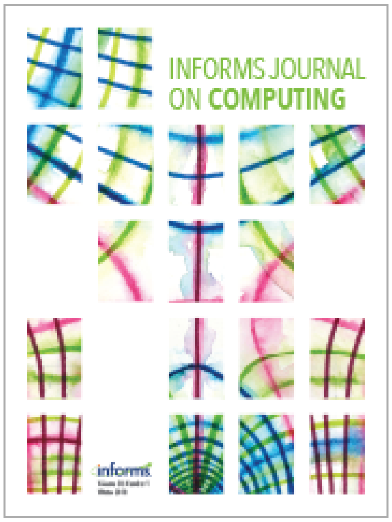


INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

On Fault-Tolerant Low-Diameter Clusters in Graphs

Yajun Lu, Hosseinali Salemi, Balabhaskar Balasundaram, Austin Buchanan

informs
https://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Vol. 34, No. 6, November–December 2022, pp. 3181–3199
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

On Fault-Tolerant Low-Diameter Clusters in Graphs

Yajun Lu,[a] Hosseinali Salemi,[b] Balabhaskar Balasundaram,[c,*] Austin Buchanan[c]

[a] Department of Management & Marketing, Jacksonville State University, Jacksonville, Alabama 36265; [b] Department of Industrial & Manufacturing Systems Engineering, Iowa State University, Ames, Iowa 50011; [c] School of Industrial Engineering & Management, Oklahoma State University, Stillwater, Oklahoma 74078
*Corresponding author

Contact: ylu@jsu.edu, https://orcid.org/0000-0002-6145-7771 (YL); hsalemi@iastate.edu (HS); baski@okstate.edu, https://orcid.org/0000-0002-3490-4257 (BB); buchanan@okstate.edu, https://orcid.org/0000-0003-2999-9666 (AB)

Received: August 18, 2021
Revised: March 31, 2022; July 22, 2022
Accepted: July 26, 2022
Published Online in Articles in Advance: September 14, 2022

https://doi.org/10.1287/ijoc.2022.1231

Abstract. Cliques and their generalizations are frequently used to model "tightly knit" clusters in graphs and identifying such clusters is a popular technique used in graph-based data mining. One such model is the *s*-club, which is a vertex subset that induces a subgraph of diameter at most *s*. This model has found use in a variety of fields because low-diameter clusters have practical significance in many applications. As this property is not hereditary on vertex-induced subgraphs, the diameter of a subgraph could increase upon the removal of some vertices and the subgraph could even become disconnected. For example, star graphs have diameter two but can be disconnected by removing the central vertex. The pursuit of a fault-tolerant extension of the *s*-club model has spawned two variants that we study in this article: robust *s*-clubs and hereditary *s*-clubs. We analyze the complexity of the verification and optimization problems associated with these variants. Then, we propose cut-like integer programming formulations for both variants whenever possible and investigate the separation complexity of the cut-like constraints. We demonstrate through our extensive computational experiments that the algorithmic ideas we introduce enable us to solve the problems to optimality on benchmark instances with several thousand vertices. This work lays the foundations for effective mathematical programming approaches for finding fault-tolerant *s*-clubs in large-scale networks.

History: Accepted by David Alderson, Area Editor for Network Optimization: Algorithms & Applications.
Funding: The computing for this project was performed at the High Performance Computing Center at Oklahoma State University supported in part through the National Science Foundation [Grant OAC-1531128]. This material is based upon work supported by the National Science Foundation under [Grants 1662757 and 1942065].
Supplemental Material: The online appendix is available at https://doi.org/10.1287/ijoc.2022.1231.

Keywords: integer programming • hereditary *s*-clubs • robust *s*-clubs • branch-and-cut • social network analysis

1. 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). An established technique in this setting involves the detection of clusters. This is done by finding a cluster of the largest cardinality or weight, finding clusters that cover or partition the graph, or enumerating all inclusionwise maximal 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 upon which the graph is based, the clique requirement may be too restrictive, resulting in small clusters that miss key members. 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. 2013). Such clique relaxations are less sensitive to edges missed because of erroneous or incomplete data underlying the graph representation. Next, we introduce the notations used and define the clique relaxations of interest.

We consider simple, unweighted graphs in this article. We denote by $G=(V,E)$ an n-vertex graph with vertex set $V=\{1,2,\dots,n\}$ and edge set $E\subseteq\binom{V}{2}:=\{e\subseteq V\,|\,|e|=2\}$ containing m edges. Given a graph $G=(V,E)$, we denote its complement by $\overline{G}=(V,\overline{E})$, where the edge set $\overline{E}:=\binom{V}{2}\setminus E$. By $G-S$, we denote the graph obtained by deleting vertices in $S\subseteq V$ and incident edges from G; for a single vertex v, we use $G-v$. By $G\setminus J$, we denote the graph obtained by deleting edges in $J\subseteq E$; for a single edge e, we use $G\setminus e$. We denote by $G[S]$, the subgraph induced by a subset of vertices S, where $G[S]:=G-(V\setminus S)$. The set of neighbors of a vertex u in graph G is denoted by $N_G(u)$. The *closed* neighborhood of vertex u includes itself and is denoted by $N_G[u]:=N_G(u)\cup\{u\}$. The *distance* between a pair of vertices u and v in G, denoted by $\text{dist}_G(u,v)$, is the minimum number of edges on a path from u to v in G. The *diameter* of G is the maximum distance between any pair of vertices in G and is denoted by diam(G). Given a positive integer s, the distance-s neighborhood of u is denoted by $N_G^s(u)$ and is defined as $N_G^s(u) := \{v \in V \mid 1 \le \text{dist}_G(u,v) \le s\}$. The *closed* distance-$s$ neighborhood of u is denoted by $N_G^s[u] := N_G^s(u) \cup \{u\}$. We use the short form uv for an edge $\{u, v\}$ and drop the subscript G when the graph under consideration is understood without any ambiguity. We recall two distance-based clique relaxations from the literature: s-clique and s-club.

Definition 1 (Luce 1950). Given a positive integer s, a subset of vertices S is called an *s-clique* if $\text{dist}_G(u,v) \le s$ for every pair of vertices $u, v \in S$.

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

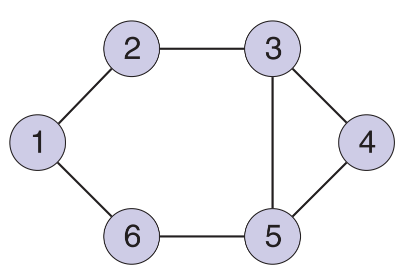
Clearly, the special case $s = 1$ in both definitions corresponds to the clique. The fundamental difference between an s-clique and an s-club is that the distance bound is applicable to the original graph in the former, and to the induced subgraph in the latter. Hence, every s-club is an s-clique, but not vice versa. Figure 1 illustrates this difference (Alba 1973).

Arguably, the s-club model is more cohesive because it guarantees that the length-bounded paths between vertices are completely contained within the induced subgraph. Originally introduced to model cohesive subgroups in social networks (Mokken 1979), s-clubs can be used to model low-diameter clusters for small values of s. In particular, the 2-club represents clusters in which every pair of its 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 group in which members may be directly acquainted or related through a mutual acquaintance in the group.

1.1. Fault-Tolerant Clubs

Although s-clubs can ensure low pairwise distances inside the cluster, they may not be *fault-tolerant* in the sense that deleting a single vertex could increase the distances or even disconnect the graph. For example, in the graph in Figure 1, the set $S_1 = \{2,3,4,5\}$ is a 2-club, but $S_1 \backslash \{3\}$ induces a disconnected subgraph. Yezerska et al. (2017) refer to this as a "fragile" 2-club and focus instead on finding 2-clubs that induce biconnected subgraphs, for example, the 2-club $S_2 = \{1,2,3,5,6\}$ in Figure 1. Nonetheless, S_2 is not fault-tolerant as the diameter of $G[S_2]$ increases if any single vertex in S_2 is deleted.

Figure 1. (Color online) Set $S = \{2,3,4,5,6\}$ Is a 2-Clique That Is Not a 2-Club as the Distance Between Vertices 2 and 6 in the Graph Induced by S Is More Than Two



Note. $\text{dist}_G(2,6) = 2$ using vertex 1 that is not in S.

