3.1: An illustration of the reduction for odd $s \geq 5$

distance-$(s-1)$ neighborhood of vertex $u \in V$. The edge set E' is constructed as follows. Figure 3.1 illustrates this reduction.

First, the vertices in $V \subset V'$ are joined by edges in exactly the same way as they are in G so that $G'[V] \simeq G$. Then we make the vertices in T pairwise adjacent, so that $G'[T]$ is complete. Similarly, the following sets also induce complete graphs in G':

$$
\begin{aligned}
A^j :=& \{a_i^j \in A : i \in [r]\} && \forall j \in \left[\frac{s-1}{2}\right] \\
B^j :=& \{b_i^j \in B : i \in [r]\} && \forall j \in \left[\frac{s-1}{2}\right] \\
W_v^j :=& \{v_i^j \in W : i \in [r]\} \qquad \forall j \in \left[\frac{s-3}{2}\right], v \in N_G^{s-1}[a] \cup N_G^{s-1}[b] \setminus \{a, b\}
\end{aligned}
$$

Then, a is made adjacent to every vertex in A^1, b is made adjacent to every vertex in B^1, and each $v \in N_G^{s-1}[a] \cup N_G^{s-1}[b] \setminus \{a, b\}$ is made adjacent to every vertex in W_v^1. Then, for every $j \in \left[\frac{s-3}{2}\right]$, every vertex in A^j is joined by an edge to every vertex in A^{j+1}, every vertex in B^j is joined by an edge to every vertex in B^{j+1}. For every $j \in \left[\frac{s-5}{2}\right]$ and for each $v \in N_G^{s-1}[a] \cup N_G^{s-1}[b] \setminus \{a, b\}$, every vertex in W_v^j is joined by an edge to every vertex in W_v^{j+1}. Finally, every vertex of T is made adjacent to every vertex in:

$$
A^{\frac{s-1}{2}} \cup B^{\frac{s-1}{2}} \cup \left(\bigcup_{v \in N_G^{s-1}[a] \cup N_G^{s-1}[b] \setminus \{a,b\}} W_v^{\frac{s-3}{2}} \right).
$$

Now, we let $r' := r$ and $S' := N_G^{s-1}[a] \cup N_G^{s-1}[b] \cup (V' \setminus V)$. This completes the construction of the instance G'; note that this can be completed in polynomial-time.

It is easy to verify that there are at least r vertex-disjoint paths of length at most s between every pair of vertices of $G'[S']$ except the pair $\{a, b\}$. Furthermore, a, b-paths of length at most s can only include vertices in V. Hence, $\langle G = (V, E); a, b, r\rangle$ is a yes-instance for Vertex-Disjoint s-Paths if and only if $\langle G', S', r'\rangle$ is a yes-instance for Is-Robust s-Club.

Case (ii) Suppose s is even. The main idea behind the reduction is very similar to the previous case. Figure 3.2 illustrates the construction for even $s \geq 5$. We define the following sets.

$$\begin{aligned} T^x &:= \{t_i^x : i \in [r]\} && \text{for } x \in \{a, b\} \\ W &:= \left\{v_i^j : i \in [r], j \in \left[\frac{s}{2} - 1\right], v \in N_G^{s-1}[a] \cup N_G^{s-1}[b]\right\} \\ V' &:= V \cup T^a \cup T^b \cup W \end{aligned}$$

As before we add edges so that $G'[V] \simeq G$, $G'[T^a \cup T^b]$ is complete, and for every $j \in \left[\frac{s}{2} - 1\right]$ and $v \in N_G^{s-1}[a] \cup N_G^{s-1}[b]$, $G[W_v^j]$ is complete, where $W_v^j := \{v_i^j \in W : i \in [r]\}$. Then, each $v \in N_G^{s-1}[a] \cup N_G^{s-1}[b]$ is made adjacent to every vertex in W_v^1, and every vertex in W_v^j is joined by an edge to every vertex in W_v^{j+1} for every $j \in \left[\frac{s}{2} - 2\right]$. Finally, for $x \in \{a, b\}$, every vertex of T^x is made adjacent to every vertex in:

$$\bigcup_{v \in N_G^{s-1}[a] \cup N_G^{s-1}[b] \setminus \{x\}} W_v^{\frac{s}{2}-1}.$$

Now, we let $r' := r$ and $S' := N_G^{s-1}[a] \cup N_G^{s-1}[b] \cup (V' \setminus V)$. This completes the polynomial-time construction of the instance G'.

As in the previous case, we can verify that there are at least r vertex-disjoint paths of length at most s between every pair of vertices of $G'[S']$ except the pair $\{a, b\}$. Furthermore, a, b-paths of length at most s can only include vertices in V. Hence, $\langle G = (V, E); a, b, r\rangle$ is a yes-instance for Vertex-Disjoint s-Paths if and only if $\langle G', S', r'\rangle$ is a yes-instance for Is-Robust s-Club.

□

We now consider the counterpart of the foregoing feasibility testing problem, Is-Robust

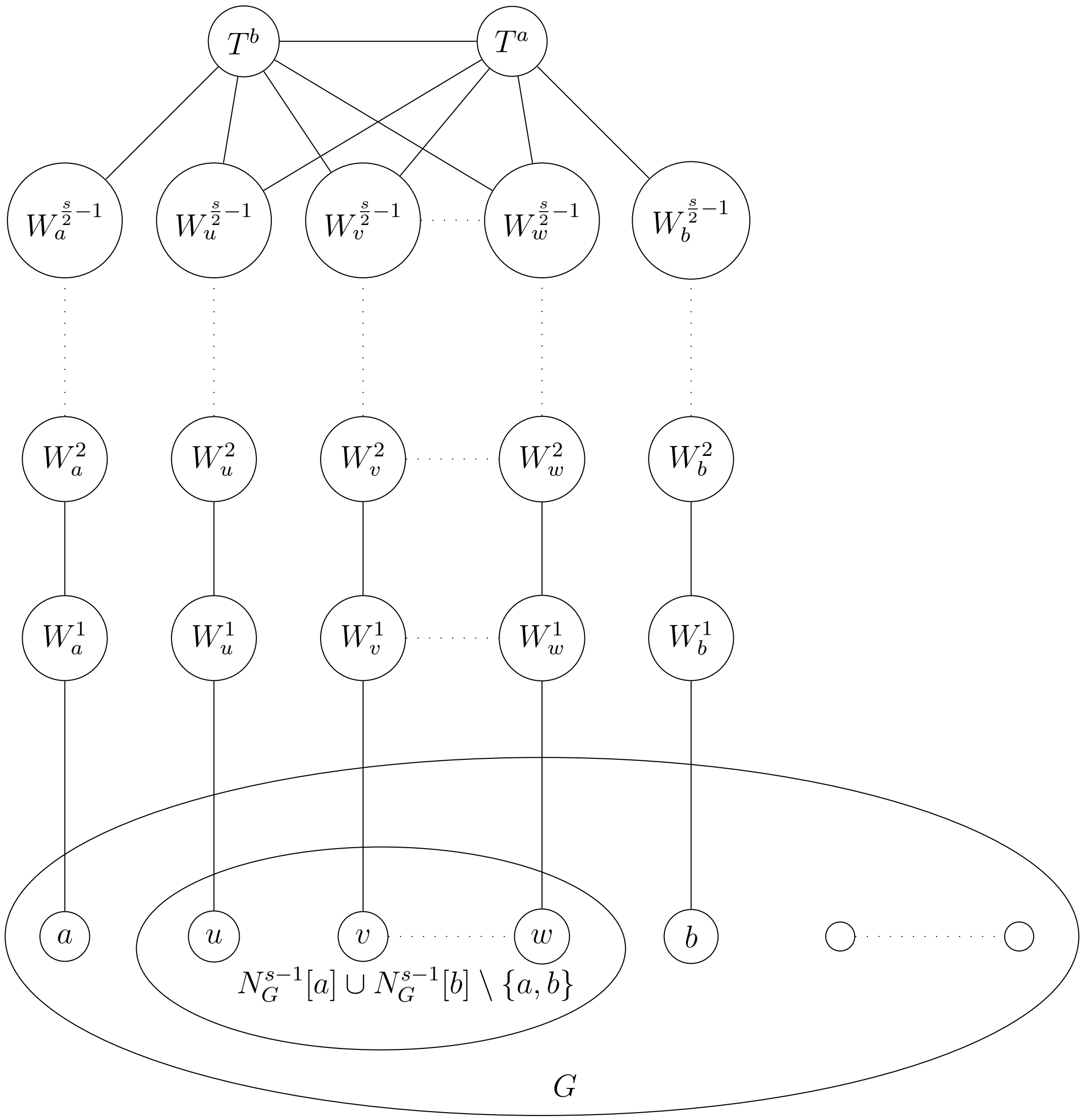

Figure 3.2: An illustration of the reduction for even $s \geq 5$

s-Club. In this version that is defined next, we treat the robustness parameter r as a constant in the problem definition and s is arbitrary, specified in the input.

Problem: Is-r-Robust Club (for constant positive integer r).

Input: Graph $G = (V, E), S \subseteq V$, and a positive integer s.

Question: Is S an r-robust s-club in G?

We prove a quick result to serve as our source problem to show that Is-r-Robust Club is NP-hard. Consider the following problem.

Problem: r-Robust Vertex-Disjoint Paths (for constant positive integer r)

Input: Graph $G = (V, E)$, a pair of vertices $a, b \in V$, and a positive integer s.

Question: Does G contain r vertex-disjoint a, b-paths of length at most s?

The special case when $r = 2$ is known to be NP-complete based on the result [Li et al., 1990] and we extend this result for every constant integer $r \geq 3$ in Lemma 2.

Lemma 2. r-Robust Vertex-Disjoint Paths *is NP-hard for every constant integer* $r \geq 3$.

Proof. Given a non-trivial instance $\langle G = (V, E); a, b, s \rangle$ of 2-Robust Vertex-Disjoint Paths with $s \geq 2$, we construct the instance $\langle G' = (V', E'); a, b, s \rangle$ of r-Robust Vertex-Disjoint Paths as follows. Define the sets,

$$P \coloneqq \{v_i^j : i \in [s-1], j \in [r-2]\},$$
$$V' \coloneqq V \cup P.$$

Join the vertices in $V \subset V'$ by edges in exactly the same way as they are in G so that $G'[V] \simeq G$. Then, for every $j \in [r-2]$ make each $P^j \coloneqq (v_1^j, \ldots, v_{s-1}^j)$ a path. Finally, join a to v_1^j and v_{s-1}^j to b for every $j \in [r-2]$ so that these are $r-2$ vertex disjoint a, b-paths of length at most s in G'. It is easy to verify that G contains two vertex-disjoint a, b-paths of length at most s if and only if G' contains r vertex-disjoint $a, b-$paths of length at most s. □

Theorem 4. Is-r-Robust Club *is NP-hard for every constant integer* $r \geq 2$.

Proof. The same reduction as Theorem 3, using r-Robust Vertex-Disjoint Paths as the source problem instead, establishes this claim. □

CHAPTER IV

Preprocessing and Graph Decomposition

In Chapter III, we have shown that r-ROBUST s-CLUB and r-HEREDITARY s-CLUB are NP-hard and thus are challenging to solve. Therefore, it is crucial to develop effective exact algorithms for solving these problems in practice, even though they may be of worst-case computational complexity. First, we introduce heuristics for finding an r-robust 2-club. Then, we develop preprocessing techniques including vertex and edge peeling to reduce the size of graph for the r,s-MRCP. Also, the subgraph induced by an r-robust s-club is contained in some block of the graph when $r \geq 2$. Based on this observation, we present an algorithm for the r,s-MRCP that uses block decomposition. Interestingly, these ideas can also be extended to the maximum r-hereditary s-club and r-connected s-club problems.

4.1 Construction heuristic

Every pair of adjacent vertices along with their common neighbors form a 2-robust 2-club. This observation motivates a heuristic for finding a 2-robust 2-club that we refer to as H2R2C. We check every pair of adjacent vertices and their common neighbors, and the one with the largest cardinality is retained as the heuristic solution. The pseudocode of H2R2C is presented in Algorithm 1. Note that the smallest non-trivial 2-robust 2-club is a triangle, and thus we return a singleton set $\{1\}$ if the heuristic solution is not larger than two vertices.

Analogously, for $r \geq 3$, finding an r-robust 2-club will require us to first identify a maximal clique C of size at least $r+1$ if it exists, which can be implemented in $O(|V|+|E|)$ [Matula and Beck, 1983]. The basic idea is to sort vertices in a degeneracy ordering, and then remove

Algorithm 1 Heuristic for 2-robust 2-club

procedure H2R2C(G)
Input: Graph $G = (V, E)$
Output: A heuristic solution of 2-robust 2-club S
Initialize: $S \leftarrow \emptyset$
    for each edge $uv \in E$ **do**
        if $|N(u) \cap N(v)| + 2 > |S|$ **then**
            $S \leftarrow N[u] \cap N[v]$
        end if
    end for
    if $|S| \leq 2$ **then**
        return $\{1\}$
    else
        return S
    end if
end procedure

Algorithm 2 Heuristic for r-robust 2-club

procedure HRR2C(G, r)
Input: Graph $G = (V, E)$ and a positive integer $r \geq 3$
Output: An r-robust 2-club S
Initialize: $S \leftarrow \emptyset, P \leftarrow \emptyset$
    Find a maximal clique C in G
    if $|C| \leq r$ **then**
        return $\{1\}$
    else
        for each edge $uv \in E(G[C])$ **do**
            $T \leftarrow (N(u) \cap N(v)) \setminus C$
            $W \leftarrow \arg\max_{F \subseteq T} \{|F| : |N(i) \cap N(j) \cap C| \geq r, \forall i, j \in F\}$
            $P \leftarrow \arg\max\{|P|, |W|\}$
        end for
        $S \leftarrow P \cup C$
        return S
    end if
end procedure

a vertex with the minimum degree. Repeat this process until the remaining vertex set is a clique. In this dissertation, we make use of the algorithm of finding a maximal clique implemented by Buchanan and Salemi [2017]. Then, we obtain a subset W such that every pair of vertices in W share at least r common vertices in C. Hence, there exist at least r vertex-disjoint paths of length no more than two among all pairs of vertices in the subgraph $G[C \cup W]$; i.e., the subset $C \cup W$ forms an r-robust 2-club. The heuristic for finding an r-robust 2-club ($r \geq 3$) referred to as HRR2C is presented in Algorithm 2.

Since every r-robust 2-club is also an r-robust s-club when $s \geq 2$, we can use the same heuristic to obtain an r-robust s-club for $s \geq 3$. Recall from Lemma 1 and Menger's Theorem [Menger, 1927] that every r-robust s-club is also r-hereditary and r-connected when $s \geq 2$ and $r \geq 2$. Therefore, the aforementioned Algorithms 1 and 2 can also serve as heuristics for finding r-hereditary and r-connected s-clubs.

4.2 Vertex peeling

In this section, we introduce a preprocessing technique called vertex peeling to remove vertices which cannot be selected in an optimal solution based on a lower bound on the optimal solution size. First, we discuss the vertex peeling technique for the r, s-MRCP, and then extend this idea to the r, s-MHCP.

4.2.1 Vertex peeling for the r, s-MRCP

The degree of every vertex in an r-robust s-club must be greater than or equal to r, so we can first employ r-Core peeling presented in Appendix B to remove some vertices, which can be accomplished using an $O(|V| + |E|)$ algorithm [Batagelj and Zaversnik, 2003].

Based on a lower bound ℓ on the optimal value of the r, s-MRCP, we are able to remove vertices that cannot be included in a solution larger than ℓ. Specifically, when a vertex has fewer than ℓ distance-s neighbors, it cannot be selected in a solution whose size is greater than ℓ by the diameter requirement. Hence, we can remove such a vertex without affecting the optimal solutions. This preprocessing technique is essentially the core peeling idea used for the maximum clique problem [Abello et al., 1999, Verma et al., 2015]. For the maximum

r-robust s-club problem, we can strengthen this further. We may still be able to remove vertex v such that $|N^s(v)| \geq \ell$. If there exist at most $r-1$ vertex-disjoint paths of length no greater than s between v and some distance-s neighbor u, then u and v cannot be included simultaneously in an r-robust s-club. Hence, a vertex $v \in V$ can be removed if $|T_v| < \ell$ where $T_v := \{u \in N^s(v) : \rho_s(G; v, u) \geq r\}$.

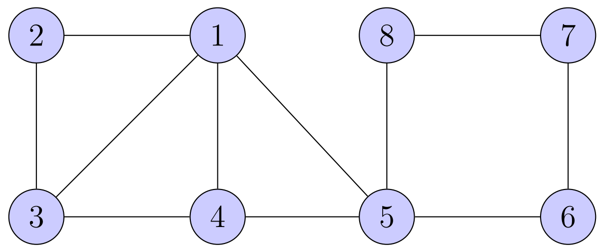

Figure 4.1: A graph without vertex peeling

Consider the graph in Figure 4.1 for instance. Suppose $r = s = 2$, and we have a 2-robust 2-club of size $\ell = 3$ (e.g., $\{1, 2, 3\}$). For vertex 1, its associated $T_1 = \{2, 3, 4, 5\}$ and thus vertex 1 is not deleted by vertex peeling. Similarly, we can check that for each vertex v in $\{2, 3, 4, 5\}$, the $|T_v| \geq 3$ and therefore they cannot be removed by vertex peeling. However, for vertices $\{6, 7, 8\}$,size of sets T_6, T_7, and T_8 is smaller than three. For instance, though the distance-2 neighbors of the vertex 7, $N^2(7) = \{5, 6, 8\}$, the set $T_7 = \{5\}$. Therefore, vertices $\{6, 7, 8\}$ and their incident edges are removed by vertex peeling shown in Figure 4.2.

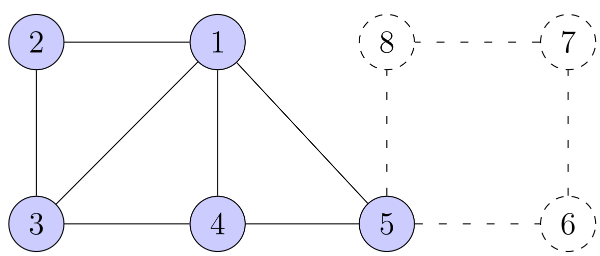

Figure 4.2: Vertices $\{6, 7, 8\}$ and their incident edges are deleted after vertex peeling.

We can recursively implement this preprocessing technique, since each vertex v that is removed may affect the size of the distance-s neighborhood of another vertex. The preprocessing technique is described in Algorithm 3 in detail. When $s = 2$, step 9 in

Algorithm 3 requires $O(|E|+|V|)$ time. When $s \in \{3,4\}$, the worst case complexity to compute $\rho_s(G;u,v)$ is $O(|E|\sqrt{|V|})$ [Lovász et al., 1978]. Hence, the time complexity of Algorithm 3 is $O(|E||V|^2)$ when $s=2$, but it is $O(|E||V|^{2.5})$ when $s \in \{3,4\}$.

Algorithm 3 Vertex Peeling for the r,s-MRCP

```
procedure VERTEX PEELING-MRC(G, r, s, ℓ)
Input: Graph G = (V, E), positive integers r, s ≥ 2, and a solution of size ℓ
Output: Modified graph G
    flag ← true
    while flag do
        r-CORE(G)
        flag ← false
        for each vertex v ∈ V do
            T ← {u ∈ N^s_G(v) : ρ_s(G; v, u) ≥ r}
            if |T| < ℓ then
                G ← G − v
                flag ← true
            end if
        end for
    end while
    return G
end procedure
```

4.2.2 Vertex peeling for the r,s-MHCP

In this section, we will extend vertex peeling to r,s-MHCP. Again assume a lower bound ℓ for the r,s-MHCP. Since the subgraph induced by an r-hereditary s-club requires that its diameter is at most s, any vertex $v \in V$ with fewer than ℓ distance-s neighbors cannot be chosen in a solution whose size is more than ℓ and thus can be removed. By Proposition 1, verifying an r-hereditary s-club is equivalent to checking if $\rho_s(G[S];u,v) \geq r$ for every pair of *non-adjacent* vertices u and v in subgraph $G[S]$ when $s=2,3$ and 4. By an argument similar to the one presented in Section 4.2.1, a vertex v might be removed although $|N^s(v)| \geq \ell$ because we may have $\rho_s(G;u,v) \leq r-1$ between vertex v and enough of its *non-adjacent* distance-s neighbors u. Notice that there are no additional requirements placed on a pair of *adjacent vertices* in r-hereditary s-clubs. Therefore, a vertex $v \in V$ can be deleted if $|T| + |N(v)| < \ell$ where $T := \{u \in N^s(v) \setminus N(v) : \rho_s(G; u, v) \geq r\}$. Also, we can recursively apply this preprocessing technique since the graph structure changes after removing some vertices. The pseudocode of vertex peeling for the r, s-MHCP is presented in Algorithm 4. The time complexity of Algorithm 4 is the same as that of Algorithm 3.

Algorithm 4 Vertex Peeling for the r, s-MHCP

```
procedure VERTEX PEELING-MHC(G, r, s, ℓ)
Input: Graph G = (V, E), positive integers r ≥ 2, s ∈ {2, 3, 4}, and a solution of size ℓ
Output: A modified graph G
    flag ← true
    while flag do
        r-CORE(G)
        flag ← false
        for each vertex v ∈ V do
            T ← {u ∈ N_G^s(v) \ N_G(v) : ρ_s(G; u, v) ≥ r}
            if |T| + |N_G(v)| < ℓ then
                G ← G − v
                flag ← true
            end if
        end for
    end while
    return G
end procedure
```

4.3 Edge peeling

In this section, we introduce a preprocessing technique for the r, s-MRCP to delete edges that cannot belong to an r-robust s-club based on a similar idea introduced by Verma et al. [2015]. First, we present the following lemma to support the methodology.

Lemma 3. *Consider a graph $G = (V, E)$ and an r-robust s-club $S \subseteq V$. For $r \geq 2$ and an edge $e = uv$ such that $\rho_s(G; u, v) \leq r - 1$, S is also an r-robust s-club in subgraph $G - e$.*

Proof. Since $\rho_s(G; u, v) \leq r - 1$, at most one of the vertices u and v can be present in S by

Definition 3. This implies that the edge $e = uv$ is not on any path in $G[S]$. □

Algorithm 5 Edge Peeling for the r, s-MRCP

```
procedure EDGE PEELING-MRC(G, r, s)
Input: Graph G = (V, E) and positive integers r, s
Output: Modified graph G = (V, E)
    while e = uv ∈ E such that ρ_s(G; u, v) ≤ r − 1 exists do
        G ← G − e
    end while
    return G
end procedure
```

In view of Lemma 3, feasible solutions remain unchanged after deleting any edge $e = uv$ such that $\rho_s(G; u, v) \leq r - 1$. Hence, we can recursively delete all such edges without affecting feasible solutions. We refer to this as edge peeling and its pseudocode is presented in Algorithm 5. This edge peeling technique is capable of sparsifying the original graph, which could help reduce the difficulty of solving the r, s-MRCP on some test-beds. Now we take the graph in Figure 4.3 as an example to illustrate how edge peeling works. Consider the case $r = s = 2$. For the edge $e = \{1, 2\}$, we have $\rho_2(G; 1, 2) = 1 \leq 1$ and thus the edge $e = \{1, 2\}$ can be deleted. Similarly, we can remove edges $\{2, 3\}, \{3, 5\}$, $\{3, 7\}, \{4, 5\}, \{4, 7\}$, and the resulting graph is shown in Figure 4.4.

When $s = 2$, $\rho_s(G; u, v) = |N(u) \cap N(v)| + 1$ and thus Algorithm 5 essentially finds an $(r - 1)$-community or r-triangle core studied by many researchers such as Cohen [2008, 2009], Wang and Cheng [2012], Zhang and Parthasarathy [2012], Verma et al. [2015], Rossi [2014], and it can be implemented to run in $O(|E|^{1.5})$ [Rossi, 2014]. When $s \in \{3, 4\}$, the worst case complexity to compute $\rho_s(G; u, v)$ is $O(|E|\sqrt{|V|})$ [Lovász et al., 1978]. Therefore, the time complexity of Algorithm 5 is $O(|E|^2\sqrt{|V|})$ when $s \in \{3, 4\}$.

Note that unlike r-robust s-clubs, there are no additional requirements placed on a pair of adjacent vertices in r-hereditary s-clubs. For this reason, we cannot utilize a similar edge peeling technique for the r, s-MHCP.

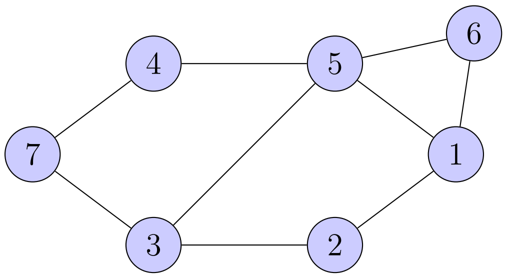

Figure 4.3: A graph G without edge peeling.

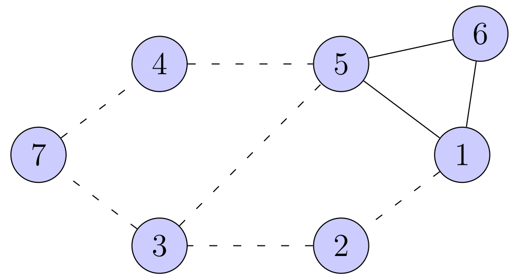

Figure 4.4: Edges $\{1,2\},\{2,3\},\{3,5\}$, $\{3,7\},\{4,5\},\{4,7\}$ are deleted by edge peeling technique.

4.4 Block decomposition algorithm

In this section, we will discuss a block decomposition approach to solve the r,s-MRCP. The main idea is to decompose the original graph G into many smaller blocks so we can restrict our solvers to look for our solution inside one block at a time. We can recursively apply such a decomposition approach along with preprocessing techniques including vertex and edge peeling introduced in Sections 4.2 and 4.3. Block decomposition idea can also be extended to the maximum r-hereditary and r-connected s-club problems.

4.4.1 Block decomposition for the r,s-MRCP

Recall that a block is a maximal biconnected subgraph and thus every r-robust s-club whenever $r \geq 2$ must be contained in some block. This observation yields the following lemma that we state without proof (trivial).

Lemma 4. *Given a graph $G=(V,E), r \geq 2$ and an r-robust (r-hereditary or r-connected) s-club $S \subseteq V$, then there exists a block B in G containing S.*

Finding all blocks in G takes $O(|V|+|E|)$ time [Hopcroft and Tarjan, 1973]. The pseudocode for finding all blocks in G is presented in Algorithm 14 in Appendix A. Note that we can apply vertex and edge peeling preprocessing techniques on each block. As a result, the preprocessed "blocks" may admit further decomposition into smaller blocks after applying the preprocessing techniques. For example, the graph G in Figure 4.5 decomposes into two blocks B_1 and B_2. Consider the case $s = 2$ and $r = 3$. If we apply edge peeling in block B_2, all edges in block B_2 would be removed. As a result, B_2 decomposes further into three isolated vertices $\{1\}$, $\{5\}$, and $\{6\}$, i.e., three blocks (trivial graphs)[1]. Based on this idea, we recursively apply Algorithm 14 along with aforementioned preprocessing techniques. This motivates us to develop a Recursive Block algorithm presented in Algorithm 6.

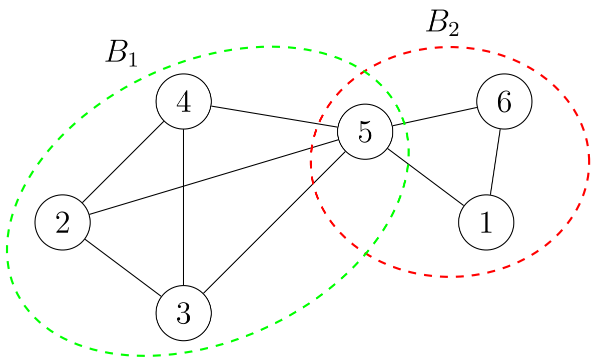

Figure 4.5: A graph G decomposes into two blocks B_1 and B_2

Algorithm 6 Recursive Block Decomposition for the r, s-MRCP

procedure Recursive Block($G, r, s, \ell, \mathcal{C}$)
Input: Graph G, positive integers $r, s \geq 2$, a solution of size ℓ, and an empty set $\mathcal{C}$
Output: The collection of candidate blocks $\mathcal{C}$
    $G \leftarrow$ Edge Peeling(G, r, s)
    $G \leftarrow$ Vertex Peeling(G, r, s, ℓ)
    $\mathcal{B} \leftarrow$ Find Block(G)
    if $|\mathcal{B}| = 1$ **then**
        $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{B}$
    else
        for each block $K \in \mathcal{B}$ **do**
            Recursive Block($K, r, s, \ell, \mathcal{C}$)
        end for
    end if
    return $\mathcal{C}$
end procedure

[1] Assuming an isolated vertex is biconnected.

Algorithm 7 Block-by-Block Decomposition Algorithm for the r, s-MRCP

procedure B/B-MRC(G, r, s, ℓ)
Input: Graph G, positive integers $r, s \geq 2$, and a solution S of size ℓ
Output: A maximum r-robust s-club S
Initialize: $\mathcal{C} \leftarrow \emptyset, i \leftarrow 1$
    $\mathcal{B} \leftarrow$ RECURSIVE BLOCK($G, r, s, \ell, \mathcal{C}$)
    Let $\mathcal{B} = \langle B_1, \ldots, B_k \rangle$ in non-increasing order of sizes
    while $|B_i| > |S|$ **do**
        $\hat{S} \leftarrow$ A maximum r-robust s-club in the block B_i
        $S \leftarrow \arg\max\{|S|, |\hat{S}|\}$
        $i \leftarrow i + 1$
    end while
    return S
end procedure

After identifying all candidate blocks (those that could contain an optimal solution) using Algorithm 6, we face a few implementation questions. In which block should we first solve the maximum r-robust s-club problem? Do we need to solve the problem in all the blocks? We choose the "greedy" strategy of solving the problem on the largest block first, and update the current best solution after each block is solved. When the block size is no greater than the current best objective value, the algorithm is terminated. This approach is referred to as block-by-block (B/B) decomposition algorithm and presented in Algorithm 7.

4.4.2 Block-by-block decomposition for the r, s-MHCP

Based on Lemma 4, every r-hereditary s-club S ($r \geq 2$) must be contained in a block of graph G. But we are unable to directly utilize Algorithm 6 to find all candidate blocks for the r, s-MHCP since their preprocessing techniques are different from those used for the r, s-MRCP. We can modify Algorithm 6 in order to find all candidate blocks for the r, s-MHCP, which is presented in Algorithm 8.

After recursively obtaining all candidate blocks by Algorithm 8, we can also develop an analogous B/B decomposition approach for the r, s-MHCP, which is presented in Algorithm 9.

Algorithm 8 Recursive Block Decomposition for the r,s-MHCP

procedure Recursive Block-MHC($G, r, s, \ell, \mathcal{C}$)
Input: Graph G, positive integers $r, s \geq 2$, a solution of size ℓ, and an empty set $\mathcal{C}$
Output: The collection of candidate blocks $\mathcal{C}$
    $G \leftarrow$ Vertex Peeling-MHC(G, r, s, ℓ)
    $\mathcal{B} \leftarrow$ Find Block(G)
    if $|\mathcal{B}| = 1$ **then**
        $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{B}$
    else
        for each block $K \in \mathcal{B}$ **do**
            Recursive Block-MHC($K, r, s, \ell, \mathcal{C}$)
        end for
    end if
    return $\mathcal{C}$
end procedure

Algorithm 9 Block-by-Block Decomposition Algorithm for the r,s-MHCP

procedure B/B-MHC(G, r, s, ℓ)
Input: Graph G, positive integers $r, s \geq 2$, and a solution S of size ℓ
Output: A maximum r-hereditary s club S
Initialize: $\mathcal{C} \leftarrow \emptyset, i \leftarrow 1$
    $\mathcal{B} \leftarrow$ Recursive Block-MHC($G, r, s, \ell, \mathcal{C}$)
    Let $\mathcal{B} = \langle B_1, \ldots, B_k \rangle$ in non-increasing order of sizes
    while $|B_i| > |S|$ **do**
        $\hat{S} \leftarrow$ A maximum r-hereditary s club in the block B_i
        $S \leftarrow \arg\max\{|S|, |\hat{S}|\}$
        $i \leftarrow i + 1$
    end while
    return S
end procedure

4.4.3 Block-by-block decomposition for the r,s-MCCP

Similar to the r,s-MHCP, the block-by-block decomposition algorithm for the r,s-MRCP can also be extended to the r,s-MCCP when $r \geq 2$. We can solve the r,s-MCCP on each block B_i of graph G and the largest r-connected s-club among all "block-optimal" solutions will become our global optimal solution. As before, we can sort blocks in non-increasing order of

Algorithm 10 Block-by-Block Decomposition Algorithm for the r, s-MCCP

procedure B/B-MCC(G, r, s)
Input: Graph G and postive integers $r, s \geq 2$
Output: A maximum r-connected s-club S
Initialize: $S \leftarrow \emptyset, i \leftarrow 1$
    $G \leftarrow r$-CORE(G)
    $\mathcal{B} \leftarrow$ FIND BLOCK(G)
    Let $\mathcal{B} = \langle B_1, \ldots, B_k \rangle$ in the non-increasing order of sizes
    while $|B_i| > |S|$ **do**
        $\hat{S} \leftarrow$ A maximum r-connected 2-club in the block B_i
        $S \leftarrow \arg\max\{|S|, |\hat{S}|\}$
        $i \leftarrow i + 1$
    end while
    return S
end procedure

sizes and use the "greedy" strategy of solving the problem on the largest block first. The algorithm is terminated if the current best solution is equal to or greater than the block size B_i. Notice that although we do not have vertex and edge peeling in Algorithms 3 and 5, we can still utilize the classical r-Core peeling to remove vertices for the r, s-MCCP. It helps the graph G decompose into smaller blocks. The detailed block-by-block decomposition approach for solving the r, s-MCCP is presented in Algorithm 10.

In this chapter, we introduce vertex and edge peeling techniques to reduce the size of the graph for the r, s-MRCP. A block-by-block decomposition approach is also proposed to speed up algorithms for solving such problems. We extend these approaches to the r, s-MHCP and r, s-MCCP. In chapters 5 and 6, we will design computational experiments to assess the performance of these approaches, and also discuss IP related approaches for the r, s-MHCP and r, s-MCCP.

CHAPTER V

Second Order 2-Clubs

In this chapter, we consider a special case, the maximum r-robust 2-club problem ($r, 2$-MRCP). First, a strengthened IP formulation is presented for the $r, 2$-MRCP. We devise a branch-and-cut algorithm based on a delayed constraint generation scheme. An extended formulation is also developed for the $r, 2$-MRCP. To assess the performance of these approaches, as well as the preprocessing techniques and decomposition discussed in Chapter IV, we conduct a computational study on large-scale real-life instances from DIMACS [Bader et al., 2013] and Stanford Network Analysis Platform [Leskovec and Krevl, 2014] as well as randomly generated graphs.

5.1 Strengthened IP formulation

Recalling from Chapter II, an IP formulation for the maximum 2-robust s-club problem proposed by Veremyev and Boginski [2012] is referred to as **OF**.

$$\textbf{(OF)} \quad \max \sum_{i \in V} x_i \tag{5.1}$$

$$\text{s.t. } r(x_i + x_j - 1) \leq \mathbb{1}_E(i,j) + \sum_{k \in N(i) \cap N(j)} x_k \qquad \forall \{i,j\} \in \binom{V}{2} \tag{5.2}$$

$$x_i \in \{0,1\} \qquad \forall i \in V \tag{5.3}$$

Consider an r-robust 2-club $S \subseteq V$. Constraints (5.2) ensure that there exist r vertex-disjoint paths of length at most two between any pair of vertices $i, j \in S$. Suppose an adjacent pair of vertices i, j are selected in S, it implies there must be at least $r-1$ vertices of $N(i) \cap N(j)$ selected in S, denoted by $C \coloneqq N(i) \cap N(j) \cap S$, and thus $|C| \geq r-1$. Now for each $k \in C$, it follows $\{i, k\} \in E$ and $\{j, k\} \in E$. Hence, we have $|N(i) \cap N(k) \cap S| \geq r-1$ and $|N(j) \cap N(k) \cap S| \geq r-1$. A similar argument also applies to any non-adjacent pair of vertices i, j. This observation allows us to strengthen constraints (5.2) as follows:

$$r(x_i + x_j - 1) \leq \mathbb{1}_E(i,j) + \sum_{k \in \Delta_{ij}^r} x_k \qquad \forall \{i,j\} \in \binom{V}{2},$$

where $\Delta_{ij}^r \coloneqq \{k \in N(i) \cap N(j) : |N(i) \cap N(k)| \geq r-1, |N(j) \cap N(k)| \geq r-1\}$.