In practice, even beyond the current setting of low-diameter clusters, fault-tolerance is a desirable attribute as it typically ensures that cluster functionality withstands vertex failures and cluster significance is preserved in the presence of noisy data underlying the graph model. For instance, it is desirable for clusters used in the communication and control of wireless sensor networks deployed during disaster recovery, battlefield situations, or in other adverse terrains to survive sensor failures and remain functional (Gupta and Younis 2003, Chen et al. 2005, Zheng et al. 2008). Network motifs have been used to study biological networks that capture coexpressions of genes or proteins in an organism (Milo et al. 2002, Alon 2007). For instance, Zhang et al. (2005) have identified *network themes*, higher-order interconnected clusters containing recurring elementary motifs, in *Saccharomyces cerevisiae* that are tied to biological phenomena. Interestingly, several of the network themes reported by the authors are also fault-tolerant low-diameter clusters, although those were not what Zhang et al. (2005) sought in their study.

It is important to note here that the s-club property for $s \ge 2$ is not hereditary in the sense of Lewis and Yannakakis (1980). That is, we cannot guarantee that the diameter bound will be preserved under vertex deletion in general, even if the induced subgraph remains connected. This has led researchers to devise notions of "strong attack tolerance," wherein the graph property in question (e.g., diameter) persists under a small number of vertex and/or edge failures. These observations motivated Veremyev and Boginski (2012) to introduce the following fault-tolerant variant of s-clubs.

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

By Definition 3, every r-robust s-club S must contain at least $r + 1$ vertices except when the definition is trivially satisfied, that is, $|S| \le 1$. Furthermore, the only r-robust s-clubs that contain exactly $r + 1$ vertices are cliques of that size. As long as an r-robust s-club contains two vertices that are not adjacent, it must contain at least $r + 2$ vertices. Although this definition bestows an s-club with fault tolerance by ensuring redundant short paths, it is not the only approach to achieve that effect. Consider the following variant introduced by Pattillo et al. (2013).

Definition 4 (Pattillo et al. 2013). Given a graph G and positive integers t and s, a subset of vertices S is called a *t-hereditary s-club* if $S \backslash T$ is an s-club for every deletion set $T \subseteq S$ containing fewer than t vertices.

If we have $t = r = 1$, Definitions 3 and 4 coincide with Definition 2 for every positive integer s, that is, every s-club is both 1-hereditary and 1-robust. Definition 4 deviates slightly from the original definition of Pattillo et al. (2013), which allowed deletion sets up to (and equal to) size t. Our redefinition is more convenient when working with both models simultaneously. Lemma 1 that follows states (without proof) a general relationship between r-robust s-clubs and t-hereditary s-clubs that can be easily verified.

Lemma 1. *Every r-robust s-club is also an r-hereditary s-club.*

The converse of Lemma 1 is not true. For example, a 4-cycle is a 2-hereditary 2-club that is not a 2-robust 2-club. The distance between adjacent vertices in a t-hereditary s-club remains one after any other vertex is deleted and hence, adjacent pairs of vertices are not subjected to any additional requirements. By contrast, an adjacent pair of vertices in an r-robust s-club still need to be connected by at least $r - 1$ additional vertex-disjoint paths of length s or less. This is one key difference that can impact the type of fault-tolerant cluster detected in practice. For example, the largest 3-robust 3-club found in the `dolphins` graph from the DIMACS Clustering Challenge benchmarks (Bader et al. 2013) contains 14 vertices, whereas the largest 3-hereditary 3-club contains 17 vertices.

Although Veremyev and Boginski (2012) and Pattillo et al. (2013) introduced the fault-tolerant s-clubs we study in this article, this notion has been previously studied in extremal graph theory and in the hop-constrained survivable network design literature. Vijayan and Murty (1964) studied extremal (t, s)-accessible graphs—a graph containing the minimum number of edges in which every pair of vertices have distance at most s even after removing any t or fewer vertices (for the case when $s = 2$). Caccetta (1979) considered the more general extremal problem of graphs of diameter s with the minimum number of edges, whose diameter does not increase above a given integer λ upon removing any t or fewer vertices. Although the aforementioned studies are closely related to the notion of t-hereditary s-clubs, Faudree et al. (2012) studied extremal graphs (with minimum number of edges) that contain at least r vertex-disjoint paths of length at most s between every pair of vertices, a model more closely related to r-robust s-clubs. In the survivable network design literature, similar problems have been studied where the goal is to design a graph with minimum total cost of creating edges while requiring that the graph possess fault-tolerance with respect to limited vertex/edge failures (Grötschel et al. 1992; Botton et al. 2013, 2015; Gouveia and Leitner 2017).

1.2. Prior Work and Our Contributions

The focus of this article is on combinatorial optimization problems seeking a maximum cardinality s-club that also satisfies an additional property of robustness or heredity, following Definitions 3 and 4. We refer to these as the maximum r-robust s-club problem (MRCP) and the maximum t-hereditary s-club problem (MHCP). We briefly review the limited literature currently available related to these problems before outlining our contributions.

Komusiewicz et al. (2019) showed that the decision version of the maximum t-hereditary and r-robust 2-club problems are NP-complete for every pair of fixed integers $t, r \geq 2$. The hardness of the problem for *arbitrary s* follows immediately from their results because any algorithm for either problem where s is arbitrary (meaning s is specified in the input) must also solve the problems for $s = 2$. By contrast, the complexity of these problems where s is *fixed* in the problem definition, for example, t-hereditary 3-club is not explicitly addressed by this result. We answer these open complexity questions in this article.

Veremyev and Boginski (2012) proposed a compact integer programming (IP) formulation for a relaxation of the MRCP in which the r paths of length at most s are only required to be distinct and not necessarily vertex-disjoint. However, when $s = 2$, the r distinct paths must also be vertex-disjoint, and therefore their formulation correctly models this special case. Almeida and Carvalho (2014) provided a compact IP formulation for r-robust 3-clubs. No general formulations are currently available for the MRCP when $s \geq 4$.

Salemi and Buchanan (2020) introduced a cut-like formulation for the maximum s-club problem and suggest a modification that formulates the MHCP. This formulation also generalizes the IP formulation of the maximum t-hereditary 2-club problem proposed by Komusiewicz et al. (2019). We prove the correctness of this formulation along similar lines as suggested by Salemi and Buchanan (2020) in this article. Recently, Veremyev et al. (2022) proposed new formulations for the maximum 2-club problem and extended their approaches to find r-robust 2-clubs.

Our contributions are summarized as follows. In Section 2, we establish the NP-completeness of the decision counterparts of the MRCP and the MHCP for all *fixed* integers $s, r, t \geq 2$. We also establish the conditions on the parameters under which the problem of verifying whether a subset of vertices is an r-robust s-club is NP-complete and verifying if it is a t-hereditary s-club is coNP-complete. In Section 3, we present cut-like formulations based on length-bounded vertex separators for the MRCP and the MHCP. Our cut-like formulations are compared with existing formulations in the literature wherever possible. In light of the worst-case exponential size of the cut-like formulations, in Section 4, we establish whether these problems admit "convenient" IP formulations (defined in Section 4) depending on the resolution of $P \stackrel{?}{=} NP$ and the values of r (or t) and s. To speed up solving the MRCP and the MHCP using our cut-like formulations in a branch-and-cut algorithm, we introduce several preprocessing and graph decomposition techniques in Section 5. We report our computational experience solving the MRCP and the MHCP for $s \in \{2, 3, 4\}$ in Section 6 and compare our solver against existing approaches in the literature whenever possible. Our computational study includes the first reported numerical results for the MRCP and the MHCP when $s \in \{3, 4\}$. Our codes are shared publicly on GitHub at https://github.com/yajun668/FaultTolerantClubs. We conclude the paper in Section 7 with a summary and remarks for future research on these and related problems.

2. Problem Complexity

In this section, we establish the intractability of the decision and verification versions of the MRCP and the MHCP, formally stated next.

Problem: *s*-Club/*r*-Robust *s*-Club/*t*-Hereditary *s*-Club (positive integers s, t, r)

Question: Given a graph G and positive integer c, does G contain an s-club/r-robust s-club/t-hereditary s-club of size at least c?

Bourjolly et al. (2002) established that *s*-Club is NP-complete for every fixed integer $s \geq 2$, and it remains NP-complete even when restricted to graphs of diameter $s + 1$ (Balasundaram et al. 2005). Testing inclusionwise maximality of s-clubs is also coNP-complete (Pajouh and Balasundaram 2012). The *s*-Club problem remains NP-hard on 4-chordal graphs for every positive integer s (Golovach et al. 2014). Branch-and-bound algorithms for finding a maximum s-club have also been studied in several articles (Bourjolly et al. 2002, Pajouh and Balasundaram 2012, Chang et al. 2013).