If $\mathbb{1}_E(i,j) + |\Delta_{ij}^r| \leq r-1$, it implies that $\mathbb{1}_E(i,j) + \sum_{k \in \Delta_{ij}^r} x_k \leq r-1$, and thus at most one of $i, j \in S$ can be selected. Therefore, this constraint can be further strengthened as constraints (5.5) and (5.6).

Putting them together, we obtain a strengthened formulation, referred to as **SF**, which is presented next.

$$\textbf{(SF)} \quad \max \sum_{i \in V} x_i \tag{5.4}$$

$$r(x_i + x_j - 1) \leq \mathbb{1}_E(i,j) + \sum_{k \in \Delta_{ij}^r} x_k \qquad \forall \{i,j\} \in \binom{V}{2} : |\Delta_{ij}^r| \geq r - \mathbb{1}_E(i,j) \tag{5.5}$$

$$x_i + x_j \leq 1 \qquad \forall \{i,j\} \in \binom{V}{2} : |\Delta_{ij}^r| \leq r - 1 - \mathbb{1}_E(i,j) \tag{5.6}$$

$$x_i \in \{0,1\} \qquad \forall i \in V \tag{5.7}$$

Now let us use the example shown in Figure 5.1 to illustrate how we strengthen constraints (5.2). Consider the case $r = 2$ and a pair of non-adjacent vertices $\{2, 5\}$, and we have $\mathbb{1}_E(2,5) = 0$ and thus $2(x_2 + x_5 - 1) \leq x_1 + x_3$. Suppose both vertices $\{2, 5\}$ are included in the solution, i.e., $x_2 = x_5 = 1$, and it implies that $x_1 = x_3 = 1$. However, vertices 1 and 2 have no common neighbors and cannot be selected together in a solution. Hence, $x_2 + x_5 \leq 1$

is valid, and stronger than the original constraint $2(x_2 + x_5 - 1) \leq x_1 + x_3$. In fact, $\Delta^2_{2,5} = \emptyset$ and we have $x_2 + x_5 \leq 1$ according to the strengthened formulation (5.5)–(5.6).

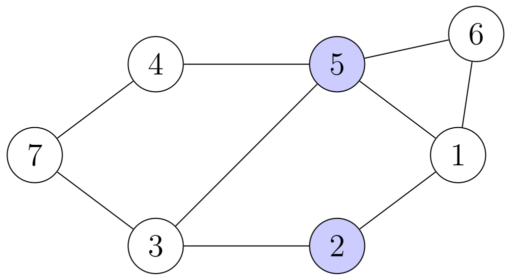

Figure 5.1: An example for illustrating the strengthened formulation

Clearly, SF is at least as strong as OF. The following lemma show that if the edges of the graph have already been recursively edge peeled to the maximal extent, constraints (5.5) and its counterpart in OF coincide.

Lemma 5. *Consider a graph $G = (V, E)$ such that $|N(i) \cap N(j)| \geq r - 1$ for every $\{i, j\} \in E$. Then $\Delta^r_{ij} = N(i) \cap N(j)$.*

Proof. Let $\{i, j\} \in \binom{V}{2}$. By definition, $\Delta^r_{ij} \subseteq N(i) \cap N(j)$. Suppose there exists a vertex $k \in (N(i) \cap N(j)) \setminus \Delta^r_{ij}$, then one of these conditions must hold: $|N(i) \cap N(k)| \leq r - 2$, $|N(j) \cap N(k)| \leq r - 2$. However, this is not possible under the given condition as $\{i, k\} \in E$ and $\{j, k\} \in E$. $\square$

5.2 Extended IP formulation

Recall that in Section 4.4 few implementation questions were considered in terms of how to take advantage of block decomposition. In this section, we will present an extended formulation which circumvents these questions. The comparison of their computational performances will be discussed in Section 5.4. First, we will present the following lemma used in the extended formulation.

Lemma 6. *Consider a block B of graph $G = (V, E)$. If a pair of vertices $i, j \in B$, then $N(i) \cap N(j)$ must be contained in block B.*

Proof. We prove this lemma by contradiction. Suppose there exists a vertex $k \in (N(i) \cap N(j)) \setminus V(B)$. Recall that B is a maximal biconnected subgraph, and therefore any two vertices of $V(B)$ are connected by at least two vertex-disjoint paths [Whitney, 1992]. It implies that there is some vertex $w \in V(B)$ such that there is only one vertex-disjoint path between k and w in subgraph $G[V(B) \cup \{k\}]$. Since there are two vertex-disjoint paths between i and j in subgraph $G[B]$, and $k \in N(i) \cap N(j)$, there must exist two vertex-disjoint paths between i and k, j and k respectively in subgraph $G[V(B) \cup \{k\}]$. Hence, w is distinct from i and j.

It follows that there are two vertex-disjoint paths p_{wi}, p'_{wi} between i and w in subgraph $G[B]$. Similarly, there also exist two vertex-disjoint paths p_{wj}, p'_{wj} between j and w in subgraph $G[B]$. Without loss of generality, we can assume that p_{wi} does not contain j and p_{wj} does not contain i. Then, there are two vertex-disjoint paths between k and w in subgraph $G[V(B) \cup \{k\}]$. This is a contradiction. $\square$

By Lemma 4, every r-robust 2-club must be contained in some block. So we can introduce a binary variable associated with each candidate block in the output of Algorithm 6 to build an extended formulation. Suppose $\mathcal{C}$ is the collection of candidate blocks that potentially contain a maximum r-robust s-club in G. For each block $B \in \mathcal{C}$, we associate a binary variable y_B. With the x variables defined as before, we obtain the following formulation.

$$\textbf{(EXTSF)} \quad \max \sum_{i \in V} x_i \tag{5.8}$$

$$\text{s.t.} \quad \sum_{B \in \mathcal{C}} y_B \leq 1 \tag{5.9}$$

$$x_i \leq \sum_{B \in \mathcal{C}: i \in V(B)} y_B \qquad \forall i \in V \tag{5.10}$$

$$r(x_i + x_j - 1) \leq \mathbb{1}_E(i,j) + \sum_{p \in \Delta^r_{ij}} x_p \quad \forall \{i,j\} \in \binom{V(B)}{2} : |\Delta^r_{ij}| \geq r - \mathbb{1}_E(i,j), B \in \mathcal{C} \tag{5.11}$$

$$x_i + x_j \leq 1 \qquad \forall \{i,j\} \in \binom{V(B)}{2} : |\Delta^r_{ij}| \leq r - 1 - \mathbb{1}_E(i,j), B \in \mathcal{C} \tag{5.12}$$

$$x_i \in \{0,1\} \qquad \forall i \in V \tag{5.13}$$

$$y_B \in \{0,1\} \qquad \forall B \in \mathcal{C} \tag{5.14}$$

Constraints (5.9) enforce that at most one block can be chosen, and constraints (5.10) ensure at least one block containing vertex i must be included if i is in a feasible solution. Since every non-trivial feasible solution must be contained in one block, it is sufficient to only consider pairs of vertices in the same block that have at least r vertex-disjoint paths of length at most two between them, as imposed by constraints (5.11). It is also worth noting that $\Delta^r_{ij} \cap V(B) = \Delta^r_{ij}$ by Lemma 6, and thus we only write Δ^r_{ij} instead of $\Delta^r_{ij} \cap V(B)$ in constraints (5.11)–(5.12).

5.3 Branch-and-Cut algorithm

Although OF and SF have $O(|V|^2)$ constraints, we propose a BC algorithm based on a delayed constraint generation scheme. Since SF is at least as strong as OF in the original graph without any preprocessing, we employ a subset of constraints of SF in a master relaxation which is presented next.

$$\textbf{(MRP)} \quad \max \sum_{i \in V} x_i$$

$$x_i + x_j \leq 1 \qquad \forall \{i,j\} \in \binom{V}{2} : |\Delta^r_{ij}| \leq r - 1 - \mathbb{1}_E(i,j)$$

$$x_i \in \{0,1\} \qquad \forall i \in V$$

In the BC algorithm, we check if the subset S corresponding to the integral solution obtained at a BC node is an r-robust 2-club. In other words, for each pair of vertices $i, j \in S$, we verify if $\mathbb{1}_E(i,j) + |N(i) \cap N(j) \cap S| \geq r$. If not, we add the lazy constraint $r(x_i + x_j - 1) \leq \mathbb{1}_E(i,j) + \sum_{k \in \Delta_{ij}^r} x_k$ to cut off this infeasible integral solution. Notice that we may be able to add many lazy constraints for each integral solution. Buchanan and Salemi [2017] pointed out that a similar approach also improved the running time for the maximum 2-club problem. In this dissertation, we compare performances between this BC algorithm and directly solving strengthened formulation using a general purpose solver.

5.4 Computational experiments

The goal of the computational experiments is to study the effectiveness of strengthened formulation, preprocessing techniques and B/B decomposition algorithm proposed in Chapter IV for solving the maximum r-robust 2-club problem. Extensions of these approaches are also assessed for the maximum r-hereditary 2-club and biconnected 2-club problems. In order to test the performance of these approaches, we select large-scale real-life network instances from the Tenth DIMACS Implementation Challenge [Bader et al., 2013] and Stanford Network Analysis Platform (SNAP) [Leskovec and Krevl, 2014], and also generate uniform random graphs (URG) by the procedure outlined in [Gendreau et al., 1993] as benchmarks in the computational study. Experimental settings including software and hardware are detailed next.

5.4.1 Test-bed description

We use 21 graph instances from the Tenth DIMACS Implementation Challenge [Bader et al., 2013], out of which 14 are often used as benchmarks for graph clustering and community detection. The remaining six graphs come from Walshaw's Graph Partitioning Archive that are used as benchmarks for graph partitioning algorithms [Khandekar et al., 2009, Meyerhenke

Table 5.1: Number of vertices, edges, and edge density for the Tenth DIMACS benchmarks used in this study.

Graph	$\|V\|$	$\|E\|$	ρ
karate	34	78	13.9%
dolphins	62	159	8.41%
lesmis	77	254	8.68 %
polbooks	105	441	8.08%
adjnoun	112	425	6.84%
football	115	613	9.35%
jazz	198	2742	14.06%
celegans_metabolic	453	2025	1.98%
email	1133	5451	0.85%
polblogs	1490	16715	1.51%
netscience	1589	2742	0.22%
add20	2395	7462	0.26%
data	2851	15093	0.37%
uk	4824	6837	0.06%
power	4941	6594	0.05%
add32	4960	9462	0.08%
hep-th	8361	15751	0.05%
whitaker3	9800	28989	0.06%
crack	10240	30380	0.058%
PGPgiantcompo	10680	24316	0.04%
cs4	22499	43858	0.017%

et al., 2009, Rahimian et al., 2015]. The number of vertices ($|V|$) and edges ($|E|$), and edge density (ρ) of each graph are summarized in Table 5.1. These graphs are typically sparse in terms of their density and the number of vertices varies in the range of 34 to 22499. These instances were also utilized by Yezerska et al. [2017] for the maximum biconnected 2-club problem.

The second set of benchmarks containing 12 undirected graphs is chosen from SNAP collection. These instances are relatively large where $|V|$ varies from 3892 to 54573, and $|E|$ varies from 17262 to 819306. The detailed information of SNAP instances is summarized in Table 5.2.

Our last set of benchmarks are synthetic instances generated by the procedure introduced

Table 5.2: Number of vertices, edges, and edge density for the SNAP benchmarks used in this study.

Graph	$\|V\|$	$\|E\|$	ρ
tvshow	3892	17262	0.228%
ego-Facebook	4039	88235	1.082%
politician	5908	41729	0.239%
government	7057	89455	0.359%
public_figure	11565	67114	0.100%
athletes	13866	86858	0.090%
company	14113	52310	0.053%
new_sites	27917	206259	0.053%
RO	41773	125826	0.014%
HU	47538	222887	0.020%
artist	50515	819306	0.064%
HR	54573	498202	0.033%

in [Gendreau et al., 1993]. In this generator, there are two parameters a and b controlling the expected density (ρ) of the generated graph, which is equal to $(a+b)/2$. The vertex degree variance increases as a function of $b-a$. On this test-bed, we set $a=b$ for all instances we generate. Under such circumstances, $\rho = a = b$, and vertex degree variance is a minimum, which discourages the vertex with the maximum degree and its associated neighbors from becoming the optimal solution. We generate 30 instances with 200 vertices where edge density is respectively set to $5\%, 10\%$ and 15% for every 10 samples. Similarly, we set $\rho = 2\%, 3\%, 4\%$ for each 10 of 30 instances with 1000 vertices, and edge density of last 30 instances with 2000 vertices are $0.5\%, 1\%, 1.5\%$ respectively.

5.4.2 Experimental settings

All approaches developed in this computational study are implemented in C++, and Gurobi™ Optimizer 8.1.0 [Gurobi Optimization, Inc., 2019] is employed to solve the IP formulations. We impose a three-hour time limit on the solve-time parameter of Gurobi. But it is worth noting that this only limits on the model solving time, that is, the time spent on each BC call. All other Gurobi parameters are left at their default setting. We conduct all numerical experiments on a 64-bit Linux® compute node with dual Intel® Xeon® E5-2620 hex-core 2.0GHz processors and 32 GB RAM.

5.4.3 Results for the maximum r-robust 2-club problem

In this section, we will consider a computational study to test the performance of our approaches for the maximum r-robust 2-club problem. Specifically, the aims of the numerical experiment are as follows.

(i) Demonstrate the effectiveness of strengthened formulation presented in Section 5.1.

(ii) Assess the performance of preprocessing techniques introduced in Section 4.2.1.

(iii) Evaluate the BC algorithm introduced in Section 5.3.

(iv) Assess the performance of extended formulation presented in Section 5.2.

(v) Assess the performance of the B/B decomposition algorithm described in Section 4.4.

Accordingly, we consider six solvers in experiments.

Table 5.3: Features of six solvers used in this computational study

Solvers		OF	SF	VP	EP	BC	Recursive Block	EXT	B/B
Solver 1	OF	★							
Solver 2	SF		★						
Solver 3	BCSF		★			★			
Solver 4	PPF		★	★	★				
Solver 5	EXT		★	★	★	★	★	★	
Solver 6	B/B		★	★	★	★	★		★

In order to test the effectiveness of SF, we compare it against OF by directly solving formulation (5.4)–(5.7) and formulation (5.1)–(5.3) introduced in Section 5.1. They respectively serve as the first (OF) and second solvers (SF). The third solver called BCSF employs SF in a delayed constraint generation scheme. We solve SF on the graph modified by preprocessing including vertex and edge peeling and call this fourth solver as PPF. We employ the extended formulation (5.8)–(5.13) along with Vertex Peeling, Edge Peeling, and Recursive Block, which serves as the fifth solver, referred to as EXT. We employ the Algorithm 7 and SF which is the sixth solver called B/B. The differences between these six solvers are summarized in Table 5.3.

Results for DIMACS instances

The graph obtained after applying Algorithms 3 and 5 is denoted by $G' = (V', E')$. Note that we do not delete vertices by Algorithm 3, instead, all edges incident with such vertices are removed. Hence, vertices of graph G' remain unchanged and $V = V'$. We report heuristic solution size, the number of new edges $|E'|$ and blocks, and total time in seconds for the heuristic and preprocessing on the DIMACS instances for $r = 2, \ldots, 7$ in Table 5.4. Note that as long as we identify a heuristic solution of size ℓ, we delete vertices that cannot be choose in a solution whose size is at least ℓ. That is why for some instances the heuristic solution size is a positive integer but $|E'| = 0$. As expected, the number of edges in graph G' significantly decreases as the parameter r increases. Especially, the number of edges for some instances such as graph “whitaker3”, “crack” and “cs4” becomes zero when $r = 3$, in which case the maximum r-robust 2 club size is one.

After these preprocessing techniques, the resulting graph G' becomes more sparse and thus it is more likely to decompose into smaller blocks. Additionally, as the parameter r increases, G' tends to decompose into more blocks. It turns out that heuristic for many instances is an optimal solution especially for $r = 2$. Furthermore, the total time for heuristic and preprocessing is under two seconds for every instance. In order to simplify the discussion, we divide the instances into two groups. The first group consists of instances with fewer than 1000 vertices and the second consists of the remaining instances. The two groups are separated by a line in Tables 5.1, 5.4, and Tables 5.6 through 5.11. So our focus will be on group two in this discussion.

Table 5.4: Heuristic solution size (Heur), the number of new edges $|E'|$ and blocks, and total time in seconds for the heuristic and preprocessing (time) on the DIMACS instances for $r = 2, \ldots, 7$

Graph	$\lvert E \rvert$	Heur	$\lvert E' \rvert^\dagger$	#blocks	time	Heur	$\lvert E' \rvert^\dagger$	#blocks	time	Heur	$\lvert E' \rvert^\dagger$	#blocks	time
				$r = 2$				$r = 3$				$r = 4$	
karate	78	12	0	34	0.00	6	0	34	0.00	6	0	34	0.00
dolphins	159	9	0	62	0.00	6	35	49	0.00	6	0	62	0.00
lesmis	254	18	0	77	0.00	14	0	77	0.00	13	0	77	0.00
polbooks	441	16	284	44	0.01	10	232	56	0.01	9	139	75	0.01
adjnoun	425	23	176	72	0.01	12	0	112	0.00	6	0	112	0.00
football	613	10	512	2	0.01	7	408	21	0.01	9	180	74	0.00
jazz	2742	71	2147	62	0.15	58	2056	70	0.13	47	1955	79	0.14
celegans	2025	104	0	453	0.03	54	0	453	0.04	30	0	453	0.02
email	5451	23	2228	812	0.28	18	803	1008	0.10	16	323	1083	0.03
polblogs	16715	232	14230	819	1.68	137	14146	831	1.93	90	13319	883	1.28
netscience	2742	22	0	1589	0.03	21	0	1589	0.03	20	0	1589	0.02
add20	7462	91	0	2395	0.16	77	0	2395	0.17	71	1479	2320	0.12
data	15093	13	11636	901	1.53	6	11666	896	2.38	1	11663	897	0.14
uk	6837	3	0	4824	0.10	1	0	4824	0.07	1	0	4824	0.06
power	6594	9	0	4941	0.18	7	0	4941	0.10	6	0	4941	0.10
add32	9462	12	0	4960	0.21	5	0	4960	0.22	1	0	4960	0.06
hep-th	15751	33	0	8361	0.42	24	0	8361	0.41	24	0	8361	0.28
whitaker3	28989	4	28989	1	0.36	1	0	9800	0.21	1	0	9800	0.21
crack	30380	4	30380	1	0.38	1	0	10240	0.36	1	0	10240	0.23
PGP	24316	96	2125	10539	0.95	60	1496	10588	0.74	43	2555	10528	0.93
cs4	43858	3	0	22499	1.68	1	0	22499	0.93	1	0	22499	0.93
				$r = 5$				$r = 6$				$r = 7$	
karate	78	1	0	34	0.00	1	0	34	0.00	1	0	34	0.00
dolphins	159	1	0	62	0.00	1	0	62	0.00	1	0	62	0.00
lesmis	254	13	0	77	0.00	13	0	77	0.00	12	0	77	0.00
polbooks	441	7	37	96	0.00	1	0	105	0.00	1	0	105	0.00
adjnoun	425	1	0	112	0.00	1	0	112	0.00	1	0	112	0.00
football	613	9	180	74	0.00	9	84	96	0.00	1	168	76	0.00
jazz	2742	47	1867	86	0.14	47	1708	98	0.10	45	1611	103	0.12
celegans	2025	19	260	418	0.01	14	249	419	0.01	10	122	434	0.01
email	5451	14	94	1118	0.02	13	81	1120	0.01	13	0	1133	0.01
polblogs	16715	67	12335	949	1.38	59	11440	1001	1.28	47	10474	1053	0.74
netscience	2742	20	0	1589	0.02	20	0	1589	0.02	20	0	1589	0.02
add20	7462	68	1479	2320	0.10	68	1368	2327	0.11	68	1368	2327	0.10
data	15093	1	11663	897	0.14	1	0	2851	0.03	1	0	2851	0.03
uk	6837	1	0	4824	0.06	1	0	4824	0.06	1	0	4824	0.08
power	6594	6	0	4941	0.10	1	0	4941	0.06	1	0	4941	0.07
add32	9462	1	0	4960	0.06	1	0	4960	0.06	1	0	4960	0.08
hep-th	15751	24	0	8361	0.27	24	0	8361	0.26	24	0	8361	0.27
whitaker3	28989	1	0	9800	0.20	1	0	9800	0.20	1	0	9800	0.20
crack	30380	1	0	10240	0.22	1	0	10240	0.22	1	0	10240	0.22
PGP	24316	42	2379	10544	0.73	41	2226	10552	0.73	40	1921	10573	0.87
cs4	43858	1	0	22499	0.93	1	0	22499	0.93	1	0	22499	0.93

† Some values are zero because we remove vertices that cannot be choose in a solution whose size is *equal to or greater than* heuristic solution size ℓ.

We solve the maximum r-robust 2-club problem for $r \in \{2, \ldots, 7\}$. As expected, OF versus SF, SF versus BCSF, and EXT versus B/B are all comparable for all parameters $r \in \{2, \ldots, 7\}$. The best objective and wall-clock running times including reading graph, preprocessing, model building and solving of each graph are reported in Tables 5.6 through 5.11 for sixth different solvers. The fastest running times are highlighted in bold font. For sub-optimal terminnation, we report their optimality gaps calculated as 100∗(UB−LB)/LB% where LB represents the current best objective and UB corresponds to the best upper bound.

Table 5.5: The root node optimality gap between the objective of LP relaxation and IP optimal value is summarized for OF and SF

	$r = 2$		$r = 3$		$r = 4$		$r = 5$		$r = 6$		$r = 7$	
Graph	OF	SF	OF	SF	OF	SF	OF	SF	OF	SF	OF	SF
karate	0%	0%	0%	0%	0%	0%	*	*	*	*	*	*
dolphins	2%	0%	0%	0%	0%	0%	*	*	*	*	*	*
lesmis	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
polbooks	4%	0%	7%	0%	8%	0%	0%	0%	*	*	*	*
adjnoun	0%	0%	0%	0%	0%	0%	*	*	*	*	*	*
football	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
jazz	0%	0%	0%	0%	0%	0%	1%	1%	6%	0%	3%	2%
celegans	0%	0%	38%	0%	109%	0%	123%	5%	59%	11%	85%	3%
email	109%	38%	94%	19%	116%	13%	111%	0%	117%	0%	75%	0%
polblogs	3%	0%	14%	2%	19%	3%	21%	1%	22%	1%	22%	1%
netscience	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
add20	0%	0%	3%	0%	0%	0%	0%	0%	0%	0%	0%	0%
data	121%	121%	149%	62%	266%	106%	263%	90%	2590%	100%	2590%	100%
uk	10%	0%	250%	*	250%	*	250%	*	250%	*	250%	*
power	94%	11%	119%	0%	92%	0%	44%	0%	671%	*	375%	*
add32	208%	51%	564%	108%	3040%	305%	2800%	*	2380%	*	2230%	*
hep-th	38%	0%	84%	0%	75%	0%	65%	0%	–	0%	–	0%
whitaker3	167%	83%	1450%	500%	–	*	–	*	–	*	–	*
crack	199%	78%	1600%	600%	1280%	*	1280%	*	1280%	*	1280%	*
PGP	–	–	–	–	–	1%	–	0%	–	0%	–	0%
cs4	–	–	–	–	–	–	–	–	–	–	–	–

– indicates that root relaxation was not solved to optimality under the 3-hour time limit
* indicates that solving the LP relaxation directly solves the IP

We begin our discussion with the comparison of SF and OF. For all instances, SF outperforms OF for each $r \in \{2, \ldots, 7\}$ in terms of running times. For some challenging instances, both solvers fail to solve to optimality, but the optimality gap using SF is smaller than OF. Furthermore, to compare the strength between OF and SF, we also retrieve their

Table 5.6: The 2-robust 2-club number and the running time in seconds for OF, SF, BCSF, PPF, EXT and B/B on the DIMACS test-bed

Graph	BObj	OF	SF	BCSF	PPF	EXT	B/B
karate	12	0.58	0.34	0.1	0.04	0.02	**0.00**
dolphins	9	0.17	0.07	0.08	0.21	0.07	**0.00**
lesmis	18	0.18	0.12	0.11	0.06	0.02	**0.00**
polbooks	20	0.39	0.20	0.16	0.24	**0.05**	0.54
adjnoun	23	0.34	0.27	0.19	0.24	**0.07**	0.09
football	14	0.33	0.28	0.17	0.26	**0.21**	0.33
jazz	79	1.19	1.16	0.60	1.15	0.79	**0.49**
celegans	104	4.53	3.87	2.21	0.81	0.09	**0.03**
email	27	584.94	192.99	109.09	52.22	6.78	**5.09**
polblogs	232	2488.86	151.61	51.44	99.25	18.79	**13.86**
netscience	22	57.62	52.38	40.23	17.68	0.16	**0.04**
add20	91	187.88	175.39	145.20	51.74	0.52	**0.18**
data	14	6667.93	4717.73	4431.41	2940.71	**1230.59**	1242.48
uk	3	839.88	496.96	460.09	346.33	0.44	**0.18**
power	9	6820.59	1095.39	868.80	370.20	0.64	**0.27**
add32	12	4917.89	3496.10	6020.98	374.18	0.58	**0.30**
hep-th	33	8965.72	3487.49	3439.43	1623.60	1.28	**0.67**
whitaker3	6	14517\100%†	14717\33%†	13368.10	14710\33%†	**12000.40**	12807.40
crack	≥ 6	15508\183%†	17186\50%†	15449\50%†	15783\50%†	16250\50%†	15331\67%†
PGP	96	15330††	15519††	10856.00	3889.03	7.22	**1.69**
cs4	3	–	–	–	–	**3.00**	3.51

– The approach did not terminate gracefully, typically due to a memory-related crash
† indicates that the approach did not find an optimal solution under the 3-hour time limit and optimality gap is reported
†† indicates that root relaxation was not solved to optimality under the 3-hour time limit

objectives of root relaxation and report the root node optimality gap in Table 5.5. We observe that there are more instances solved to optimality at root relaxation using SF than OF. Additionally, for the instances whose relaxations are not optimal, the gaps by SF are smaller than the ones by OF. These results also demonstrate that SF is stronger than OF on this test bed.

To assess the performance of BC algorithm presented in Section 5.3, we compare BCSF against SF. Recall that we add constraints (5.6) on-the-fly only needed for the solver BCSF. Though our strengthened formulation has $O(|V|^2)$ constraints, BCSF outperforms SF for nearly all instances. Even challenging instances "PGPgiantcompo" and "whitaker3" when $r = 2$ are solved to optimality by BCSF, where SF fails to do so within a three-hour limit. There are only two exceptions, "add32" and "hep-th", on which SF is slightly better than

Table 5.7: The 3-robust 2-club number and the running time in seconds for OF, SF, BCSF, PPF, EXT and B/B on the DIMACS test-bed

Graph	BObj	OF	SF	BCSF	PPF	EXT	B/B
karate	6	0.05	0.08	0.10	0.07	0.09	**0.00**
dolphins	7	0.10	0.05	0.07	**0.04**	0.05	0.12
lesmis	14	0.15	0.10	0.10	0.04	0.04	**0.00**
polbooks	15	0.53	0.17	0.16	0.19	**0.07**	**0.07**
adjnoun	12	0.30	0.20	0.22	0.06	0.07	**0.00**
football	13	0.39	0.21	0.16	0.21	**0.08**	0.11
jazz	73	1.27	1.31	0.65	1.09	0.69	**0.50**
celegans	54	25.92	3.14	2.26	0.80	0.10	**0.04**
email	23	163.09	38.80	51.68	17.81	0.39	**0.35**
polblogs	182	2835.27	285.25	69.17	199.82	31.09	**26.42**
netscience	21	60.29	42.40	38.67	17.68	0.14	**0.04**
add20	77	366.67	146.27	149.13	51.79	0.33	**0.19**
data	12	4949.46	1481.93	1060.05	1154.85	**252.38**	263.00
uk	1	836.46	484.73	453.30	347.00	0.66	**0.18**
power	7	2165.10	508.09	464.94	370.33	0.71	**0.19**
add32	5	4313.07	1183.88	2409.23	374.20	0.98	**0.31**
hep-th	24	9603.61	3280.69	5348.96	1625.68	1.28	**0.66**
whitaker3	1	14765\1100%†	15396\500%†	14489\300%†	2614.84	1.80	**0.60**
crack	1	15516\1600%†	17070\600%†	$-$	2949.14	2.28	**0.79**
PGP	71	15341††	15526††	10149.80	4027.49	2.45	**1.39**
cs4	1	$-$	$-$	$-$	$-$	8.14	**2.86**

$^{-}$ The approach did not terminate gracefully, typically due to a memory-related crash
† indicates that the approach did not find an optimal solution under the 3-hour time limit and optimality gap is reported
†† indicates that root relaxation was not solved to optimality under the 3-hour time limit

BCSF for some values of parameter r.

Now we turn our attention to the performance of PPF. We observe that PPF outperforms OF and SF on nearly all instances. Especially for the challenging instances with large vertices such as "hep-th", "whitaker3", "crack" and "PGP", in which it usually takes PPF around one hour, but average running times are around three hours by OF/SF for most values of parameter r. By far, computational results show that the solver SF is better than OF, BCSF outperforms SF, and preprocessing techniques play a significant role to speed up algorithms. Therefore, we employ BCSF and preprocessing techniques in the remaining two solvers EXT and B/B.

Table 5.8: The 4-robust 2-club number and the running time in seconds for OF, SF, BCSF, PPF, EXT and B/B on the DIMACS test-bed

Graph	BObj	OF	SF	BCSF	PPF	EXT	B/B
karate	6	0.11	0.08	0.11	0.07	0.11	**0.00**
dolphins	6	0.09	0.04	0.08	0.03	0.05	**0.00**
lesmis	13	0.18	0.10	0.11	0.03	0.04	**0.00**
polbooks	12	0.44	0.24	0.20	0.18	**0.05**	0.12
adjnoun	6	0.22	0.08	0.15	0.06	0.06	**0.00**
football	12	0.32	0.20	0.22	0.19	0.15	**0.04**
jazz	65	1.45	1.18	0.66	1.07	0.64	**0.46**
celegans	30	13.88	3.22	2.54	0.78	0.08	**0.02**
email	19	145.17	38.11	26.04	17.92	0.15	**0.11**
polblogs	158	2565.27	332.73	86.75	331.24	27.74	**20.41**
netscience	20	63.96	38.86	35.60	18.85	0.11	**0.03**
add20	75	267.77	136.06	177.75	63.97	0.55	**0.37**
data	8	3195.96	966.54	995.27	999.69	472.20	**330.49**
uk	1	1005.82	481.45	455.83	350.24	0.63	**0.18**
power	6	1320.37	506.76	467.65	379.43	0.61	**0.19**
add32	1	3543.26	782.52	1168.93	377.12	0.65	**0.23**
hep-th	24	6169.38	3506.48	3238.74	1664.70	1.17	**0.53**
whitaker3	1	14499††	4073.47	3787.91	2578.66	1.83	**0.59**
crack	1	15488\1100%†	5045.93	4787.60	2940.19	2.03	**0.65**
PGP	64	15331††	15516\6300%†	9674.60	4340.26	2.86	**1.67**
cs4	1	–	–	–	–	8.36	**2.88**

$^{-}$ The approach did not terminate gracefully, typically due to a memory-related crash
† indicates that the approach did not find an optimal solution under the 3-hour time limit and optimality gap is reported
†† indicates that root relaxation was not solved to optimality under the 3-hour time limit

We finally move our discussion to EXT and B/B. The running times of EXT and B/B are just a few seconds for most instances, even some of them that take thousands of seconds using OF. Both EXT and B/B solve all instances to optimality except "crack" when $r = 2$. Especially, for the most challenging instance "cs4", it only takes B/B around three seconds. However, all other four approaches struggle for "cs4" and they did not terminate gracefully due to memory-related crash. There is another interesting observation that as the parameter r increases, both EXT and B/B become more effective, i.e. their running times decrease. This can be attributed to more edges being deleted by the preprocessing technique and the modified graph decomposing into many smaller blocks when the parameter r increases. In

Table 5.9: The 5-robust 2-club number and the running time in seconds for OF, SF, BCSF, PPF, EXT and B/B on the DIMACS test-bed

Graph	BObj	OF	SF	BCSF	PPSF	EXT	B/B
karate	1	0.09	0.52	**0.05**	0.14	0.15	0.11
dolphins	1	0.07	0.06	0.06	0.06	0.06	**0.03**
lesmis	13	0.16	0.13	0.20	0.06	0.06	**0.00**
polbooks	10	0.37	0.16	0.17	0.10	**0.04**	0.07
adjnoun	1	0.15	0.10	0.11	0.15	**0.07**	0.10
football	12	0.33	0.47	0.16	0.21	**0.04**	**0.04**
jazz	60	1.56	1.17	0.73	1.06	0.67	**0.50**
celegans	20	9.21	3.82	2.70	1.30	0.09	**0.07**
email	16	55.78	28.19	23.88	8.80	0.10	**0.08**
polblogs	146	2621.69	91.41	76.35	148.19	16.54	**12.54**
netscience	20	66.35	40.58	37.67	17.66	0.12	**0.03**
add20	70	301.97	134.20	139.20	62.34	0.52	**0.39**
data	8	2875.27	372.77	571.39	367.45	456.20	**131.78**
uk	1	836.44	482.67	454.48	345.97	0.55	**0.30**
power	6	966.44	540.68	509.41	370.46	0.45	**0.19**
add32	1	2797.57	538.57	521.44	374.18	0.59	**0.21**
hep-th	24	5468.22	2479.65	2357.58	1655.71	1.15	**0.52**
whitaker3	1	14517.1††	4072.19	3776.61	2612.31	1.79	**0.58**
crack	1	15512\1100%†	5045.29	4770.65	2966.23	1.95	**0.65**
PGP	57	15350.3††	5329.66	5735.65	3965.14	2.52	**1.50**
cs4	1	–	–	–	–	8.16	**2.88**

– means that the approach did not terminate gracefully, typically due to a memory-related crash
† indicates that the approach did not find an optimal solution under the 3-hour time limit and optimality gap is reported
†† indicates that root relaxation was not solved to optimality under the 3-hour time limit

the comparison between EXT and B/B, the latter is slightly better for most instances.

In comparison to PPF, both EXT and B/B have superior performance in terms of average running times. To summarize, the B/B outperforms all other approaches, and the PPF dominates the direct solution of formulations for the large scale real-life instances. These comparisons directly or indirectly demonstrate that SF is stronger than OF, as we expected.

Results for SNAP instances

In this part, we will run our experiment on SNAP instances which are typically larger than DIMACS instances. The purpose is to test if our approaches are still effective on larger

Table 5.10: The 6-robust 2-club number and the running time in seconds for OF, SF, BCSF, PPF, EXT and B/B on the DIMACS test-bed

Graph	BObj	OF	SF	BCSF	PPSF	EXT	B/B
karate	1	0.26	0.09	**0.08**	0.09	**0.08**	0.09
dolphins	1	0.06	**0.04**	0.07	0.07	0.06	0.06
lesmis	13	0.21	0.12	0.11	0.04	0.03	**0.00**
polbooks	1	0.20	0.09	0.11	0.08	**0.03**	0.06
adjnoun	1	0.14	0.07	0.12	0.07	0.07	**0.05**
football	11	0.37	0.22	0.20	0.18	0.05	**0.04**
jazz	51	5.35	1.25	0.83	0.95	0.55	**0.39**
celegans	16	5.63	2.08	1.84	1.29	0.10	**0.09**
email	14	54.61	23.65	20.31	8.61	0.10	**0.09**
polblogs	134	1934.23	80.65	52.71	65.63	12.38	**9.12**
netscience	20	67.15	48.39	46.25	17.52	0.18	**0.03**
add20	69	269.24	134.35	140.03	61.46	0.49	**0.36**
data	1	494.29	285.51	288.15	81.68	0.35	**0.15**
uk	1	839.03	480.44	456.31	345.20	0.53	**0.16**
power	1	1493.02	481.50	455.68	369.30	0.56	**0.18**
add32	1	2050.99	537.33	512.03	373.06	0.59	**0.20**
hep-th	24	13038.5††	2438.72	2330.09	1655.92	1.12	**0.52**
whitaker3	1	14513.5††	4060.14	3789.24	2576.38	1.81	**0.59**
crack	1	15505\1100%†	5039.55	4791.10	2968.82	1.96	**0.64**
PGP	53	15355.4††	5260.05	4924.18	3918.13	2.51	**1.41**
cs4	1	–	–	–	–	8.24	**2.83**

$^{-}$ means that the approach did not terminate gracefully, typically due to a memory-related crash
† indicates that the approach did not find an optimal solution under the 3-hour time limit and optimality gap is reported
†† indicates that root relaxation was not solved to optimality under the 3-hour time limit

real-life instances. Based on the results of DIMACS instances, we observe that detected maximum r-robust 2-club is very small when parameter r becomes big. It essentially turns out to be a clique and there might not be meaningful insights as a result. Therefore, we only consider $r = 2, 3$ and 4 for SNAP instances.