Komusiewicz et al. (2019) showed that the *r*-Robust 2-Club and *t*-Hereditary 2-Club problems are NP-complete for every fixed integer $r \geq 2$ and $t \geq 2$, respectively. Table 1 summarizes the known complexity results related to the *s*-Club problem and its fault-tolerant extensions.

We prove the NP-completeness of the decision counterparts of MRCP and MHCP for every fixed integer $s, r, t \geq 2$ on general graphs and obtain complexity results on some special graph classes as corollaries. We then show that even the verification problems (checking whether a given subset of vertices is an r-robust or t-hereditary s-club) are intractable in certain circumstances where the parameters involved are treated as part of the input. The complexity results pertaining to the verification problems are especially important due to their implications for algorithm development and for the existence of convenient IP formulations (defined in Section 4).

2.1. NP-Hardness of Optimization

The following theorems establish that *r*-Robust *s*-Club and *t*-Hereditary *s*-Club are NP-complete using reductions from *s*-Club. The problems are trivially NP-hard when parameters r, t, s are not fixed in the problem definition, as they all include Clique as a special case where $s = 1$. The proofs of the results are included in Section 1 of the online appendix.

Theorem 1. *r-Robust s-Club is NP-complete for every pair of fixed integers $s \geq 2$ and $r \geq 2$, even on graphs with domination number one.*

Theorem 2. *t-Hereditary s-Club is NP-complete for every pair of fixed integers $s \geq 2$ and $t \geq 2$, even on graphs with domination number one.*

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 nonnegative integer k, a k-chordal graph contains no chordless cycles of length greater than k. Therefore, 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 fixed integer $s \geq 1$, and it remains NP-complete on the subclass of chordal graphs for every fixed *even* integer $s \geq 2$ (Asahiro et al. 2010). Golovach et al. (2014) also proved that 2-Club is NP-hard on graphs with clique cover number three, that is, on graphs whose vertex sets can be covered using three cliques. These results allow us to show the NP-hardness of the MRCP and the MHCP on restricted graph classes as corollaries of Theorems 1 and 2.

Corollary 1. *For every pair of fixed integers $r, t \geq 2$, r-Robust s-Club and t-Hereditary s-Club remain NP-complete,*

1. *On 4-chordal graphs for every fixed integer $s \geq 1$, and*
2. *On chordal graphs for every fixed even integer $s \geq 2$.*

Corollary 2. *For every pair of fixed integers $r, t \geq 2$, r-Robust 2-Club and t-Hereditary 2-Club remain NP-complete on graphs with clique cover number three.*

2.2. Hardness of Verification

We begin this section by recalling the definition of a length-bounded vertex separator (Lovász et al. 1978, Baier et al. 2010, Salemi and Buchanan 2020).

Definition 5. Given a pair of nonadjacent vertices u and v in graph $G = (V, E)$, a subset of vertices $C \subseteq V \backslash \{u, v\}$ is called a length-s u, v-separator if $\text{dist}_{G-C}(u, v) > s$.

Proposition 1, established by Lovász et al. (1978), relates the size of length-bounded vertex separators to the number of length-bounded vertex-disjoint paths. More importantly, this proposition offers a length-bounded counterpart of Menger's theorem for lengths in the set {2, 3, 4} (Menger 1927, Lawler 1976). 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

Table 1. Main Complexity Results Related to s-Clubs and Other Variants

Problem	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-complete on 4-chordal graphs for every positive integer s (Golovach et al. 2014), on bipartite graphs for every fixed $s \geq 3$, and on chordal graphs for every *even* fixed integer $s \geq 2$ (Asahiro et al. 2010). Testing maximality by inclusion is coNP-complete for every fixed integer $s \geq 2$ (Pajouh and Balasundaram 2012). NP-hard to approximate within a factor of $n^{1/2-\epsilon}$ in general graphs for any $\epsilon > 0$ and a fixed $s \geq 2$ (Asahiro et al. 2018). Polynomial-time solvable on the following graph classes: trees, interval graphs, and graphs with bounded treewidth or cliquewidth for every fixed $s \geq 1$ (Schäfer 2009); chordal bipartite, strongly chordal and distance hereditary graphs for every fixed $s \geq 1$; weakly chordal graphs for every fixed *odd* s (Golovach et al. 2014). $O(n^{1/2})$-approximable for fixed $s \geq 2$ (Asahiro et al. 2018). Fixed-parameter tractable when parameterized by solution size (Schäfer et al. 2012).
2-Club	NP-hard on the following graph classes: split graphs (Asahiro et al. 2010); graphs with clique cover number three and diameter three; graphs with domination number two and diameter three (Hartung et al. 2015). Polynomial-time solvable on bipartite graphs in $O(n^5)$ (Schäfer 2009). Approximable by a factor of $n^{1/3}$ on split graphs (Asahiro et al. 2010).
r-Robust and t-Hereditary 2-Club	NP-complete for every pair of fixed positive integers $r, t \geq 1$ (Komusiewicz et al. 2019). Fixed-parameter tractable when parameterized by $\ell = \lvert V \rvert - k$ where k is solution size; does not admit a $(2-\epsilon)^{\ell} n^{O(1)}$-time algorithm for any $\epsilon > 0$ if the Strong Exponential Time Hypothesis is true (Komusiewicz et al. 2019).

of length at most s in G, and for nonadjacent vertices u and v, let $\kappa_s(G; u, v)$ denote the minimum cardinality of a length-s u,v-separator. For convenience, we sometimes refer to these invariants as $\rho_s(\cdot)$ and $\kappa_s(\cdot)$ without specifying the arguments.

Proposition 1 (Lovász et al. 1978). *Consider a graph G with n vertices containing nonadjacent vertices u, v, and a positive integer s. Then,*

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

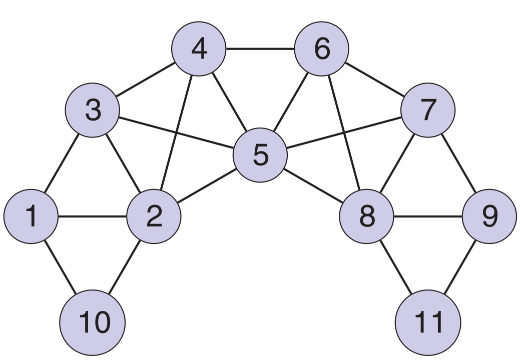
Furthermore, for $s \in \{2, 3, 4\}$,

$$\rho_s(G; u, v) = \kappa_s(G; u, v).$$

Figure 2 provides an example illustrating that $\rho_s(G; u, v)$ could be strictly smaller than $\kappa_s(G; u, v)$ when $s \geq 5$. Note that $\text{dist}_G(10, 11) = 4$. Although several paths of length five and one of length four exist between vertices 10 and 11, no more than one can be included in a vertex-disjoint collection of paths of length at most five. After deleting any single vertex from the set {2,5,8}, we can still find a length-5 path in this graph between vertices 10 and 11. Hence, $\rho_5(G; 10, 11) = 1$, but $\kappa_5(G; 10, 11) = 2$.

When $s = 2$, verifying if a vertex subset S is an r-robust 2-club amounts to checking if every adjacent pair of vertices in S have at least $r - 1$ common neighbors in $G[S]$ and every nonadjacent pair have at least r common

Figure 2. (Color online) In Graph G, $\rho_5(G; 10, 11) = 1$, but $\kappa_5(G; 10, 11) = 2$, Whereas $\rho_4(G; 10, 11) = 1 = \kappa_4(G; 10, 11)$ as {2} Is a Length-4 Separator for Vertices 10 and 11



neighbors in $G[S]$. For distinct vertices u and v that are adjacent in $G[S]$, clearly $\rho_2(G[S];u,v) = |N_G(u) \cap N_G(v) \cap S| + 1$. If they are not adjacent, then $\rho_2(G[S];u,v) = |N_G(u) \cap N_G(v) \cap S|$. Moreover, for nonadjacent vertices u and v, the set of common neighbors $N_G(u) \cap N_G(v) \cap S$ is the unique minimum cardinality length-2 u, v-separator in $G[S]$. Hence, we can verify if S is a t-hereditary 2-club by checking if every nonadjacent pair of vertices have at least t common neighbors in $G[S]$.