In order to assess the performance of preprocessing and decomposition approaches, we report heuristic solution size, the number of new edges $|E'|$ and the number of blocks, and total time in seconds for heuristic and preprocessing on the SNAP instances when $r = 2, 3, 4$ in Table 5.12. From the table, we can see that the number of edges decreases significantly and it indicates that preprocessing techniques play a key role. Furthermore, as expected G'

Table 5.11: The 7-robust 2-club number and the running time in seconds for OF, SF, BCSF, PPF, EXT and B/B on the DIMACS test-bed

Graph	BObj	OF	SF	BCSF	PPSF	EXT	B/B
karate	1	0.13	0.11	**0.03**	0.08	**0.03**	0.07
dolphins	1	0.06	0.06	0.06	0.05	**0.04**	0.06
lesmis	12	0.21	0.12	0.10	0.06	0.02	**0.00**
polbooks	1	0.22	0.07	0.12	0.07	**0.03**	0.05
adjnoun	1	0.15	0.08	0.10	0.08	**0.05**	**0.05**
football	10	0.36	0.17	0.22	0.20	**0.04**	**0.04**
jazz	50	3.71	1.23	0.73	0.88	0.56	**0.40**
celegans	12	5.10	1.75	1.77	1.08	0.07	**0.06**
email	13	42.17	20.67	19.92	7.40	0.15	**0.02**
polblogs	124	1787.22	79.91	58.50	49.67	12.47	**6.47**
netscience	20	64.56	48.47	61.19	17.54	0.10	**0.03**
add20	69	245.62	131.23	163.99	61.49	0.49	**0.31**
data	1	475.07	295.67	331.64	81.60	0.29	**0.14**
uk	1	841.05	481.68	460.05	344.67	0.55	**0.24**
power	1	856.84	483.68	459.86	368.64	0.57	**0.22**
add32	1	2136.66	539.79	517.06	372.49	0.59	**0.23**
hep-th	24	$13033^{\dagger\dagger}$	2476.56	2358.03	1620.52	1.13	**0.52**
whitaker3	1	$14510.8^{\dagger\dagger}$	4071.40	3765.63	2613.10	1.79	**0.57**
crack	1	15499\1100%†	5041.78	4789.68	2966.55	1.96	**0.63**
PGP	47	$15339.5^{\dagger\dagger}$	5275.72	4956.88	3815.81	2.58	**1.41**
cs4	1	–	–	–	–	8.33	**2.83**

$^{-}$ means that the approach did not terminate gracefully, typically due to a memory-related crash

† indicates that the approach did not find an optimal solution under the 3-hour time limit and optimality gap is reported

†† indicates that root relaxation was not solved to optimality under the 3-hour time limit

Table 5.12: Heuristic (Heur), the number of new edges $|E'|$ and blocks, and total time in seconds for the heuristic solution size and preprocessing (time) on the SNAP instances for $r = 2, 3, 4$

		$r=2$				$r=3$				$r=4$			
Graph	$\lvert E\rvert$	Heur	$\lvert E'\rvert$	#blocks	time	Heur	$\lvert E'\rvert$	#blocks	time	Heur	$\lvert E'\rvert$	#blocks	time
tvshow	17262	96	2710	3791	0.23	95	2706	3792	0.27	93	2662	3795	0.26
ego-Facebook	88235	295	41086	3280	5.92	214	39339	3310	7.77	198	34391	3418	8.50
politician	41729	190	4524	5694	1.20	140	7038	5518	1.54	93	9487	5397	1.90
government	89455	352	32414	5656	10.14	155	38021	5444	14.75	101	42862	5250	14.51
public_figure	67114	161	18486	10793	4.96	67	21462	10519	9.22	45	18919	10746	6.19
athletes	86858	172	0	13866	2.84	53	9628	13028	4.59	50	3239	13620	2.35
company	52310	77	0	14113	1.19	36	2576	13866	1.23	35	2792	13861	0.86
new_sites	206259	164	30987	25703	39.55	50	40180	25007	40.76	43	20128	26457	14.77
RO	125826	41	0	41773	10.71	7	3925	40722	15.70	9	159	41734	8.05
HU	222887	30	1380	47360	22.31	21	1853	47290	18.19	24	711	47453	13.78
artist	819306	737	351641	40642	795.60	417	275983	43166	591.74	125	306691	41018	529.31
HR	498202	71	159433	42970	185.91	23	185074	38197	118.42	19	124853	43136	95.48

Table 5.13: The best objective and running time in seconds for OF, SF, BCSF, PPF, EXT and B/B on the SNAP test-bed

Graph	BObj	OF	SF	BCSF	PPF	EXT	B/B
			$r = 2$				
tvshow	100	658.21	430.02	450.37	280.07	1.45	**0.78**
ego-Facebook	300	11234\426%†	860.86	1079.42	477.71	90.38	**50.53**
politician	190	11797††	1532.75	1715.61	958.06	3.87	**2.61**
government	670	3628.33	4659.97	5028.47	3547.13	96.21	**65.84**
public_figure	258	–	–	–	–	39.80	**27.52**
athletes	172	–	–	–	–	4.96	**3.54**
company	77	–	–	–	–	3.39	**1.91**
new_sites	435	–	–	–	–	263.32	**241.75**
RO	41	–	–	–	–	28.31	**17.02**
HU	43	–	–	–	–	43.91	**31.23**
artist	≥737	–	–	–	–	21057\1240%†	18793\1240%†
HR	≥71	–	–	–	–	–	10800††
			$r = 3$				
tvshow	97	524.1	415.65	466.76	274.62	1.49	**0.80**
ego-Facebook	274	10800 \83300%†	959.49	1089.77	446.44	92.47	**48.24**
politician	159	10801††	2400.15	1866.54	929.60	5.21	**3.10**
government	335	10802††	6013.73	4075.72	2794.12	189.33	**133.99**
public_figure	230	–	–	–	–	96.46	**65.32**
athletes	115	–	–	–	–	19.16	**15.84**
company	59	–	–	–	–	3.81	**2.16**
new_sites	77	–	–	–	–	5341.81	**5023.37**
RO	16	–	–	–	–	33.86	**22.22**
HU	34	–	–	–	–	40.92	**28.31**
artist	≥417	–	–	–	–	16029\1663%†	14620\1663%†
HR	≥23	–	–	–	–	–	24671.70††
			$r = 4$				
tvshow	97	975.81	388.96	466.74	272.60	1.56	**0.79**
ego-Facebook	259	11237\81900%†	995.60	986.05	418.07	73.67	**43.85**
politician	147	11807††	1607.57	1725.45	951.84	10.18	**5.38**
government	245	12745††	12846††	12772††	12150\148%†	2118.80	**1955.89**
public_figure	228	–	–	–	–	30.91	**21.18**
athletes	57	–	–	–	–	6.57	**3.93**
company	53	–	–	–	–	3.48	**1.74**
new_sites	55	–	–	–	–	**377.15**	390.74
RO	10	–	–	–	–	25.85	**14.42**
HU	28	–	–	–	–	36.64	**22.58**
artist	≥125	–	–	–	–	19288\7498%†	17584\7498%†
HR	≥19	–	–	–	–	18982\337%†	18867\59058%†

$^{-}$ The approach did not terminate gracefully, typically due to a memory-related crash
† indicates that the approach did not find an optimal solution under the 3-hour time limit and optimality gap is reported
†† indicates that root relaxation was not solved to optimality under the 3-hour time limit

decomposes into a large number of blocks. These results are consistent with our discussions with the DIMACS instances.

Now we turn our attention to the performance of different solvers. The best objectives, the running times on each SNAP instance used in this computational study are reported in Table 5.13. These instances are very challenging and thus all solvers except EXT and B/B did not terminate gracefully due to memory-related issues for 9 out of 12 instances. The solver OF can only solve instances "tvshow" ($r = 2, 3$ and 4) and "government" ($r = 2$) to optimality. It is encouraging that our best solver B/B is capable of solving all instances except "artist" and "HR".

Results for URG instances

Results for DIMACS and SNAP instances show that our approaches are very effective for solving large-scale real-life graphs. In this part, we will run our experiment on URG instances described in Section 5.4.1 and evaluate the performance of proposed approaches.

Table 5.14: Average number of edges ($|E|$) of original graph G, new edges ($|E'|$) and blocks (#block) of resulting graph G' on the URG instances is reported

			$r=2$			$r=3$			$r=4$		
$\lvert V\rvert$	$\rho(\%)$	$\lvert E\rvert$	$\lvert E'\rvert$	#blocks	BObj	$\lvert E'\rvert$	#blocks	BObj	$\lvert E'\rvert$	#blocks	BObj
	5	1001	9	196	6	0	200	2	0	200	1
200	10	1983	1704	2	11	170	140	6	0	200	1
	15	2970	2935	1	23	2742	1	11	16	194	6
	2	9973	0	1000	7	0	1000	4	0	1000	1
1000	3	14958	8821	9	9	1	1000	5	0	1000	1
	4	20070	16054	1	$\geq$11	378	849	6	0	1000	1
	0.5	10006	0	2000	4	0	2000	1	0	2000	1
2000	1	20025	0	2000	6	0	2000	3	0	2000	1
	1.5	29857	9154	328	7	0	2000	4	0	2000	1

To see the initial effect of preprocessing and decomposition, the average number of new edges and blocks of resulting graph G' and the average best objectives are reported in Table 5.14; all averages over 10 samples. The running times results for URG instances reported in Table 5.15 are consistent with those for DIMACS and SNAP we discussed previously, which show that SF is stronger than OF, and B/B is superior to all the others. It is worth noting

Table 5.15: A comparison of running time (seconds) averaged over 10 samples on URG instances; fastest(on average) running times are highlighted in bold font

$\|V\|$	$\rho(\%)$	OF	SF	BCSF	PPF	EXT	B/B
				$r=2$			
	5	1.99	0.83	0.67	0.24	0.08	**0.03**
200	10	36.60	28.81	56.05	**28.29**	50.90	54.87
	15	**1160.36**	1364.85	10802.91†	1368.40	10802.39†	10803.00†
	2	424.54	66.08	56.08	5.56	0.16	**0.07**
1000	3	7871.80	1174.21	476.61	1128.10	591.35	**463.36**
	4	10810.32†	10787.84†	10811.07†	10781.75†	10812.52†	10810.60†
	0.5	205.87	97.47	91.42	32.37	0.19	**0.04**
2000	1	606.87	219.88	204.43	32.37	0.17	**0.07**
	1.5	10859.68†	901.30	789.48	668.85	**426.50**	428.22
				$r=3$			
	5	0.36	0.24	0.26	0.20	0.05	**0.03**
200	10	5.02	2.05	3.93	0.53	0.20	**0.13**
	15	50.56	46.36	495.77	**46.01**	441.74	447.50
	2	21.87	9.82	9.77	5.40	0.08	**0.02**
1000	3	376.10	22.03	29.26	5.47	0.08	**0.02**
	4	4868.97	60.26	117.78	9.30	0.54	**0.26**
	0.5	78.70	52.39	48.74	32.00	0.17	**0.06**
2000	1	91.74	62.28	58.76	32.00	0.14	**0.06**
	1.5	1870.28	88.39	89.91	32.01	0.14	**0.05**
				$r=4$			
200	5	0.34	0.22	0.22	0.22	**0.04**	**0.04**
	10	1.11	0.34	0.87	0.20	**0.03**	**0.03**
	15	6.86	6.70	62.69	0.24	0.04	**0.03**
1000	2	15.67	9.25	8.82	5.46	0.09	**0.06**
	3	200.05	10.41	10.11	5.50	0.09	**0.05**
	4	383.45	19.32	45.54	5.50	0.09	**0.04**
2000	0.5	78.80	51.95	48.89	32.24	0.19	**0.06**
	1	89.20	60.31	57.84	32.23	0.19	**0.07**
	1.5	1213.46	66.56	64.12	32.27	0.19	**0.08**

† indicates that the approach did not find an optimal solution under the 3-hour time limit and optimality gap is reported

that all approaches have similar performance for the instances with $n = 1000, \rho = 4\%, r = 2$. This is because just a few edges are deleted by preprocessing techniques and graph G' only includes one block. And for such instances preprocessing techniques and B/B decomposition algorithm play a lesser role in improving the efficiency of solving the maximum r-robust

2-club problem. We also notice that for instances with $|V| = 200$ and $\rho = 15\%$, the solver BCSF did not solve to optimality, but OF and SF successfully solved them. This means that BC algorithm may perform worse than directly implementing IP formulation for some instances with higher edge density. Since EXT and B/B also employ the BC algorithm, both of them struggled to solve these instances.

5.4.4 Results for the maximum r-hereditary 2-club problem

In this part, we will assess the performance of approaches extended to the maximum r-hereditary 2-club problem. Specifically, preprocessing techniques and the B/B decomposition algorithm presented in Chapter IV for the $r, 2$-MHCP are evaluated via a computational study. To achieve this goal, we design two solvers and compare them. The first solver, referred to as OF, directly solves original formulation (2.46)–(2.48) of the maximum r-hereditary 2-club problem. Our second solver employs OF (2.46)–(2.48), Vertex Peeling-MHC presented in Algorithm 4 and B/B-MHC in Algorithm 9, which is called B/B. From the previous section, we already know that our approaches are effective to solve large-scale real-life instances and synthetic graphs. In this section, we only utilize DIMACS instances as our test-bed and choose parameter $r \in \{2, \dots, 7\}$.

The largest r-hereditary 2-club number and running times in seconds for OF and B/B on DIMACS test-beds are reported in Table 5.16. Clearly, the second solver (B/B) outperforms the first (OF) for nearly all instances. Especially for the challenging instances like “polblogs”, “power” and “hep-th”, B/BOF significantly reduces running times comparing with OF. One extreme case is the instance “PGP” in which OF fails to solve it to optimality but B/B successfully solved it within two seconds for $r = 2, \dots, 7$. It is worth mentioning that both solvers struggled for the challenging instance “cs4” and they did not terminate gracefully under a three-hour limit. Note that running times of both OF and B/B are similar for instance “data”. This is mainly because the graph itself is a block in which B/B reduces to

OF.

Table 5.16: The largest r-hereditary 2-club number and running times in seconds for OF and B/B on DIMACS test-bed; The fastest solver is highlighted in bold font

Graph	BObj	OF	B/B	BObj	OF	B/B	BObj	OF	B/B
		$r=2$			$r=3$			$r=4$	
karate	12	0.10	**0.00**	6	0.09	**0.00**	6	0.14	**0.00**
dolphins	9	0.13	**0.08**	7	0.12	**0.08**	6	0.11	**0.00**
lesmis	18	0.12	**0.00**	14	0.14	**0.00**	13	0.14	**0.00**
polbooks	20	0.40	**0.06**	15	0.50	**0.05**	13	0.41	**0.11**
adjnoun	23	0.42	**0.06**	12	0.55	**0.06**	9	0.51	**0.15**
football	14	0.23	**0.33**	13	**0.24**	0.28	13	**0.23**	0.26
jazz	79	1.37	**0.46**	73	1.55	**0.37**	65	1.57	**0.43**
celegans	104	6.97	**0.03**	54	12.93	**0.03**	30	14.36	**0.02**
email	27	634.18	**35.85**	23	244.70	**5.70**	20	241.15	**0.16**
polblogs	232	530.04	**297.51**	182	2562.68	**519.31**	159	2155.25	**575.71**
netscience	22	55.82	**0.03**	21	66.11	**0.03**	20	57.63	**0.03**
add20	91	200.97	**0.15**	77	268.27	**0.19**	75	252.69	**0.17**
data	14	4123.41	**2872.56**	12	4838.10	**3215.50**	9	**4213.76**	4810.52
uk	4	2529.21	**0.56**	3	**1350.92**	1546.36	3	**1266.59**	1534.58
power	9	2291.90	**0.30**	7	2981.30	**0.28**	6	2790.08	**0.28**
add32	12	4050.80	**0.28**	5	4551.47	**0.34**	4	5267.46	**63.87**
hep-th	33	11483.70	**0.75**	24	8854.91	**0.63**	24	7797.32	**0.71**
whitaker3	≥6	14749\83%†	14531\50%†	≥3	14752\200%†	14536\333%†	≥3	14751\100%†	14517\433%†
crack	≥6	15716\183%†	16020\183%†	≥5	15723\240%†	15576\240%†	≥3	15724\467%†	15535\467%†
PGP	96	15548\533900%†	**1.53**	71	15571\533900%†	**1.18**	64	15558\533900%†	**1.79**
cs4	–	–	–	–	–	–	–	–	–
		$r=5$			$r=6$			$r=7$	
karate	5	**0.08**	**0.08**	5	0.09	**0.08**	5	**0.08**	**0.08**
dolphins	5	0.11	**0.07**	5	0.11	**0.07**	5	0.10	**0.07**
lesmis	13	0.14	**0.00**	13	0.15	**0.00**	12	0.15	**0.00**
polbooks	11	0.46	**0.04**	9	**0.47**	0.52	6	1.07	**0.68**
adjnoun	5	**1.37**	1.50	5	0.85	**0.61**	5	0.66	**0.45**
football	12	0.24	**0.22**	12	0.23	**0.21**	11	0.24	**0.22**
jazz	60	1.67	**0.32**	51	1.65	**0.30**	50	1.86	**0.27**
celegans	22	27.48	**0.07**	16	28.04	**0.09**	13	41.92	**0.20**
email	16	204.02	**0.12**	14	183.01	**0.13**	13	159.05	**0.05**
polblogs	147	1829.97	**529.45**	134	1782.67	**432.24**	124	1859.97	**303.43**
netscience	20	60.21	**0.03**	20	60.14	**0.03**	20	60.13	**0.03**
add20	70	220.20	**0.17**	69	225.20	**0.14**	69	228.51	**0.12**
data	8	**3920.18**	4165.90	8	**4006.78**	4426.83	6	**3747.69**	4224.22
uk	3	**1254.94**	1606.79	3	**1262.18**	1539.76	3	**1268.41**	1539.61
power	6	2901.97	**0.23**	6	2900.70	**639.61**	6	2895.95	**638.27**
add32	4	5175.02	**62.52**	4	5182.74	**62.68**	4	5488.32	**62.24**
hep-th	24	6836.70	**0.61**	24	7632.32	**0.60**	24	8627.61	**0.60**
whitaker3	≥3	14752\100%†	14959\433%†	≥3	14755\100%†	15023\433%†	≥3	14752\100%†	14609\433%†
crack	≥3	15719\467%†	15542\467%†	≥3	15725\467%†	15555\467%†	≥3	15724\467%†	14609\467%†
PGP	57	15556\533900%†	**1.69**	53	15559\533900%†	**1.45**	47	15560\533900%†	**1.49**
cs4	–	–	–	–	–	–	–	–	–

– The approach did not terminate gracefully, typically due to a memory-related crash

† indicates that the approach did not find an optimal solution under the 3-hour time limit and optimality gap is reported

5.4.5 Results for the maximum biconnected 2-club problem

Yezerska et al. [2017] presented a BC algorithm for solving the maximum biconnected 2-club problem, which is referred to as YBC. We apply the YBC along with B/B decomposition Algorithm 10, and call it as B/BYBC. To assess the performance of the B/B decomposition approach, we re-run the YBC algorithm in [Yezerska et al., 2017] under our experimental setting described in Section 5.4.2. Here we impose a one-hour solve-time limit same as in [Yezerska et al., 2017]. The comparison of the performance between B/BYBC and YBC is reported in Table 5.17 including the known best objective, optimality gap and Gurobi solve-time for the DIMACS instances.

Table 5.17: Comparison of running times in seconds and optimality gap between B/BYBC and YBC

			B/BYBC			YBC		
Graph	$\lvert V\rvert$	$\lvert E\rvert$	grbSolTime	Size	Gap(%)	grbSolTime	Size	Gap(%)
karate	34	78	0.03	17	0	**0.02**	17	0
dolphins	62	159	0.13	12	0	**0.07**	12	0
polbooks	105	441	0.13	28	0	**0.12**	28	0
adjnoun	112	425	**0.11**	48	0	0.19	48	0
football	115	613	2.79	16	0	**2.73**	16	0
jazz	198	2742	**0.36**	103	0	0.37	103	0
celegans	453	2025	**2.84**	222	0	8.49	222	0
email	1133	5451	**22.73**	69	0	31.75	69	0
polblogs	1490	16715	165.36	346	0	**125.48**	346	0
netscience	1589	2742	**0.17**	25	0	79.24	25	0
add20	2395	7462	**46.36**	124	0	69.15	124	0
data	2851	15093	3600.15	17	11.76	3600.12	17	11.76
uk	4824	6837	**1486.55**	5	0	2210.64	5	0
power	4941	6594	**779.17**	14	0	2120.16	14	0
add32	4960	9462	**16.52**	30	0	3600.42	12	250
hep-th	8361	15751	**2078.77**	45	0	3601.20	2	417950
whitaker3	9800	28989	3601.79	6	200	3601.79	6	163233
crack	10240	30380	3601.75	6	170567	3605.13	6	170567
PGP	10680	24316	**264.36**	196	0	3602.01	56	267.86
cs4	22499	43858	–	–	–	–	–	–

– The approach did not terminate gracefully, typically due to a memory-related crash

Note that the times reported in this table only includes Gurobi solve-time in order to keep consistent with the original results in [Yezerska et al., 2017]. It is observed that B/BYBC outperforms YBC algorithm for nearly all instances except five instances whose running times are under 130 seconds. Interestingly, B/BYBC is able to solve three instances "add32", "hep-th" and "PGP" to optimality that the YBC algorithm cannot solve under a one-hour time limit. Both fail to solve three challenging instances "data", "whitaker3" and "crack" to optimality, although gaps by B/BYBC are better than YBC.

5.4.6 Summary

Our proposed strengthened formulation, preprocessing techniques and B/B decomposition algorithm are very effective to solve the maximum r-robust 2 club problem on large-scale real-life instances and randomly generated graphs. It is worth mentioning that block decomposition of a graph might be itself when the graph is dense enough. In this case, B/B decomposition algorithm degenerates and does not help reduce running times. Adding constraints on-the-fly is typically faster than directly solving the formulation even though the maximum r-robust 2 club problem formulation has $O(|V^2|)$ constraints. But we also notice that BCSF is worse than SF for a few URG instances, in which case a lot of nodes were explored and a large number of lazy cuts were added.

Our B/B decomposition algorithm can be extended to the maximum r-hereditary 2-club and biconnected 2-club problems. We compare B/B decomposition algorithm against other approaches, and the computational results show that B/B decomposition algorithm is very effective to solve large-scale real-life instances. Our effective algorithms are based on IP approaches, and we implement our approaches on a commercial optimization solver. Readers who are interested in other solvers may refer to the paper by Komusiewicz et al. [2019] who presented an effective combinatorial algorithm using data reduction rules for the maximum r-robust 2-club, r-hereditary 2-club, and r-connected 2-club problems.

CHAPTER VI

Second Order s-Clubs

In this chapter, a cut-like IP formulation for the r, s-MRCP when $s \in \{2, 3, 4\}$ is presented. We devise a BC algorithm based on a delayed constraint generation scheme for the r, s-MRCP. Moreover, a hybrid block-by-block decomposition approach is proposed for solving such problems. This decomposition approach is also extended to solving the r, s-MHCP. The benefits of the algorithmic ideas are empirically evaluated through our computational studies.

6.1 Maximum r-robust s-club problem

6.1.1 Cut-like IP formulation

Recall that $\rho_s(G; u, v)$ denotes the maximum number of vertex-disjoint u, v-paths of length at most s in G for a pair of vertices $u, v \in V$. For a pair of non-adjacent vertices u, v, $\kappa_s(G; u, v)$ denotes the minimum number of vertices in $V \setminus \{u, v\}$ whose deletion disconnects all u, v-paths of length at most s in G. A length-s u, v-separator for every pair of *non-adjacent* vertices u and v is described in Definition 6. However, vertex separators do not exist for *adjacent* vertices u and v, and thus we extend the definition as follows.

Definition 7. *Given a graph $G = (V, E)$ and a positive integer $s \geq 1$, a subset $C \subseteq V \setminus \{u, v\}$ of vertices is called an extended length-s u, v-separator in graph $G = (V, E)$ if $d_{G-C-uv}(u, v) > s$.*

Note that in Definition 7, if the edge $uv \notin E$, then ambiguous $d_{G-C-uv} = d_{G-C}$. By Proposition 1, $\rho_s(G; u, v) = \kappa_s(G; u, v)$ when $s \in \{2, 3, 4\}$ for every pair of *non-adjacent* vertices u and v. Based on this result and similar formulations by Buchanan and Salemi [2017], we present a cut-like formulation for the r, s-MRCP when $s \in \{2, 3, 4\}$ next.

$$\max \sum_{i \in V} x_i \tag{6.1}$$

$$r(x_u + x_v - 1) \leq \mathbb{1}_E(u, v) + \sum_{i \in C} x_i \qquad \forall (u, v, C) \tag{6.2}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{6.3}$$

In formulation (6.1)–(6.3), $\forall (u, v, C)$ is a short-hand for every pair of vertices u, v and all extended length-s u, v-separators C. We prove the correctness of this cut-like formulation in the following theorem.

Theorem 5. *The cut-like formulation* (6.1)-(6.3) *is correct for the r, s-MRCP when $s \in \{2, 3, 4\}$.*

Proof. We shall show that $S \subseteq V$ is an r-robust s-club if and only if its characteristic vector $x^S \in \{0, 1\}^n$ satisfies all constraints of (6.2). We prove the contrapositive for both directions.

(*Sufficiency* $\Leftarrow$). Suppose S is not an r-robust s-club, it implies that there exist two vertices $u, v \in S$ such that $\rho_s(G[S]; u, v) \leq r - 1$. Let us consider two cases.

Case (i) Suppose $\{u, v\} \notin E$. Then, it follows from Proposition 1 that $\kappa_s(G[S]; u, v) \leq r - 1$. Hence, there exists a length-s u, v-separator $C' \subseteq S$ in subgraph $G[S]$ such that $|C'| \leq r - 1$. Clearly, $C := C' \cup (V \setminus S)$ is an extended length-s u, v-separator in graph G. It follows that x^S violates the extended length-s u, v-separator C inequality, since $r(x_u^S + x_v^S - 1) = r$ and $\mathbb{1}_E(u, v) + \sum_{i \in C} x_i^S = 0 + |S \cap C| = |C'| \leq r - 1$.

Case (ii) Suppose $\{u, v\} \in E$. Then, it follows that $\rho_s(G[S] - uv; u, v) \leq r - 2$. By Proposition 1, $\kappa_s(G[S] - uv; u, v) \leq r - 2$. Hence, there exists a length-s u, v-separator $C' \subseteq S$ in subgraph $G[S] - uv$ such that $|C'| \leq r - 2$. As before, $C := C' \cup (V \setminus S)$ is an extended length-

s u,v-separator in graph G. It follows that x^S violates the extended length-s u,v-separator C inequality, since $r(x_u^S + x_v^S - 1) = r$ and $\mathbb{1}_E(u,v) + \sum_{i \in C} x_i^S = 1 + |S \cap C| = 1 + |C'| \leq r - 1$.

(*Necessity* $\Rightarrow$). Let $S \subseteq V$ and suppose $r(x_u^S + x_v^S - 1) > \mathbb{1}_E(u,v) + \sum_{i \in C} x_i^S$ for some extended length-s u,v-separator C in G. It implies that $u, v \in S$ and thus $\mathbb{1}_E(u,v) + \sum_{i \in C} x_i^S \leq r - 1$.

Case (i) Suppose $\{u,v\} \notin E$, it follows that $|S \cap C| \leq r - 1$. The set $C' = S \cap C$ is a length-s u,v-separator in $G[S]$. If not, $G[S] \setminus C'$ will contain a u,v-path of length at most s, which cannot be disconnected by deleting any vertex in $C \setminus S$; a contradiction to the assumption that C is an extended length-s u,v-separator in G. Hence, $\kappa_s(G[S]; u, v) \leq |C'| \leq r - 1$. By Proposition 1, $\rho_s(G[S]; u, v) \leq r - 1$, and S is not an r-robust s-club.

Case (ii) Suppose $\{u,v\} \in E$, it follows that $|S \cap C| \leq r - 2$. By a similar argument, the set $C' = S \cap C$ is a length-s u,v-separator in $G[S] - uv$. Hence, $\kappa_s(G[S] - uv; u, v) \leq |C'| \leq r - 2$. By Proposition 1, $\rho_s(G[S] - uv; u, v) \leq r - 2$, and S is not an r-robust s-club. $\square$

The proof of our cut-like formulation of r,s-MRCP only works when $s = 2, 3,$ and 4. This is because when $s \geq 5$, the length-bounded counterpart of Menger's theorem no longer holds [Lovász et al., 1978]. Hence, the sufficient condition of Theorem 5 does not hold for $s \geq 5$. Interestingly, the necessary condition in Theorem 5 still holds, and therefore the cut-like formulation (6.1)-(6.3) can serve as a relaxation of the r,s-MRCP formulation for $s \geq 5$.

6.1.2 Branch-and-Cut algorithm

In this section, we will present a BC algorithm to solve the r,s-MRCP for $s = 2, 3$ and 4. Notice that it is sufficient to only consider inclusion-wise minimal extended length-s u,v-separator C for the cut-like formulation (6.1)-(6.3). When $s = 2$, minimal extended length-s u,v-separator C coincides with common neighbors of u and v. Hence, when $s = 2$, the cut-like formulation (6.1)-(6.3) is compact and reduces to the original formulation (2.27)–(2.29) of the maximum r-robust 2-club problem presented by Veremyev and Boginski [2012]. However, for $s = 3$ and 4, the cut-like formulation has exponentially many constraints in the worst case, and hence we devise a BC algorithm based on a delayed constraint generation scheme. To serve as the master relaxation at the root node of the branch-and-cut tree, we employ the following s-clique formulation.

$$\max \sum_{i \in V} x_i \tag{6.4}$$

$$\text{s.t. } x_u + x_v \leq 1 \qquad \forall \{u, v\} \in \binom{V}{2} : d_G(u, v) \geq s + 1 \tag{6.5}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{6.6}$$

However, when we implemented this, we found out that the master relaxation problem is too weak and the computational effort required by separation procedure is too expensive. Therefore, we have to impose additional constraints to strengthen the master relaxation. For each pair of vertices $u, v \in V$, at most one of them can be selected in an r-robust s-club if $\rho_s(G; u, v) \leq r - 1$. So, we introduce the following notion of an r-robust s-clique.

Definition 8. *Given a graph $G = (V, E)$ and positive integers r and s, a subset $S \subseteq V$ is called an r-robust s-clique if for every pair of vertices $u, v \in S$, $\rho_s(G; u, v) \geq r$.*

Therefore, instead of employing an s-clique formulation, we utilize an r-robust s-clique formulation which yields a strengthened master relaxation (6.7)-(6.9).

$$\max \sum_{i \in V} x_i \tag{6.7}$$

$$\text{s.t. } x_u + x_v \leq 1 \qquad \forall \{u, v\} \in \binom{V}{2} : \rho_s(G; u, v) \leq r - 1 \tag{6.8}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{6.9}$$

Based on this strengthened master relaxation, we develop a BC algorithm for solving the r, s-MRCP when $s \in \{2, 3, 4\}$. The basic procedure is as follows. When we solve the master relaxation (6.7)-(6.9) using a branch-and-bound (BB) approach and obtain an integral optimal solution $x^* \in \{0, 1\}^n$ at some node of the BB tree, we need to check if selected vertices $S \coloneqq \{i \in V : x_i^* = 1\}$ form an r-robust s-club. Specifically, for each pair of vertices $u, v \in S$, we have to check if $\rho_s(G[S]; u, v) \geq r$. If not, we add a (lazy) cut—a violated minimal extended length-s u, v-separator inequality.

When $s = 2$, verifying if $\rho_s(G[S]; u, v) \geq r$ reduces to checking if there are at least r common neighbors of u and v in subgraph $G[S]$. However, for $s \in \{3, 4\}$, it is not as straightforward to check. Itai et al. [1982] showed that for $s \in \{3, 4\}$, $\rho_s(G; u, v)$ can be computed in $O(|E|\sqrt{|V|})$ time for every pair of vertices u and v by employing Dinic's algorithm for the maximum flow problem when vertices and edges have unit capacity in the network [Dinic, 1970]. We make edges and vertices unit capacity in order to run Dinic's algorithm in $O(|E|\sqrt{|V|})$. The following is the textbook "vertex splitting" procedure for transforming a given undirected graph $G = (V, E)$ and $a, b \in V$ into a maximum flow network $G' = (V', E')$ to compute $\rho_s(G; a, b)$.

a) Designate a as source vertex and b as sink vertex.

b) For each vertex $v \in V \setminus \{a, b\}$, add two vertices v', v'' to V', and directed edges (v', v'') and (v'', v') to E'; Assign the capacity of both edges (v', v'') and (v'', v') to 1.

c) For each edge $(a, v) \in E$, add directed edge (a, v') to E'; for edge $(v, b) \in E$, add directed edge (v'', b) to E'; for each undirected edge $\{u, v\} \in E$, add (u'', v') and (v'', u') to E'. Assign each edge a capacity of 1.

Therefore, we can check if $\rho_s(G; u, v) \geq r$ in $O(|E|\sqrt{|V|})$ time for every pair of vertices u and v using algorithms presented by Itai et al. [1982] when $s = 3$ and 4. The worst case complexity to check if S is an r-robust s-club is $O(|E||V|^{\frac{5}{2}})$. In fact, it is not strictly necessary to always compute $\rho_s(G[S]; u, v)$ to verify if S is an r-robust s-club. We can terminate early when flow value meets or exceeds r. Our computational study shows that this early termination significantly speeds up separation as we would expect.

Next, we will discuss how to identify a minimal extended length-s u, v-separator to generate violated inequalities. The set $C' := \bar{C} \cup (V \setminus S)$ is an extended length-s u, v-separator in G, where $\bar{C}$ is the minimum cardinality extended length-s u, v-separator in subgraph $G[S]$, obtained as the minimum cut from Dinic's algorithm when $\rho_s(G[S]; u, v) \leq r - 1$.

Suppose we are running a Dinic's algorithm in a unit capacity directed graph $G' = (V', E')$ with source a and sink b constructed from $G[S]$. To obtain a minimum cut, we modify the final residual network without affecting the maximum flow by setting the capacity of any edge $(u, v) \in E'$ to ∞ except the edges $(v', v'') \in E'$ for each $v \in V \setminus \{a, b\}$. Then, the set of vertices that are reachable from the source in the residual network induces a minimum source-sink cut in G'. We can convert these edges in the minimum cut, which can only be the vertex split edges, into a minimum vertex separator $\bar{C}$ of $G[S]$.

Next, we modify the algorithm for obtaining a minimal length-s, u, v- separator in [Buchanan and Salemi, 2017] to Algorithm 11 to make C' a *minimal* extended length-s u, v-separator in G. Buchanan and Salemi [2017] also noted that MINIMALIZE can be improved in practice as follows: (1) For each vertex $c \in C'$ such that $d_G(u, c) + d_G(v, c) \geq s + 1$, we can first remove it before *initialize*; (2) Each vertex $c \in N(u) \cap N(v)$ belongs to the extended length-s u, v-separator C; i.e., skip steps 9-12 (for-loop) in Algorithm 11.

6.1.3 Hybrid B/B decomposition algorithm

Recall that by Lemma 4 every r-robust s-club ($r \geq 2$) must be contained in a block. Ideally, like the maximum r-robust 2-club problem, we can first apply RECURSIVE BLOCK algorithm and then utilize B/B decomposition for the r, s-MRCP when $r \geq 2, s \in \{3, 4\}$. However, unlike the maximum r-robust 2-club problem, in practice, edge peeling is very expensive to

Algorithm 11 Minimal extended length-s u, v-separator

procedure MINIMALIZE(G, u, v, C')
Input: Graph $G = (V, E), u, v \in V$, and an extended length-s u, v-separator C'
Output: A minimal extended length-s u, v-separator C
    Initialize: $C \leftarrow C'$
    if $\{u, v\} \in E$ **then**
        $G \leftarrow G - uv$
    end if
    for each $c \in C$ **do**
        $G_c \leftarrow G \setminus (C \setminus \{c\})$
        Compute $d_{G_c}(u, c)$ and $d_{G_c}(v, c)$
        if $d_{G_c}(u, c) + d_{G_c}(v, c) \geq s + 1$ **then**
            $C \leftarrow C \setminus \{c\}$
        end if
    end for
    return C
end procedure

employ for the maximum r-robust s-club problem when $s = 3$ and 4 because we have to compute length-bounded vertex-disjoint paths using Dinic's algorithm. Hence, we propose the following compromise that permits two rounds of block decomposition at most.