We can compute $\rho_s(G;u,v)$ in $O(|E|\sqrt{|V|})$ time for $s \in \{3,4\}$, (Lovász et al. 1978, Itai et al. 1982), which along with Proposition 1 can be used to verify if S is an r-robust or t-hereditary s-club for arbitrary r and t in polynomial time. It is because a subset of vertices S is an r-robust s-club if $\rho_s(G[S];u,v) \geq r$ for every pair of distinct vertices u and v in S, and it is a t-hereditary s-club if $\kappa_s(G[S];u,v) = \rho_s(G[S];u,v) \geq t$ for every pair of nonadjacent vertices u and v in S.

The tractability of the verification problems for $s \geq 5$, wherein the length-bounded counterpart of Menger's theorem does not hold, is addressed in the following discussion. The new complexity results on optimization and verification are summarized in Table 2, and it can be seen from the table that the complexity depends on whether the parameters s, r, and t are fixed or arbitrary.

Problem: Is r-Robust s-Club (positive integers s, r)

Question: Given a graph $G = (V,E)$ and a subset $S \subseteq V$, is S an r-robust s-club in G?

Theorem 3. *r-Robust s-Club is NP-complete for every fixed integer $s \geq 5$ and arbitrary positive integer r.*

Theorem 4. *r-Robust s-Club is NP-complete for every fixed integer $r \geq 2$ and arbitrary positive integer s.*

Theorems 3 and 4 establish that the verification problem is NP-complete if one of the two parameters $s \geq 5$ and $r \geq 2$ is fixed and the other arbitrary. The following theorem states that verification of t-hereditary s-clubs is also difficult when t is arbitrary; however, it is easy when t is a fixed integer. The proofs of these results are included in Section 2 of the online appendix.

Problem: Is t-Hereditary s-Club (positive integers s, t)

Question: Given a graph $G = (V,E)$ and a subset $S \subseteq V$, is S a t-hereditary s-club in G?

Theorem 5. *t-Hereditary s-Club is coNP-complete for every fixed integer $s \geq 5$ and arbitrary positive integer t.*

Remark 1. If instead, parameter t is fixed in the problem and s is specified in the input, verifying whether S is a t-hereditary s-club can be completed in polynomial time by enumerating every deletion set $T \subseteq S$ of size less than t and verifying if $\text{diam}(G[S \setminus T]) \leq s$. If this diameter bound is satisfied for every such deletion set T, then S is a t-hereditary s-club; otherwise, S is not a t-hereditary s-club.

3. Integer Programming Formulations

We introduce *cut-like formulations* for the MRCP and the MHCP and compare their strength with existing formulations in the literature when available. The cut-like formulations that we introduce are based on vertex separators that disconnect all length-bounded paths between a specified pair of vertices. Similar ideas have been used to impose connectivity constraints in other settings (Carvajal et al. 2013, Wang et al. 2017, Salemi and Buchanan 2020). We also consider the complexity of the associated separation problems due to the worst-case exponential size of our formulations.

In the formulations introduced in this section, for a pair of distinct, nonadjacent vertices u and v, we let $\mathcal{C}_{uv}(G)$ denote the collection of all length-s u, v-separators in G. For every pair of vertices $uv \in \binom{V}{2}$, we use $\mathbb{1}_E(u,v)$ as the edge indicator function, that is, $\mathbb{1}_E(u,v) = 1$ if $uv \in E$ and zero otherwise.

3.1. Cut-Like Formulation for the MRCP for $s \in \{2,3,4\}$

The cut-like formulation for the MRCP is proposed next, which we show is correct when $s \in \{2,3,4\}$ in Theorem 6. As alluded to in Section 2.2, this is a consequence of Proposition 1 offering a length-bounded Mengerian theorem only when $s \in \{2,3,4\}$, but not when $s \geq 5$.

In Formulation (1) that follows, binary variable x_i equals one if and only if vertex $i \in V$ is included in the r-robust s-club. If a pair of vertices u, v are included in the subset S, Constraints (1b) ensure that at least $r - \mathbb{1}_E(u,v)$ vertices from every length-s u, v-separator in $G \setminus uv$ must be also chosen. This ensures that the minimum cardinality of a length-s u, v-separator in $G[S] \setminus uv$, which by Proposition 1 equals the maximum number of vertex-disjoint paths of length at most s between u and v in $G[S] \setminus uv$, is at least $r - \mathbb{1}_E(u,v)$.

$$\max \sum_{i \in V} x_i \tag{1a}$$

$$\text{s.t.}\ (r - \mathbb{1}_E(u,v))(x_u + x_v - 1) \leq \sum_{i \in C} x_i \quad \forall\, C \in \mathcal{C}_{uv}(G \setminus uv),\ \forall uv \in \binom{V}{2}, \tag{1b}$$

$$x_i \in \{0,1\} \quad \forall\, i \in V. \tag{1c}$$

Theorem 6. *Given a graph $G = (V,E)$ and parameter $s \in \{2,3,4\}$, a subset of vertices S is an r-robust s-club if and only if its characteristic vector x^S satisfies the constraints of Formulation (1).*

Table 2. Summary of Complexity Results Established in Section 2

Problem	Parameter(s) fixed in the problem	Parameter specified in the input	Complexity
r-Robust s-Club	$s \geq 2$ and $r \geq 2$		NP-complete
t-Hereditary s-Club	$s \geq 2$ and $t \geq 2$		NP-complete
Is r-Robust s-Club	$s \geq 5$	r	NP-complete
	$s \leq 4$	r	Polynomial-time
	$r \geq 2$	s	NP-complete
Is t-Hereditary s-Club	$s \geq 5$	t	CoNP-complete
	$s \leq 4$	t	Polynomial-time
	$t \geq 2$	s	Polynomial-time

Proof (⇒). Let $S \subseteq V$ and suppose $(r - \mathbb{1}_E(u,v))(x_u^S + x_v^S - 1) > \sum_{i\in C} x_i^S$ for some $C \in \mathcal{C}_{uv}(G\backslash uv)$. It implies that $u, v \in S$ and thus $\sum_{i\in C} x_i^S < r - \mathbb{1}_E(u,v)$. Let $C' = S \cap C$, then $|C'| \leq r - 1 - \mathbb{1}_E(u,v)$. Because $C \in \mathcal{C}_{uv}(G\backslash uv)$, C' is a length-s u, v-separator in $G[S]\backslash uv$.

Hence, S is not an r-robust s-club based on the following chain of inequalities:

$$\begin{aligned}\rho_s(G[S];u,v) &= \rho_s(G[S]\backslash uv;u,v) + \mathbb{1}_E(u,v)\\ &\leq \kappa_s(G[S]\backslash uv;u,v) + \mathbb{1}_E(u,v)\\ &\leq |C'| + \mathbb{1}_E(u,v) \leq r-1.\end{aligned}$$

(The foregoing inequality does not make use of the length-bounded Menger's theorem for $s \in \{2,3,4\}$, only the dual relationship between $\rho_s(\cdot)$ and $\kappa_s(\cdot)$. Therefore, the length-s u, v-separator Inequality (1b) will be satisfied by the characteristic vector of every r-robust s-club even when $s \geq 5$.)

(⇐) Suppose S is not an r-robust s-club. It follows that there exist two vertices $u, v \in S$ such that $\rho_s(G[S]; u, v) \leq r-1$. Then, it follows from Proposition 1 that for $s \in \{2,3,4\}$,

$$\begin{aligned}\kappa_s(G[S]\backslash uv;u,v) + \mathbb{1}_E(u,v) &= \rho_s(G[S]\backslash uv;u,v) + \mathbb{1}_E(u,v)\\ &= \rho_s(G[S];u,v) \leq r-1.\end{aligned}$$

Now consider a minimum size length-s u, v-separator C' in $G[S]\backslash uv$. Then, $|C'| \leq r - 1 - \mathbb{1}_E(u,v)$. As before, $C' \cup (V\backslash S)$ belongs to $\mathcal{C}_{uv}(G\backslash uv)$, and the corresponding Constraint (1b) is violated by the characteristic vector of S. □

As noted in the proof of Theorem 6, Formulation (1) is a relaxation of the feasible region of the MRCP when $s \geq 5$. In this case, Formulation (1) may be satisfied by binary vectors that do not correspond to r-robust s-clubs. For instance, the vertex set of the graph in Figure 2 is not a 2-robust 5-club as $\rho_5(G;10,11) = 1$. However, we can satisfy all Constraints (1b) by setting $x_i = 1$ for every vertex i in the graph in Figure 2. In particular, as $\kappa_5(G;10,11) = 2$, every $C \in \mathcal{C}_{10,11}(G\backslash\{10,11\})$ contains at least two vertices and the left-hand side of Constraints (1b) is at most two.