First, we identify all candidate blocks of graph G, denoted by $\langle B_1, \ldots, B_k \rangle$. We can continue to preprocess each B_i by vertex and edge peeling, as a result B_i may further decompose into smaller blocks. Denote by $\mathcal{C}_i$ the collection of candidate blocks obtained in this manner from B_i. We associate a binary variable y_B for each block $B \in \mathcal{C}_i$ and develop an extended formulation for the r, s-MRCP master relaxation (6.7)–(6.9) for block B_i.

$$\max \sum_{p \in V(B_i)} x_p \tag{6.10}$$

$$\text{s.t.} \sum_{B \in \mathcal{C}_i} y_B \leq 1 \tag{6.11}$$

$$x_p \leq \sum_{B \in \mathcal{C}_i : p \in V(B)} y_B \qquad \forall p \in V(B_i) \tag{6.12}$$

$$x_u + x_v \leq 1 \qquad \forall \{u, v\} \in \binom{V(B)}{2} : \rho_s(B_i; u, v) \leq r - 1, B \in \mathcal{C}_i \tag{6.13}$$

$$x_p \in \{0,1\} \qquad \forall p \in V(B_i) \tag{6.14}$$

$$y_B \in \{0,1\} \qquad \forall B \in \mathcal{C}_i \tag{6.15}$$

To find a maximum r-robust s club in block B_i, we can utilize the BC algorithm discussed in Section 6.1.2 by employing this extended formulation (6.10)-(6.15) as master relaxation. This so-called hybrid B/B decomposition approach is presented in Algorithm 12.

Algorithm 12 Hybrid B/B Decomposition Algorithm for the r, s-MRCP

```
procedure HYBRID B/B-MRC(G, r, s, ℓ)
Input: Graph G = (V, E), lower bound ℓ, and positive integers r ≥ 2, s ∈ {3, 4}
Output: A maximum r-robust s-club S
Initialize: S ← ∅, T ← ∅, 𝓑 ← ∅, i ← 1
    G ← r-CORE(G)
    𝓑 ← FIND BLOCK(G)
    Let 𝓑 = ⟨B_1, ..., B_k⟩ in non-increasing order of sizes
    while |B_i| > |S| do
        B_i ← EDGE PEELING (B_i, r, s)
        B_i ← VERTEX PEELING(B_i, r, s, ℓ)
        𝓒_i ← FIND BLOCK(B_i)
        W ← Find a maximum r-robust s-club in B_i
        S ← arg max{|S|, |W|}
        i ← i + 1
    end while
    return S
end procedure
```

6.2 Extension to r-hereditary s-club

Recall that Buchanan and Salemi [2017] suggested a cut-like formulation (2.43)–(2.45) for the r, s-MHCP, but no computational results were reported. In this section, we will extend our computational approaches for the r, s-MRCP to r, s-MHCP including branch-and-cut and B/B decomposition for $s = 2, 3$ and 4.

6.2.1 Branch-and-Cut algorithm

Like the r, s-MRCP, the cut-like r, s-MHCP formulation has exponentially many constraints. Accordingly, we devise a BC algorithm based on a delayed constraint generation scheme. Because length-bounded Menger's theorem holds when $s = 2, 3$ and 4 [Lovász et al., 1978], verifying an r-hereditary s-club is equivalent to checking if $\rho_s(G[S]; u, v) \geq r$ for every pair of *non-adjacent* vertices u and v in subgraph $G[S]$ when $s = 2, 3$ and 4. Therefore, we can slightly modify the strengthened master relaxation for the r, s-MRCP (6.7)-(6.9), and obtain a strengthened master relaxation for the r, s-MHCP when $s \in \{2, 3, 4\}$, as presented below.

$$\sum_{i \in V} x_i \tag{6.16}$$

$$\text{s.t. } x_u + x_v \leq 1 \qquad \forall \{u, v\} \in \binom{V}{2} \setminus E : \rho_s(G; u, v) \leq r - 1 \tag{6.17}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{6.18}$$

Based on this strengthened master relaxation, we devise a BC algorithm for solving the r, s-MHCP. The basic procedure is as follows. When we solve the strengthened master relaxation problem (6.16)-(6.18) and obtain an integral solution $x^* \in \{0, 1\}^n$ at some node of the BC tree, we need to check if selected vertices $S \coloneqq \{i \in V : x_i^* = 1\}$ form an r-hereditary s-club. Specifically, we have to check if $\rho_s(G[S]; u, v) \geq r$ for every pair of non-adjacent vertices u and v in subgraph $G[S]$ when $s = 2, 3$ and 4. If $\rho_s(G[S]; u, v) \leq r - 1$ for some pair of non-adjacent vertices u and v, we must add a (lazy) cut—a violated minimal length-s, u, v-separator inequality $r(x_u + x_v - 1) \leq \sum_{i \in C} x_i$ that eliminates this solution without cutting off any incidence vectors of r-hereditary s-clubs.

6.2.2 Hybrid B/B decomposition algorithm

As before, we can combine B/B decomposition algorithm and extended IP formulation for the r, s-MHCP master relatxation. This approach is referred to as hybrid B/B decomposition algorithm for the r, s-MHCP. First, we identify all candidate blocks of graph G, denoted by $\langle B_1, \ldots, B_k \rangle$. We can continue to preprocess each B_i by vertex peeling. A block B_i may further decompose into smaller blocks, and let $\mathcal{C}_i$ be the collection of them. Like the extended formulation for the r, s-MRCP master relaxation, we associate a binary variable y_B for each block $B \in \mathcal{C}_i$ and develop the following formulation for the r, s-MHCP master relaxation.

$$\max \sum_{p \in V(B_i)} x_p \tag{6.19}$$

$$\text{s.t.} \sum_{B \in \mathcal{C}_i} y_B \leq 1 \tag{6.20}$$

$$x_p \leq \sum_{B \in \mathcal{C}_i : p \in V(B_i)} y_B \qquad \forall p \in V(B_i) \tag{6.21}$$

$$x_u + x_v \leq 1 \qquad \forall \{u, v\} \in \binom{V(B)}{2} \setminus E(B) : \rho_s(B_i; u, v) \leq r - 1, B \in \mathcal{C}_i \tag{6.22}$$

$$x_p \in \{0, 1\} \qquad \forall p \in V(B_i) \tag{6.23}$$

$$y_B \in \{0, 1\} \qquad \forall B \in \mathcal{C}_i \tag{6.24}$$

To find a maximum r-hereditary s-club in the block B_i when $s \in \{3, 4\}$, we can utilize the BC algorithm discussed in Section 6.2.1 by employing master relaxation (6.19)-(6.24). The detailed hybrid B/B decomposition approach is presented in Algorithm 13.

6.3 Computational experiments

The goal of the computational experiments is to study the effectiveness of the cut-like IP formulation, BC algorithm, preprocessing techniques, and the hybrid B/B decomposition

Algorithm 13 Hybrid B/B Decomposition Algorithm for the r, s-MHCP

procedure Hybrid B/B-MHC(G, r, s, ℓ)
Input: Graph $G = (V, E)$, lower bound ℓ, and positive integers $r \geq 2, s \in \{3, 4\}$
Output: A maximum r-hereditary s-club S
Initialize: $S \leftarrow \emptyset, T \leftarrow \emptyset, \mathcal{B} \leftarrow \emptyset, i \leftarrow 1$
  $G \leftarrow r$-Core(G)
  $\mathcal{B} \leftarrow$ Find Block(G)
  Let $\mathcal{B} = \langle B_1, \ldots, B_k \rangle$ in non-increasing order of sizes
  while $|B_i| > |S|$ **do**
    $B_i \leftarrow$ Vertex Peeling-MHC(B_i, r, s, ℓ)
    $\mathcal{C}_i \leftarrow$ Find Block(B_i)
    $W \leftarrow$ Find a maximum r-hereditary s-club in B_i
    $S \leftarrow \arg\max\{|S|, |W|\}$
    $i \leftarrow i + 1$
  end while
  return S
end procedure

algorithm proposed in Section 6.1 for solving the r, s-MRCP. We also extend these approaches to the r, s-MHCP as in Chapter V. In order to test the performance of these approaches, we select large-scale real-life networks from the Tenth DIMACS Implementation Challenge on Clustering Instances [Bader et al., 2013] as benchmarks. Numerical results are reported and discussed for the r, s-MRCP and r, s-MHCP. Experimental settings including software and hardware are detailed next.

6.3.1 Experimental settings

All approaches conducted in this computational study are implemented in C++, and Gurobi™ Optimizer 8.1.0 [Gurobi Optimization, Inc., 2019] is employed to solve IP formulations. We impose a three-hour time limit on *solve-time* parameter of Gurobi. However, on some instances Gurobi did not strictly enforce the time-limit parameter resulting in significantly longer running times; these must be interpreted as a failure to solve to optimality within the three-hour limit. We use Gurobi's barrier implementation to solve the root node relaxation while choosing the dual simplex solver at all other nodes of the branch-and-bound tree. In

Gurobi, the parameter MIPFocus strikes a balance between finding new feasible solutions and proving that the current solution is optimal. We focus more attention on proving optimality by setting MIPFocus = 2. The parameter NodefileStart allows nodes to be compressed and written to disk when memory is exhausted. We set the parameter NodefileStart to 0.1 since we need to solve some large instances consuming large memory. All other Gurobi parameters are left at their default setting. We conduct all numerical experiments on a 64-bit Linux® compute node with dual Intel® Xeon® E5-2620 hex-core 2.0GHz processors and 32 GB RAM. When we implement the computation of length-bounded vertex-disjoint paths, we do parallel programming by employing "OpenMP" library [Dagum and Menon, 1998].

6.3.2 Results for the r, s-MRCP

We solve the r, s-MRCP for $r \in \{2, 3, 4\}$ and $s \in \{3, 4\}$. The best objectives and running times in seconds including preprocessing, Gurobi solve-time and wall-clock time are reported in Tables 6.1. We observe that our approaches are effective to solve the r, s-MRCP on DIMACS instances. Optimal solutions are found for all instances except "email" ($s = 3, r = 3$) and "hep-th" ($s = 4, r = 2$) within the three-hour Gurobi solve-time limit. Like the $r, 2$-MRCP, we report their optimality gaps calculated by $100*(\text{UB}-\text{LB})/\text{LB}\%$ for sub-optimal termination.

When we conduct this computational study, we exploit several approaches to speed up the algorithm. Consider the case $s = 3$ and $r = 3$. Initially, we employ s-clique formulation (6.4)–(6.6) as the master relaxation, in which case the solver did not terminate gracefully, typically due to a memory-related crash. Then, we strengthen the master relaxation by utilizing r-robust s-clique formulation (6.7)-(6.9). To reduce the time for verifying the feasibility of the integral solution at each BC node, we terminate the computation of $\rho_s(G; u, v)$ early, when it is verified that $\rho_s(G; u, v) \geq r$. These two measures led to a substantial improvement. All instances but "email" are solved to optimality which takes a total of 31 hours 23 minutes of wall-clock time across all instances. Furthermore, after we employ vertex and edge peeling

Table 6.1: Best objectives (Obj) and running times in seconds including preprocessing (PP), Gurobi solve-time (grbSolve) and wall-clock time (Wall) are reported on DIMACS instances for the r, s-MRCP with $r \in \{2, 3, 4\}$ and $s \in \{3, 4\}$

	$r=2$				$r=3$				$r=4$			
Graph	PP	grbSolve	Wall	Obj	PP	grbSolve	Wall	Obj	PP	grbSolve	Wall	Obj
					$s=3$							
karate	0.02	0.07	0.16	21	0.01	0.06	0.12	11	0.00	0.30	0.35	9
dolphins	0.05	0.07	0.13	22	0.04	0.06	0.11	14	0.04	0.02	0.06	7
lesmis	0.07	0.05	0.14	35	0.06	0.02	0.08	25	0.05	0.04	0.10	21
polbooks	0.23	1.14	1.39	39	0.26	0.64	0.93	31	0.25	0.36	0.62	24
adjnoun	0.26	1.07	1.34	63	0.29	0.79	1.09	47	0.25	0.39	0.65	31
football	0.45	2.74	3.21	40	0.55	5.52	6.10	27	0.55	10.76	11.35	17
jazz	1.68	5.46	7.16	158	1.95	9.51	11.48	145	2.02	11.44	13.49	136
celegans	27.30	20.99	48.38	234	27.82	21.43	49.35	141	10.85	9.49	20.42	99
email	363.97	2936.89	3302.00	138	262.59	10800\14%†	11063.70	≥88	183.98	7957.72	8142.34	66
polblogs	1007.76	1607.18	2615.73	672	834.81	3723.11	4558.78	605	678.40	2997.95	3677.16	557
netscience	0.20	0.02	0.23	24	0.15	0.00	0.17	21	0.06	0.00	0.08	20
power	1686.43	0.16	1686.78	17	0.14	0.02	0.27	12	0.10	0.01	0.22	12
hep-th	11444.30	502.21	11948.10	52	8951.85	95.09	9047.98	38	364.27	10.04	374.73	32
PGP	3571.64	227.89	3801.53	239	2108.34	105.78	2215.54	170	979.80	75.78	1056.54	124
					$s=4$							
karate	0.03	0.22	0.52	26	0.02	0.18	0.36	13	0.01	0.03	0.47	6
dolphins	0.08	0.10	0.20	32	0.08	0.36	0.45	22	0.06	0.16	0.22	7
lesmis	0.12	0.31	0.44	44	0.10	0.08	0.18	33	0.09	0.03	0.12	21
polbooks	0.39	0.91	1.31	57	0.40	1.10	1.52	44	0.41	2.01	2.44	33
adjnoun	0.40	1.43	1.85	94	0.41	2.56	2.98	81	0.41	5.29	5.71	67
football	0.69	9.16	9.86	112	0.74	12.61	13.36	98	0.77	321.73	322.53	54
jazz	3.12	19.96	23.09	186	3.55	22.78	26.35	181	3.98	24.16	28.17	174
celegans	34.77	250.27	285.09	374	36.24	90.21	126.50	291	12.24	114.99	127.27	205
email	748.33	6653.78	7402.62	502	599.60	6437.03	7037.13	403	406.47	5123.80	5530.63	337
polblogs	1432.34	9948.22	11381.10	1000	1156.69	8777.53	9934.69	913	1001.99	8025.34	9027.78	852
netscience	0.36	0.12	0.50	27	0.31	0.00	0.33	21	0.09	0.00	0.11	20
power	4431.19	0.81	4432.32	28	0.18	0.21	0.50	11	0.12	0.00	0.22	6
hep-th	190645.00	10800\7%†	201453.00	≥173	50353.00	6266.20	56622.30	108	1166.33	2198.09	3365.71	70
PGP	89015.10	3696.62	92718.60	446	9895.70	822.77	10721.90	306	10858.10	1324.64	12185.00	226

† indicates that the approach did not find an optimal solution under the 3-hour time limit and optimality gap is reported

introduced in Algorithm 3 and 5, and hybrid B/B decomposition Algorithm 12, the total wall-clock time is reduced to 13 hours 40 minutes. More importantly, the Gurobi solve-times for large instances "hep-th" and "PGP" significantly decrease to under 100 seconds. However, preprocessing times are still large for some challenging instances. It is mainly because of the time required to compute length-bounded vertex-disjoint paths by running the modified Dinic's algorithm. In practice, it is not always necessary to run Dinic's algorithm to compute $\rho_s(G; u, v)$ for every pair of vertices u and v. When $N(u) \cap N(v) \geq r$, or u, v are in a heuristic solution of an r-robust s-club, we know $\rho_s(G; u, v) \geq r$. If $d_G(u, v) > s$, then $\rho_s(G; u, v) = 0$. In these cases, there is no need to run Dinic's algorithm to compute $\rho_s(G; u, v)$.

To take advantage of multicore processors, we utilize parallel programming to calculate length-bounded vertex-disjoint paths. Finally, the total wall-clock time decreases to seven hours twenty-nine minutes along using these approaches. Of course, preprocessing times are still very large for "hep-th" and "PGP", and there is room for improvement in the future.

6.3.3 Results for the r, s-MHCP

Table 6.2: Best objectives (Obj) and running times in seconds including preprocessing (PP), Gurobi solve-time (grbSolve) and wall-clock time (Wall) are reported on DIMACS instances for the r, s-MHCP with $r \in \{2, 3, 4\}$ and $s \in \{3, 4\}$

	$r=2$				$r=3$				$r=4$			
Graph	PP	grbSolve	Wall	Obj	PP	grbSolve	Wall	Obj	PP	grbSolve	Wall	Obj
$s=3$												
karate	0.02	0.07	0.13	21	0.01	0.10	0.15	11	0.00	0.01	0.02	9
dolphins	0.04	0.08	0.13	22	0.04	0.04	0.09	17	0.03	0.06	0.10	7
lesmis	0.07	0.05	0.13	35	0.05	0.02	0.08	25	0.04	0.03	0.08	21
polbooks	0.22	1.02	1.26	39	0.26	0.59	0.87	31	0.23	0.27	0.52	24
adjnoun	0.25	0.93	1.20	63	0.25	0.64	0.90	47	0.21	0.95	1.18	32
football	0.43	6.21	6.67	40	0.49	7.07	7.58	27	0.53	5.56	6.13	17
jazz	1.63	5.26	6.91	158	1.90	9.08	11.00	145	2.00	10.96	12.98	136
celegans	29.42	19.53	49.04	234	27.38	17.07	44.54	141	10.62	9.27	19.99	99
email	473.22	4468.92	4943.36	138	250.77	10800\31%†	11051.90	$\geq$81	174.77	10800\9%†	10975.50	$\geq$66
polblogs	1145.92	1567.20	2713.94	672	801.78	3548.41	4350.81	605	673.80	2068.68	2743.29	558
netscience	0.19	0.02	0.23	24	0.19	0.00	0.37	21	0.06	0.00	0.08	20
power	881.87	0.13	882.16	17	0.17	0.02	0.32	12	0.10	0.00	0.20	12
hep-th	12114.20	384.83	12500.50	52	8337.19	143.64	8481.77	38	382.53	16.99	399.98	32
PGP	3818.95	195.71	4016.60	239	2278.07	96.82	2376.10	170	967.73	75.15	1043.80	124
$s=4$												
karate	0.03	0.08	0.16	26	0.01	0.07	0.12	13	0.00	0.03	0.07	6
dolphins	0.08	0.10	0.20	32	0.07	0.44	0.51	22	0.05	0.18	0.25	7
lesmis	0.12	0.29	0.41	44	0.08	0.06	0.15	33	0.08	0.02	0.11	21
polbooks	0.36	0.90	1.27	57	0.40	1.65	2.08	44	0.94	1.85	2.80	33
adjnoun	0.37	1.32	1.70	94	0.37	2.21	2.59	81	0.35	3.64	3.99	67
football	0.64	2.60	3.25	114	0.72	7.79	8.52	98	9.81	419.67	429.51	54
jazz	3.03	18.18	21.23	186	3.44	20.61	24.07	181	21.46	21.46	42.95	174
celegans	33.05	225.19	258.28	374	34.14	216.47	250.65	291	54.59	133.81	188.45	205
email	1339.74	7374.71	8714.91	502	569.34	5279.27	5849.03	403	563.75	4489.41	5053.58	337
polblogs	1357.84	9402.12	10761.40	1000	1085.93	8144.51	9231.01	913	1523.64	7635.63	9159.72	852
netscience	0.48	0.12	0.95	27	0.33	0.01	0.72	21	0.11	0.00	0.39	20
power	10896.80	0.93	10898.10	28	0.23	0.25	0.59	11	0.16	0.01	0.28	6
hep-th	181498	11594\10%†	193099	173	45305.60	7499.71	52808.00	108	1777.28	3838.07	5616.58	70
PGP	81458.8	3935.85	85401.4	446	8677.39	745.13	9425.84	306	5490.71	884.21	6377.10	226

† indicates that the approach did not find an optimal solution under the 3-hour time limit and optimality gap is reported

Similar to the r, s-MRCP, we solve the r, s-MHCP for $r \in \{2, 3, 4\}$ and $s \in \{3, 4\}$. The r-hereditary s-club number and running times in seconds including preprocessing, Gurobi solve-time and wall-clock time are reported in Tables 6.2. We observe that our approaches are effective for solving these large-scale instances. Within the three-hour Gurobi solve-time limit, all instances are solved to the optimality except instances "email" ($s = 3, r = 3, 4$), and "hep-th" ($s = 4, r = 2$). Similar to results of r, s-MRCP, strengthened master relaxation, preprocessing techniques including vertex and edge peeling, and hybrid B/B decomposition approach significantly improve the algorithm to solve the r, s-MHCP.

However, we also observe that preprocessing times are very large in some cases. It is mainly because of the time required to compute $\rho_s(G; u, v)$ for non-adjacent vertices u and v by running Dinic's algorithm. This phenomenon is very similar to what was observed for the r, s-MRCP. Note that we also avoid the computation of $\rho_s(G; u, v)$ for some cases as we did for the r, s-MRCP. Furthermore, the best objectives of r, s-MRCP and r, s-MHCP are the same for most cases. When their optimal solutions are different, the r-hereditary s-club found is larger than r-robust s-club found. In other words, r-hereditary s-clubs model is more relaxed than r-robust s-clubs in practice. This result is consistent with Lemma 1.

CHAPTER VII

Conclusion and Future Work

Second-order s-clubs include fault-tolerant clusters that preserve low diameter when vertices/edges fail. These models can be used in bioinformatics, social networks, and telecommunications when the data underlying the graph is not reliable. In this chapter, we summarize our contributions to develop theoretical and algorithmic results related to finding second-order s-clubs in graphs. Future research directions are also briefly outlined.

7.1 Contributions

In this dissertation, we establish the first formal NP-hardness results for the r, s-MRCP and r, s-MHCP on arbitrary and restricted graph classes for integer constants $r \geq 2$ and $s \geq 2$. Interestingly, it is also NP-hard to test whether a subset is an r-robust s-club when $r \geq 2$ is fixed and s is a part of the input, and so is its counterpart, i.e., while fixing $s \geq 5$ but r is a part of the input. This provides us some insights on the challenge of solving the r, s-MRCP.

A strengthened IP formulation is presented for the maximum r-robust 2-club problem, which significantly decreases the running-time requirements compared with the original formulation by Veremyev and Boginski [2012] based on our computational study. Furthermore, we develop a cut-like formulation for the r, s-MRCP when $s \in \{2, 3, 4\}$, based on length-bounded vertex separators. This is the first IP formulation for the maximum r-robust 4-club problem in the literature. A branch-and-cut algorithm is also devised based on a delayed constraint generation scheme for the r, s-MRCP when $s \in \{2, 3, 4\}$.

Effective preprocessing techniques including vertex and edge peeling are developed for the r, s-MRCP when $s \in \{2, 3, 4\}$, which enables us to recursively delete many vertices and edges in a given graph without affecting the optimal solutions. Furthermore, we propose a B/B decomposition approach for solving such problems. The computational benefits of the algorithmic ideas are empirically evaluated through our computational studies. This is the first reported numerical results for the maximum r-robust s-club problems when $s = 3$ and 4. Our approach permits us to solve problems optimally on very large, sparse real-life networks from the tenth DIMACS Implementation Challenge on Graph Clustering [Bader et al., 2013] and Stanford Network Analysis Platform [Leskovec and Krevl, 2014] test-beds for the r, s-MRCP where $s \in \{2, 3, 4\}$. Interestingly, most of these preprocessing techniques and B/B decomposition approach can be adapted for the maximum r-hereditary s-club and r-connected s-club problems. Besides the algorithmic contributions to the field of combinatorial optimization, the solvers developed may be beneficial to practitioners in bioinformatics and social network analysis.

7.2 Future work

Investigation on the general value of $s \geq 5$ for the r, s-MRCP would enrich the literature. Our cut-like IP Formulation (6.1)–(6.3) can serve as a relaxation of the r, s-MRCP formulation for $r \geq 5$. Similar to [Moradi and Balasundaram, 2018] (see also [Lu et al., 2018]), it is valuable to devise a BC algorithm for $s \geq 5$ based on a delayed constraint generation scheme by adding CHC, if effective approaches can be developed for feasibility testing when $s \geq 5$.

Extending the block-by-block decomposition approach would be beneficial to speed up solvers to find second-order s-clubs. Note that a subgraph is called an r-block if it is a maximal r-connected subgraph [Matula, 1978], and the classical block is simply a 2-block. In fact, we can extend the results of Lemma 4 to r-blocks. That is, there exists an r-block B_i such that $G[S] \subseteq B_i$ for any r-robust s-club and r-hereditary s-club S. Therefore, we can decompose the original graph into smaller r-blocks instead of 2-blocks, and then solve the r, s-MRCP and r, s-MHCP on each r-block. However, the effectiveness of this approach highly depends on how quickly we are able to decompose the graph into candidate r-blocks. Wen et al. [2017] and Li et al. [2017] have studied approaches for finding r-blocks in a graph. Their methods were based on the block separation lemma proposed by Matula [1978]. It would be beneficial to investigate how to find r-blocks quickly, and applying the r-block decomposition algorithm in solving the r, s-MRCP and r, s-MHCP.

Another future direction is to investigate fault-tolerant s-clubs under edge deletion. Developing strong IP formulations, computational complexity results and decomposition algorithms would contribute to broadening the knowledge-base of second-order s-clubs.

BIBLIOGRAPHY

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

M. T. Almeida and F. D. Carvalho. An analytical comparison of the LP relaxations of integer models for the k-club problem. *European Journal of Operational Research*, 232(3):489–498, 2014.

Y. Asahiro, E. Miyano, and K. Samizo. Approximating maximum diameter-bounded subgraphs. In *Latin American Symposium on Theoretical Informatics*, pages 615–626. Springer, 2010.

D. A. Bader, H. Meyerhenke, P. Sanders, and D. Wagner, editors. *Graph partitioning and graph clustering: Tenth* DIMACS *Implementation Challenge Workshop*, volume 588 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Soc., Providence, RI, 2013.

B. Balasundaram and F. M. Pajouh. Graph theoretic clique relaxations and applications. In P. M. Pardalos, D.-Z. Du, and R. Graham, editors, *Handbook of Combinatorial Optimization*, pages 1559–1598. Springer, New York, 2nd edition, 2013. ISBN 978-1-4419-7996-4.

B. Balasundaram, S. Butenko, and S. Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10(1):23–39, 2005.

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.

V. Batagelj and M. Zaversnik. An $\mathcal{O}(m)$ algorithm for cores decomposition of networks. *arXiv preprint cs/0310049*, 2003.

J. Bondy and U. Murty. *Graph Theory*. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 1846289696.

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

J.-M. Bourjolly, G. Laporte, and G. Pesant. An exact algorithm for the maximum k-club problem in an undirected graph. *European Journal of Operational Research*, 138(1):21–28, 2002.

A. Buchanan and H. Salemi. Parsimonious formulations for low-diameter clusters. Optimization Online Eprints, September 2017.

A. Buchanan and H. Validi. The optimal design of low-latency virtual backbones. Optimization Online Eprints, May 2018.

M.-S. Chang, L.-J. Hung, C.-R. Lin, and P.-C. Su. Finding large k-clubs in undirected graphs. *Computing*, 95(9):739–758, 2013.

J. Cohen. Trusses: Cohesive subgraphs for social network analysis. *National Security Agency technical report*, 16:3–1, 2008.

J. Cohen. Graph twiddling in a mapreduce world. *Computing in Science & Engineering*, 11 (4):29, 2009.

D. J. Cook and L. B. Holder. *Mining Graph Data*. John Wiley & Sons, 2006.

L. Dagum and R. Menon. OpenMP: An Industry-Standard API for Shared-Memory Programming. *IEEE Comput. Sci. Eng.*, 5(1):46–55, Jan. 1998. ISSN 1070-9924.

R. Diestel. *Graph Theory*. Springer-Verlag, Berlin, 1997.

E. A. Dinic. Algorithm for solution of a problem of maximum flow in networks with power estimation. In *Soviet Math. Doklady*, volume 11, pages 1277–1280, 1970.

M. Gendreau, P. Soriano, and L. Salvail. Solving the maximum clique problem using a tabu search approach. *Annals of Operations Research*, 41(4):385–403, 1993.

P. A. Golovach and D. M. Thilikos. Paths of bounded length and their cuts: Parameterized complexity and algorithms. *Discrete Optimization*, 8(1):72–86, 2011.

P. A. Golovach, P. Heggernes, D. Kratsch, and A. Rafiey. Finding clubs in graph classes. *Discrete Applied Mathematics*, 174:57–65, 2014.

J. Guo, C. Komusiewicz, R. Niedermeier, and J. Uhlmann. A more relaxed model for graph-based data clustering: s-plex cluster editing. *SIAM Journal on Discrete Mathematics*, 24 (4):1662–1683, 2010.

Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2019. URL `http://www.gurobi.com`.

S. Hartung, C. Komusiewicz, and A. Nichterlein. Parameterized algorithmics and computational experiments for finding 2-clubs. *Journal of Graph Algorithms and Applications*, 19 (1):155–190, 2015.

S. Hartung, C. Komusiewicz, A. Nichterlein, and O. Suchý. On structural parameterizations for the 2-club problem. *Discrete Applied Mathematics*, 185:79–92, 2015.

J. Hopcroft and R. Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.

A. Itai, Y. Perl, and Y. Shiloach. The complexity of finding maximum disjoint paths with length constraints. *Networks*, 12(3):277–286, 1982.

R. Khandekar, S. Rao, and U. Vazirani. Graph partitioning using single commodity flows. *Journal of the ACM (JACM)*, 56(4):19, 2009.

C. Komusiewicz, A. Nichterlein, R. Niedermeier, and M. Picker. Exact algorithms for finding well-connected 2-clubs in sparse real-world graphs: Theory and experiments. *European Journal of Operational Research*, 275(3):846–864, 2019.

J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

C.-L. Li, S. T. McCormick, and D. Simchi-Levi. The complexity of finding two disjoint paths with min-max objective function. *Discrete Applied Mathematics*, 26(1):105–115, 1990.

Y. Li, Y. Zhao, G. Wang, F. Zhu, Y. Wu, and S. Shi. Effective k-vertex connected component detection in large-scale networks. In *International Conference on Database Systems for Advanced Applications*, pages 404–421. Springer, 2017.

L. Lovász, V. Neumann-Lara, and M. Plummer. Mengerian theorems for paths of bounded length. *Periodica Mathematica Hungarica*, 9(4):269–276, 1978.

Y. Lu, E. Moradi, and B. Balasundaram. Correction to: Finding a maximum k-club using the k-clique formulation and canonical hypercube cuts. *Optimization Letters*, 12(8):1959–1969, 2018.

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

D. W. Matula. k-Blocks and ultrablocks in graphs. *Journal of Combinatorial Theory, Series B*, 24(1):1–13, 1978.

D. W. Matula and L. L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM (JACM)*, 30(3):417–427, 1983.

B. McClosky and I. V. Hicks. Combinatorial algorithms for the maximum k-plex problem. *Journal of combinatorial optimization*, 23(1):29–49, 2012.

K. Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10:95–115, 1927.

H. Meyerhenke, B. Monien, and T. Sauerwald. A new diffusion-based multilevel algorithm for computing graph partitions. *Journal of Parallel and Distributed Computing*, 69(9):750–761, 2009.

R. J. Mokken. Cliques, clubs and clans. *Quality and Quantity*, 13(2):161–173, 1979.

E. Moradi and B. Balasundaram. Finding a maximum k-club using the k-clique formulation and canonical hypercube cuts. *Optimization Letters*, 12(8):1947–1957, 2018.

F. M. Pajouh and B. Balasundaram. On inclusionwise maximal and maximum cardinality k-clubs in graphs. *Discrete Optimization*, 9(2):84–97, May 2012.

F. M. Pajouh, Z. Miao, and B. Balasundaram. A branch-and-bound approach for maximum quasi-cliques. *Annals of Operations Research*, 216(1):145–161, 2014.

J. Pattillo, A. Veremyev, S. Butenko, and V. Boginski. On the maximum quasi-clique problem. *Discrete Applied Mathematics*, 161(1-2):244–257, 2013a.

J. Pattillo, N. Youssef, and S. Butenko. On clique relaxation models in network analysis. *European Journal of Operational Research*, 226(1):9–18, 2013b.

M. A. Porter and S. D. Howison. The role of network analysis in industrial and applied mathematics: A physical-applied-mathematics perspective. arXiv preprint arXiv:1703.06843, 2017.

F. Rahimian, A. H. Payberah, S. Girdzijauskas, M. Jelasity, and S. Haridi. A distributed algorithm for large-scale graph partitioning. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 10(2):12, 2015.

D. Rose, R. Tarjan, and G. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976.

R. A. Rossi. Fast triangle core decomposition for mining large graphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 310–322. Springer, 2014.

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

A. Schäfer, C. Komusiewicz, H. Moser, and R. Niedermeier. Parameterized computational complexity of finding small-diameter subgraphs. *Optimization Letters*, 6(5):883–891, 2012.

S. Shahinpour and S. Butenko. Distance-based clique relaxations in networks: s-clique and s-club. In B. I. Goldengorin, V. A. Kalyagin, and P. M. Pardalos, editors, *Models, Algorithms, and Technologies for Network Analysis*, volume 59, pages 149–174. Springer New York, 2013a.

S. Shahinpour and S. Butenko. Algorithms for the maximum k-club problem in graphs. *Journal of Combinatorial Optimization*, 26(3):520–554, 2013b.

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

A. Veremyev, O. A. Prokopyev, and E. L. Pasiliao. Critical nodes for distance-based connectivity and related problems in graphs. *Networks*, 66(3):170–195, 2015.

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.

J. Wang and J. Cheng. Truss decomposition in massive networks. *Proceedings of the VLDB Endowment*, 5(9):812–823, 2012.

S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge University Press, 1994.

D. Wen, L. Qin, X. Lin, Y. Zhang, and L. Chang. Enumerating k-vertex connected components in large graphs. arXiv preprint arXiv:1703.08668, 2017.

H. Whitney. Congruent graphs and the connectivity of graphs. In *Hassler Whitney Collected Papers*, pages 61–79. Springer, 1992.

O. Yezerska, F. M. Pajouh, and S. Butenko. On biconnected and fragile subgraphs of low diameter. *European Journal of Operational Research*, 263(2):390–400, 2017.

Y. Zhang and S. Parthasarathy. Extracting analyzing and visualizing triangle k-core motifs within networks. In *2012 IEEE 28th International Conference on Data Engineering*, pages 1049–1060. IEEE, 2012.

APPENDIX A

The algorithm for finding all blocks described in Section 4.4

Algorithm 14 Block Decomposition

```
procedure FIND BLOCK(G)
Input: Graph G = (V, E)
Output: All the blocks B
Initialize: depth ← 0, block ← ∅, B ← ∅, stack ← ∅, disc ← −1, low ← −1, parent ← −1
    for each v ∈ V do
        if disc[v] == −1 then
            BLOCKDFS(v, disc, low, parent, stack, depth)
            while stack ≠ ∅ do
                push stack.top() to block
                stack.pop()
            end while
            if block ≠ ∅ then
                B ← B ∪ block
                block ← ∅
            end if
        end if
    end for
    return B
end procedure
procedure BLOCKDFS(v, disc, low, parent, stack, depth)
Initialize: children ← 0, disc[v] = low[v] ← depth + 1, depth ← depth + 1
    for each u ∈ N(v) do
        if disc[u] == −1 then
            children ← children + 1
            parent[u] ← v
            push edge (v, u) to stack
            BlockDFS(u, disc, low, parent, stack, depth)
            low[v] ← min(low[v], low[u])
            if (disc[v] == 1 and children >1) or (disc[v] > 1 and low[u] ≥ disc[v]) then
                block ← ∅
                while stack.top() ≠ (u, v) do
                    push stack.top() to block
                    stack.pop()
                end while
                push stack.top() to block
                stack.pop()
                push block to B
                block ← ∅
            end if
        else if parent[v] ≠ u  and  disc[u] < low[v] then
            low[v] ← disc[u]
            push edge (v, u) to stack
        end if
    end for
end procedure
```

APPENDIX B

The r-core peeling described in Section 4.2

Algorithm 15 Core Peeling

procedure r-Core(G)
Input: Graph $G = (V, E)$ and positive integer r
Output: An r-core or null graph
  while $\delta(G) < r$ and $|V(G)| \geq 1$ **do**
    $v \leftarrow \underset{u \in V(G)}{\arg\min} \deg_G(u)$
    $G \leftarrow G - v$
  end while
  return G
end procedure

VITA

Yajun Lu

Candidate for the Degree of

Doctor of Philosophy

Dissertation: FINDING SECOND-ORDER CLUBS

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Zhumadian, Henan, China.

Education:

Completed the requirements for Doctor of Philosophy in Industrial Engineering and Management at Oklahoma State University, Stillwater, Oklahoma in July, 2019

Received the M.S. degree in Industrial Engineering at Huazhong University of Science and Technology, Wuhan, China in March, 2011

Received the B.S. degree in Industrial Engineering at Zhongyuan University of Technology, Zhengzhou, China in July, 2008

Experience:

Employed by Oklahoma State University in the position of Instructor in Stillwater, Oklahoma during the Spring 2019 semester

Employed by Oklahoma State University in the position of Research/Teaching Assistant in Stillwater, Oklahoma from August 2014 to January 2019

Employed by Huawei Technologies Co., Ltd., in the position of Industrial Engineer in Shenzhen, China from March 2011 to July 2014

Professional Memberships:

Operations Research and the Management Sciences (INFORMS)

Institute of Industrial and Systems Engineers (IISE)

Toastmasters International

Alpha Pi Mu