In a practical implementation of cut-like Formulation (1), we would solve a relaxation that uses only a subset of constraints, then use a delayed constraint generation scheme to find a violated cut-like constraint on-the-fly to strengthen the relaxation. The separation problem is to identify Constraint (1b) violated by a given solution $x^* \in [0,1]^n$ to the relaxation or conclude that all such constraints are satisfied (Grötschel et al. 1993).

To solve this separation problem, we can treat x_i^* for $i \in V$ as vertex weights, and for each $\{u, v\} \in \binom{V}{2}$ find a length-s u,v-separator C of minimum weight $\sum_{i\in C} x_i^*$ in G. If we find a pair $\{u, v\}$ and a minimum weight separator C such that $(r - \mathbb{1}_E(u,v))(x_u^* + x_v^* - 1) > \sum_{i\in C} x_i^*$, we have identified a violated constraint; otherwise, we may conclude that no violated constraint exists. A minimum-weight length-s u, v-separator can be found in polynomial time for $s \in \{2,3,4\}$ (Lovász et al. 1978, Itai et al. 1982).

We can further strengthen Formulation (1) using the "conflict" inequalities:

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

The validity of these inequalities follows from the observation that a pair of distinct vertices u and v that do not have an adequate number of vertex-disjoint paths of length at most s cannot be simultaneously included in an r-robust s-club. In addition to strengthening the linear programming (LP) relaxation of Formulation (1), these inequalities are candidates for an initial relaxation that can be used in the aforementioned delayed constraint generation framework. We describe this approach in greater detail in Section 5.4.

The MRCP has been formulated for $s = 2$ and $s = 3$ in the literature. Proposition 10 in Section 3 of the online appendix establishes that the cut-like Formulation (1) has a tighter LP relaxation than the formulation of the maximum r-robust 2-club problem presented by Veremyev and Boginski (2012). Proposition 11 in Section 3 of the online appendix shows that the LP relaxations of the maximum r-robust 3-club problem formulation proposed by Almeida and Carvalho (2014) and that of the cut-like Formulation (1) strengthened by Inequalities (2) are incomparable. Nonetheless, our computational results in Section 6 using a decomposition branch-and-cut algorithm using Formulation (1) initialized by Inequalities (2) is faster overall than solving the formulation of Almeida and Carvalho (2014).

3.2. Cut-Like Formulation for the MHCP

Unlike an r-robust s-club, a pair of adjacent vertices in a t-hereditary s-club is not required to satisfy any additional requirements, because deletion of vertices will not affect the distance between adjacent vertices. In Formulation (3) that follows, binary variable x_i equals one if and only if vertex $i \in V$ is included in the t-hereditary s-club. Constraint (3b) ensures that if nonadjacent vertices u and v are selected in S, then at least t vertices are selected from each length-s u, v-separator C.

$$\max \sum_{i\in V} x_i \tag{3a}$$

$$\text{s.t.} \quad t(x_u + x_v - 1) \le \sum_{i\in C} x_i \quad \forall C \in \mathcal{C}_{uv}(G),\ \forall uv \in \overline{E}, \tag{3b}$$

$$x_i \in \{0,1\} \qquad \forall i \in V. \tag{3c}$$

Proposition 2. *Given a graph $G = (V, E)$, a subset of vertices S is a t-hereditary s-club if and only if its characteristic vector x^S satisfies the constraints of Formulation (3).*

Proof (⇒). Let $S \subseteq V$ and suppose that $t(x_u^S + x_v^S) - \sum_{i\in C} x_i^S > t$ for some $C \in \mathcal{C}_{uv}(G)$. This implies that $u, v \in S$ and $|C \cap S| < t$. Let $D = C \cap S$. Then S violates the definition of a t-hereditary s-club as $|D| < t$, and $S\backslash D$ is not an s-club as there is no u, v-path of length at most s in $G[S\backslash C]$.

(⇐) Suppose that $S \subseteq V$ is not a t-hereditary s-club in G. Then, it contains a deletion set $D \subset S$ (possibly empty) with $|D| < t$ such that $S\backslash D$ is not an s-club. Hence, there exist vertices $u, v \in S\backslash D$ such that the distance between them in $G[S\backslash D]$ is greater than s. Therefore, $D \cup (V\backslash S)$ is a length-s u, v-separator in G, and it can be verified that x^S violates the corresponding Constraint (3b). □

Clearly, it is sufficient to only consider length-s u, v-separators in (3b) that are minimal by inclusion, and $\mathcal{C}_{uv}(G)$ can be safely redefined to only contain minimal members. In particular, when $s = 2$, $\mathcal{C}_{uv}(G) = \{N(u) \cap N(v)\}$ as the common neighbors form the unique minimal length-2 u, v-separator. As a result, Formulation (3) generalizes the formulation of t-hereditary 2-clubs presented by Komusiewicz et al. (2019).

Although we are only required to consider minimal length-bounded separators in a complete and correct formulation, there can still be prohibitively many such sets to enumerate. We can use a delayed constraint generation scheme analogous to the one discussed in Section 3.1 but using the following separation problem instead.

Problem: Separation of length-s u, v-separator Inequalities (3b).

Input: A graph $G = (V, E)$, $x^* \in [0, 1]^n$, and positive integers t and s.

Output: If any exist, nonadjacent vertices $u, v \in V$ and a length-s u, v-separator $C \subseteq V\backslash\{u, v\}$ such that $t(x_u^* + x_v^* - 1) > \sum_{i\in C} x_i^*$.

As discussed in Section 3.1, we can solve this separation problem in polynomial time for $s \in \{2, 3, 4\}$ by finding a minimum-weight length-s u, v-separator, but the problem is NP-hard when $s \ge 5$ (Lovász et al. 1978, Itai et al. 1982). Consequently, Salemi and Buchanan (2020) show that when $t = 1$, determining whether a given point x^* satisfies all length-s u, v-separator Inequalities (3b) is coNP-complete for each $s \ge 5$ even if x^* is binary. Their result applies for every $t \ge 2$ after a slight modification.

Proposition 3. *For every pair of fixed integers $s \ge 5$ and $t \ge 1$, it is coNP-complete to determine whether a given $x^* \in \mathbb{R}^n$ satisfies all length-s u, v-separator Inequalities (3b).*

4. Existence of Convenient Formulations

In light of the exponential size of the cut-like formulations in the worst case, in this section we study whether the MRCP and the MHCP admit *convenient* IP formulations. First, for a formulation to be convenient, we must be able to write it down quickly (i.e., in polynomial time). This implies that the formulation must have polynomial size (i.e., be a *compact* formulation). Second, we require that candidate solutions to the problem (e.g., the MRCP or the MHCP) can quickly be converted into candidate solutions for the IP formulation. We use this notion because of our focus on formulations from which we can identify solutions to the problem being formulated with ease. Even if an IP formulation of the MRCP (or MHCP) with "extra" integer variables in addition to the characteristic vector is compact (polynomial size), it still may not be easy to verify if the characteristic vector of a candidate solution to the MRCP (or MHCP) is a feasible solution for such a formulation. This is because finding the values of the extra integer variables given the characteristic vector values may be hard. We formally define convenient formulations in Definition 6.

Definition 6. A mixed-integer linear programming formulation F for an optimization problem P is *convenient* if the following properties hold:

1. There is a polynomial-time algorithm for constructing formulation F when given an instance of problem P, and
2. There is a polynomial-time algorithm that, when given a candidate solution p to an instance of problem P, constructs a candidate solution f to formulation F such that p is feasible if and only if f is feasible, and the objective values of f and p are equal.

The results established in this section are summarized in Table 3. This may explain why previous works

Table 3. Existence of Convenient Formulations

Model	Constant fixed in the problem	Parameter specified in the input	Existence
r-Robust s-club	$s \leq 4$	r	Exist
	$s \geq 5$	r	Unlikely
	$r \geq 2$	s	Unlikely
t-Hereditary s-club	$s \leq 4$	t	Exist
	$s \geq 5$	t	Unlikely
	$t \geq 1$	s	Exist

(Veremyev and Boginski 2012, Almeida and Carvalho 2014) and ours have failed to create convenient formulations for general values of parameter s.

4.1. Existence of Convenient Formulations for *r*-Robust *s*-Clubs

Convenient formulations for r-robust 2-clubs and r-robust 3-clubs already exist in the literature (Veremyev and Boginski 2012, Almeida and Carvalho 2014). As the separation problem of MRCP for $s = 4$ can be reduced to a min-cut problem, based on the technique introduced by Martin (1991), we can create a convenient formulation for the MRCP when $s = 4$. The construction of convenient formulations in this manner is not very practical for our problems, so we do not discuss it here any further. Interested readers can refer to Carr and Lancia (2002) on building compact extended formulations provided that the separation problems admit compact LP formulations.

From Theorems 3 and 4, we can deduce the unlikelihood of convenient formulations for $s \geq 5$ and $r \geq 2$ as stated by the following two propositions proved in Section 4 of the online appendix.

Proposition 4. *If P ≠ NP, then for every fixed integer* $s \geq 5$ *there is no convenient IP formulation for the MRCP for arbitrary positive integer* r.

Proposition 5. *If P ≠ NP, then for every fixed integer* $r \geq 2$ *there is no convenient IP formulation for the MRCP for arbitrary positive integer* s.

4.2. Existence of Convenient Formulations for *t*-Hereditary *s*-Clubs

Formulation (3) is convenient when $s = 2$ because the common neighbors of nonadjacent vertices u and v form a unique minimal length-s u, v-separator. Convenient formulations also exist for t-hereditary s-clubs when $s \in \{3, 4\}$. From the convenient formulation of Almeida and Carvalho (2014) for r-robust 3-clubs (see Formulation (9) in Section 3 of the online appendix), we can drop the constraints on adjacent pairs of vertices to obtain a convenient formulation for t-hereditary 3-clubs. As the separation problem of the MHCP for $s = 4$ can be reduced to a min-cut problem (Lovász et al. 1978), analogous to our discussion regarding r-robust 4-clubs, the techniques presented by Martin (1991) can be used to construct a convenient formulation for the MHCP.

Next, we state in Proposition 6 an unlikelihood result for each $s \geq 5$ (proved similar to Proposition 4) based on the hardness result established in Theorem 5.

Proposition 6. *If P ≠ NP, then for every fixed integer* $s \geq 5$ *there is no convenient IP formulation for the MHCP for arbitrary positive integer* t.

Despite the foregoing negative result, we point out that convenient formulations for the MHCP do exist when t is a fixed positive integer, as stated in Proposition 7 (proved in Section 4 of the online appendix). Contrast the unlikelihood result of Proposition 5 with the affirmative result of Proposition 7; this is a consequence of Theorem 4 establishing the hardness of verifying r-robust s-clubs for fixed integer r. However, the formulations proposed in the proof of Proposition 7 are not expected to be of practical interest in solving large-scale instances of the problem.

Proposition 7. *There exist size* $O(tsn^{t+1})$ *IP formulations for the MHCP.*

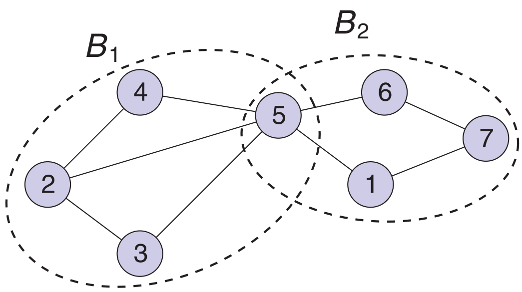
Corollary 3. *The MHCP admits a convenient formulation for every fixed* $t \geq 1$.

5. Recursive Block Decomposition Algorithm

In this section, we turn our focus to computational techniques that are effective in solving the problems using the cut-like formulations introduced in Section 3. Given a graph $G = (V, E)$, a *block* is a maximal biconnected[1] subgraph of G, and the *block decomposition* of G is the collection of all the blocks of G (Figure 3). Every vertex of G belongs to some block in the decomposition, and two distinct blocks of G can share at most one vertex (if two blocks share two or more vertices, none of them can be a cut-vertex and the blocks can be merged together, contradicting their maximality; see West 2001). Every r-robust (t-hereditary) s-club, whenever $r, t \geq 2$, must be contained within a single block of G. Based on this observation, we present a block decomposition approach to solve the MRCP and the MHCP.

The main idea is to decompose the original graph G into many smaller blocks so we can restrict our attention to one block at a time. Furthermore, with the help

Figure 3. (Color online) Graph G That Decomposes into Two Blocks B_1 and B_2



of a feasible solution obtained using a heuristic we can apply preprocessing techniques on each block. As a result, the preprocessed "blocks" may no longer be biconnected, entailing further decomposition into even smaller blocks. This motivates our *recursive block decomposition* approach to solve the MRCP and the MHCP, described in Algorithm 1. We note here that maximal r-connected subgraphs of G could be used in this approach instead of blocks, whenever $r \geq 3$. However, our preliminary experiments indicated that repeatedly finding them was time consuming and the tradeoff was not favorable in terms of finding potentially smaller subgraphs on which to solve the problems. Pertinently, the computational complexity of finding all blocks in a graph G is $O(m+n)$ (Hopcroft and Tarjan 1973), whereas, to our best knowledge, finding all maximal r-connected subgraphs of G can only be performed in $O(mn^2\min\{r, \sqrt{n}\})$ time (Matula 1978, Carmesin et al. 2014).

Algorithm 1 (Recursive Block Decomposition for the MRCP and the MHCP)

Input: A graph $G = (V, E)$.
Output: A maximum cardinality r-robust (t-hereditary) s-club K.

```
1  find the block decomposition 𝓑 of G
2  K ← a heuristic solution of MRCP (MHCP) on the
   largest block in 𝓑
3  while 𝓑 ≠ ∅ do
4      pick D ∈ 𝓑 with the most vertices
5      if |D| ≤ |K| then
6          return K
7      𝓑 ← 𝓑 \ {D}
8      preprocess block D by vertex peeling using sol-
       ution K
9      find the block decomposition 𝓕 of D
10     if |𝓕| = 1 then
11         K' ← a maximum r-robust (t-hereditary) s-club
           in D
12         if |K'| > |K| then
13             K ← K'
14     else
15         𝓑 ← 𝓑 ∪ 𝓕
16 return K
```

We choose a "greedy" strategy for solving the MRCP (MHCP) on a block with most vertices first and then update the current best solution as needed after each block is considered. If the block with the most vertices has fewer vertices than the current largest solution, the algorithm is terminated, and the current best solution is indeed optimal. In line 1 of Algorithm 1, we find the block decomposition of G in $O(m+n)$ time (Hopcroft and Tarjan 1973). On a block with the most vertices, we find a heuristic solution (line 2) that serves as a lower bound. In the while-loop, we preprocess the current block (line 8), decompose it into smaller blocks if possible (line 9), and add them to collection $\mathcal{B}$ for future consideration (line 15). If it cannot be further decomposed after preprocessing, we solve the MRCP (MHCP) on this block (line 11). The algorithm terminates when the largest unexplored block contains fewer vertices than the current best objective value.

Next, we discuss ideas for reducing the computational burden involved in computing $\rho_s(\cdot)$, which is a frequent task in the heuristic used in line 2, preprocessing used in line 8, and implementing the initial relaxation used in the decomposition branch-and-cut algorithm in line 11 of Algorithm 1. We discuss other details in Sections 5.2, 5.3, and 5.4.

5.1. Computing Bounds on $\rho_s(\cdot)$

Recall from Section 2.2 that $\rho_s(G; u, v)$ denotes the maximum number of internally vertex-disjoint u, v-paths of length at most s in G. When $s = 2$, we know that $\rho_s(G; u, v) = |N(u) \cap N(v)| + \mathbb{1}_E(u, v)$, which can be computed in $O(n)$ time.

For $s \in \{3, 4\}$, we can use the approach introduced by Itai et al. (1982) that applies max flow–min cut theorem to an auxiliary flow network that can be constructed in $O(m)$ time, to compute $\rho_s(G \backslash uv; u, v)$ and find a minimum cardinality length-s u, v-separator. If we compute $\rho_s(G \backslash uv; u, v)$ using a flow augmenting algorithm, we can terminate early, after we confirm that the flow value is at least r (before computing its actual value). Hence, for a given pair of vertices $u, v \in V$, we can check if $\rho_s(G; u, v) \geq r$ using the Ford and Fulkerson (1956) algorithm in $O(rm)$ time. As we only consider small values of r in our experiments, the approach essentially takes $O(m)$ time, if we treat r as a constant.

Limiting the number of u, v-pairs for which we need to compute, $\rho_s(\cdot)$ can be even more significant from a computational perspective. For example, if $|N(u) \cap N(v)| \geq r$, then $\rho_s(G; u, v) \geq r$. Likewise, if u and v are in different connected components, or even if they are in the same connected component but in different *blocks*, we know that $\rho_s(G; u, v) \leq 1$. These observations motivate us to explore techniques to quickly or incrementally compute upper and lower bounds of $\rho_s(\cdot)$ that can be used to reduce the overall computational overhead. Given the value of $\rho_2(\cdot)$, we can calculate upper bounds of $\rho_s(\cdot)$ for any $s \geq 3$ based on a recursive relationship between $\rho_{s-1}(\cdot)$ and $\rho_s(\cdot)$ stated in Lemma 2 (proved in Section 5 of the online appendix).

Lemma 2. *Given a graph $G = (V, E)$, a pair of vertices $uv \in \binom{V}{2}$, and a positive integer $s \geq 3$, we have*

$$\rho_s(G; u, v) \leq \mathbb{1}_E(u, v) + \sum_{w \in N(v) \setminus \{u\}} \min\{1, \rho_{s-1}(G; u, w)\}.$$

Consider any valid upper bound $\overline{\rho}_s(G; u, v) \geq \rho_s(G; u, v)$. We will refer to $\overline{\rho}_s(G; u, \cdot)$ as the *single-source upper bounds.* Given $\overline{\rho}_{s-1}(G; u, \cdot)$, we can compute upper bounds $\overline{\rho}_s(G; u, v)$ in $O(\deg(v))$ time and single-source upper bounds $\overline{\rho}_s(G; u, \cdot)$ in $O(m)$ time using the recursion:

$$\overline{\rho}_s(G; u, v) = \mathbb{1}_E(u, v) + \sum_{w \in N(v) \setminus \{u\}} \min\{1, \overline{\rho}_{s-1}(G; u, w)\}. \quad (4)$$

For each pair of vertices $uv \in \binom{V}{2}$, Algorithm 3 in Section 6 of the online appendix describes a simple heuristic to obtain a lower bound of $\rho_s(G; u, v)$ for any $s \geq 3$, which we denote by $\hat{\rho}_3(G; u, v)$. Essentially, the algorithm constructs a matching in the bipartite graph $G_{uv} = (V_{uv}, E_{uv})$, where $E_{uv} := \{\{p, q\} \in E \mid p \in N(u) \setminus N[v], q \in N(v) \setminus N[u]\}$ and V_{uv} is the union of endpoints of edges in E_{uv}. The size of the matching then gives us a lower bound on the number of disjoint length-3 paths between vertices u and v, because each edge in the matching is the inner edge of such a path. This algorithm can be implemented to run in $O(m)$ and is usually fast in practice. Previous empirical studies also report that the simple greedy matching heuristic used in this algorithm can usually produce a solution at least 90% the size of an optimum, even though it is only known to guarantee a 2-approximation (Möhring and Müller-Hannemann 1995, Magun 1998, Langguth et al. 2010).

If the lower bound $\hat{\rho}_s(G; u, v)$ is at least r or the upper bound $\overline{\rho}_s(G; u, v)$ is at most $r - 1$, there is no need to run the Ford–Fulkerson algorithm. Using this observation significantly decreases the number of pairs of vertices that require the application of the Ford–Fulkerson algorithm and the overall running time taken to check if $\rho_s(G; u, v) \geq r$ (see Table 9 in Section 7 of the online appendix). Taking the instance PGP as an example, it suffices to verify $\rho_4(\cdot) \geq 2$ for 727,213 of 57,025,860 pairs of vertices using the Ford–Fulkerson algorithm, reducing the running time required from 155.08 seconds to 5.12 seconds (which includes the time to compute the lower and upper bounds).

5.2. Heuristics

In this section, we discuss heuristics for finding a feasible solution that we subsequently use for preprocessing in Section 5.3. The first heuristic described in Algorithm 2 for finding an r-robust s-club, generalizes the greedy vertex elimination heuristic proposed by Bourjolly et al. (2000) for finding an s-club.

A pair of vertices $i, j \in V$ cannot be included in the same r-robust s-club if $\rho_s(G; i, j) \leq r - 1$. We call a pair of vertices i and j *compatible* if they satisfy $\rho_s(G; i, j) \geq r$. Our heuristic first builds a maximal subset $S \subseteq V$ that is pairwise compatible. Essentially, we seek a maximal clique S in the compatibility graph $G^c = (V, E^c)$, where $E^c := \left\{ ij \in \binom{V}{2} \,\middle|\, \rho_s(G; i, j) \geq r \right\}$. This is similar to a technique used for s-clubs by Salemi and Buchanan (2020), where they begin by first constructing its (weakly) hereditary counterpart (Pattillo et al. 2013), that is, an s-clique on an analogous compatibility graph. Although $\rho_s(G; i, j) \geq r$, it is possible that $\rho_s(G[S]; i, j) \leq r - 1$ in the vertex subset S used in Algorithm 2. Hence, we need to check if S is an r-robust s-club in G; if not, we select a vertex $v \in S$ that has the most vertices w such that $\rho_s(G[S]; v, w) \leq r - 1$ and remove it from S. We repeat this step until S is an r-robust s-club.

Algorithm 2 (Heuristic for Finding an r-Robust s-Club)

```
    Input: A graph G = (V, E).
    Output: An r-robust s-club S.
 1  create compatibility graph G^c ← (V, E^c), where
      E^c := { ij ∈ (V choose 2) | ρ_s(G; i, j) ≥ r }
 2  S ← a maximal clique in G^c
 3  while S ≠ ∅ do
 4      τ_i ← 0, ∀i ∈ S
 5      for ij ∈ (S choose 2) do
 6          if ρ_s(G[S]; i, j) ≤ r − 1 then
 7              τ_i ← τ_i + 1
 8              τ_j ← τ_j + 1
 9      v ← arg max_{i∈S} τ_i
10      if τ_v ≥ 1 then
11          S ← S \ {v}
12      else
13          return S
```

When $s \in \{3, 4\}$, line 1 in Algorithm 2 can be completed in $O(rn^2m)$, and line 2 can be implemented to run in $O(|V| + |E^c|)$ time (Walteros and Buchanan 2020); we use the implementation provided by Salemi and Buchanan (2020) for this step. The while-loop in line 3 may require at most $\omega(G^c)$ (the clique number) iterations to complete, and the for-loop (line 5) in each iteration requires at most $O(rn^2m)$ time to complete.

The computational effort needed by this heuristic is dominated by the computation of pairwise $\rho_s(\cdot)$ values. Therefore, when constructing the compatibility graph in line 1, we take as much advantage of the bounds discussed in Section 5.1 as possible. We first check if the lower bound $\hat{\rho}_s(G; i, j)$ is at least r, in which case we create edge ij. Next we check if the upper bound $\overline{\rho}_s(G; i, j)$ is at most $r - 1$, in which case we can conclude that the pair is incompatible. The lower and upper bounds are similarly used in line 6 to check the condition of the if-statement faster. The Ford–Fulkerson algorithm is used to exactly verify the conditions only in cases where verification using the bounds has been inconclusive. As a result, the running times of the heuristic are reasonable for a one-time application. As reported in Table 11 in Section 7 of the online appendix, the longest time taken by the heuristic was 9.58 seconds for the instance `email` with $s = 4$ and $r = 3$ and the average time taken across all instances is 0.43 seconds. As reported in Table 13 in Section 7 of the online appendix, r-robust s-clubs found by this heuristic were subsequently proved to be optimal in 50 of 126 instances.

We can modify Algorithm 2 to find t-hereditary s-clubs for $s \in \{2,3,4\}$ by modifying lines 1, 5, and 6 as follows: we set $E^c := \{ij \in \binom{V}{2} \mid ij \in E \text{ or } \rho_s(G; i, j) \geq t\}$ in line 1; change $ij \in \binom{S}{2}$ to $ij \in \binom{S}{2} \setminus E$ in line 5; and use $t-1$ instead of $r-1$ in line 6. As reported in Tables 12 and 14 in Section 7 of the online appendix, the average time taken by this heuristic across all instances is 0.48 seconds, and t-hereditary s-clubs found by this heuristic were subsequently proved to be optimal in 48 of 126 instances.

5.3. Preprocessing

Vertex peeling is a generic term applied to techniques in which we delete vertices from the graph based on a heuristic solution, without affecting the optimality guarantee and correctness of a subsequent exact algorithm. We discuss a vertex peeling technique applicable to the MRCP and then extend this idea to the MHCP for $s \in \{2,3,4\}$.

Given a solution of size ℓ for the MRCP, we can delete a vertex that has fewer than ℓ distance-s neighbors, as it cannot be part of a solution whose size is greater than ℓ. This technique is often used when solving the maximum s-club problem (Veremyev and Boginski 2012, Lu et al. 2018, Moradi and Balasundaram 2018, Salemi and Buchanan 2020). For the MRCP, we can strengthen this idea by deleting a vertex $v \in V$ if it has fewer than ℓ *compatible* distance-s neighbors, that is, $|T_v| < \ell$, where $T_v := \{u \in N^s_G(v) \mid \rho_s(G; v, u) \geq r\}$.

Consider the graph on the left in Figure 4 that contains the 2-robust 2-club {1, 2, 3}, that is, $\ell = 3$. For each vertex $v \in \{1,2,3,4,5\}$, we can see that $|T_v| \geq 3$ and therefore, they are not removed by vertex peeling. However, for each $v \in \{6,7,8\}$, we have $|T_v| < 3$. In particular, although the distance-2 neighborhood of vertex 7 contains three vertices, the set T_7 is a singleton containing just vertex 5. Therefore, vertices {6, 7, 8} and their incident edges are removed by vertex peeling shown on the right in Figure 4.

In addition, vertices of degree less than r can also be removed from G as the degree of every vertex in an r-robust s-club must be at least r. In other words, every r-robust s-club in G is contained within its r-core (the maximal induced subgraph of G with minimum degree at least r). We can recursively implement these ideas as each vertex v that is removed may affect the size of the distance-s neighborhood or the degree of another vertex. Pseudocode for vertex peeling is described in Algorithm 4 in Section 6 of the online appendix.

The r-core of G in line 2 of Algorithm 4 can be be found using an $O(m+n)$ algorithm (Matula and Beck 1983, Batagelj and Zaveršnik 2011). The repeat-until loop may execute at most n times, and each iteration can be completed in $O(rn^2m)$ time if we exhaustively verify $\rho_s(\cdot) \geq r$ for every vertex pair. Despite what the worst-case complexity suggests, our vertex peeling implementation is reasonably quick on our test bed of instances. As mentioned before, it is not always necessary to compute $\rho_s(G; u, v)$ if we can determine that $\hat{\rho}_s(G; u, v) \geq r$ or $\overline{\rho}_s(G; u, v) \leq r-1$. The longest time taken by vertex peeling is 10.73 seconds for the instance `PGP` with $s = 4$ and $r = 2$, and the procedure took 0.44 seconds on average across our test bed. Approximately 90% of the instances in our test bed were preprocessed in less than one second (see Table 15 in Section 7 of the online appendix). Across our test bed, on average 24.3% vertices were removed by vertex peeling, whereas approximately 3.3% vertices were not considered by Algorithm 1 because of early termination (see Table 22 in Section 7 of the online appendix). The number of vertex pairs we need to consider is also reduced by decomposing the graph into blocks and applying vertex peeling to each block.

We can extend the vertex peeling ideas discussed above to the MHCP given a t-hereditary s-club of size ℓ. Recall from Proposition 1 that verifying if S is a t-hereditary s-club, for $s \in \{2,3,4\}$, is equivalent to checking if $\rho_s(G[S]; u, v) \geq t$ for every pair of nonadjacent vertices u and v in S. Hence, a vertex v may be deleted if $\rho_s(G; u, v) \leq t-1$ for a sufficient number of its nonadjacent distance-s neighbors, that is, if $|W_v| + |N_G(v)| < \ell$, where $W_v := \{u \in N^s_G(v) \setminus N_G(v) \mid \rho_s(G; u, v) \geq t\}$. The pseudocode of vertex peeling for the MHCP when

Figure 4. (Color online) Vertex Peeling for the MRCP

$s \in \{2,3,4\}$ can be obtained with small modifications described in the comments of Algorithm 4 in Section 6 of the online appendix. We found its running time performance on our test bed to be comparable to its MRCP counterpart (see Table 16 in Section 7 of the online appendix), and 23.7% vertices on average across our test bed were removed by vertex peeling (see Table 23 in Section 7 of the online appendix).

5.4. Delayed Constraint Generation

In this section, we describe decomposition approaches for solving the MRCP and the MHCP when $s \in \{2,3,4\}$. We use the following relaxation based on conflict Inequalities (2) at the root node of the branch-and-cut (BC) tree for the MRCP when $s \in \{2,3,4\}$:

$$\max \sum_{i \in V} x_i \tag{5a}$$

$$\text{s.t.} \quad x_u + x_v \le 1 \quad \forall uv \in \binom{V}{2} : \rho_s(G; u, v) \le r - 1, \tag{5b}$$

$$x_i \in \{0,1\} \quad \forall i \in V. \tag{5c}$$

The BC algorithm starts by solving the initial relaxation (5) at the root node and branches when the LP relaxation optimum is fractional. It also prunes the search tree as usual when the node LP relaxation is infeasible or when the incumbent solution has an objective value that is better than or equal to the node LP bound. If we obtain an integral optimum $x^* \in \{0,1\}^n$ at some node of the BC tree, we check if the selected vertices $S := \{i \in V \mid x_i^* = 1\}$ form an r-robust s-club. Specifically, for each pair of vertices $u, v \in S$, we have to check if $\rho_s(G[S]; u, v) \ge r$. If S is an r-robust s-club, then that node can be pruned by feasibility and the incumbent is updated if necessary.

If we detect a pair of vertices $u, v \in S$ such that $\rho_s(G[S]; u, v) \le r - 1$, then S is not an r-robust s-club, and we construct a length-s u, v-separator that corresponds to Constraint (1b) violated by x^*. If $s = 2$, then $N(u) \cap N(v)$ is the unique minimal length-2 u, v-separator in $G \backslash uv$. If $s \in \{3,4\}$, we first identify a minimum cardinality length-s u, v-separator C in $G[S] \backslash uv$ using the max flow–min cut theorem on the auxiliary flow network construction described by Itai et al. (1982). The set $S' := C \cup (V \backslash S)$ is then a length-s u, v-separator in $G \backslash uv$. It is then made minimal using the Minimalize procedure of Salemi and Buchanan (2020), which removes a vertex $w \in S'$, chosen arbitrarily, after verifying that w cannot belong to a length-s u, v-path in $G' := G - (S \backslash \{w\})$. This is done by checking whether $\text{dist}_{G'}(u, w) + \text{dist}_{G'}(v, w) > s$. Their implementation also suggests speedups to this procedure. For instance, by initially deleting every vertex $w \in S'$ such that $\text{dist}_G(u, w) + \text{dist}_G(v, w) > s$, we can save time on repeated distance computations. We can also skip the vertices in $N(u) \cap N(v) \cap S'$ as they must belong to a minimal separator.

Using Proposition 1 when $s \in \{2,3,4\}$, we can tackle the MHCP using a decomposition BC algorithm that starts by solving the relaxation presented here:

$$\max \quad \sum_{i \in V} x_i \tag{6a}$$

$$\text{s.t.} \quad x_u + x_v \le 1 \quad \forall uv \in \overline{E} : \rho_s(G; u, v) \le t - 1, \tag{6b}$$

$$x_i \in \{0,1\} \quad \forall i \in V. \tag{6c}$$

As before, if we encounter an integral solution $x^* \in \{0,1\}^n$ at some node of the BC tree, we need to check if the selected vertices $S := \{i \in V \mid x_i^* = 1\}$ form a t-hereditary s-club. Specifically, we check if $\rho_s(G[S]; u, v) \ge t$ for every pair of nonadjacent vertices u and v in the induced subgraph $G[S]$. If $\rho_s(G[S]; u, v) \le t - 1$ for some pair u and v, we add a length-s u, v-separator inequality $t(x_u + x_v - 1) \le \sum_{i \in C} x_i$ violated by x^*, in a manner analogous to the foregoing discussion for the MRCP.

During implementation, we take advantage of the lower and upper bounds introduced in Section 5.1 to quickly build the conflict constraints in the initial relaxations (5) and (6), by limiting the number of times we exactly verify if $\rho_s(G; u, v)$ is small enough. Furthermore, we solve the MRCP and the MHCP using the recursive block decomposition algorithm. The BC algorithms described above are only applied on blocks that are irreducible by vertex peeling, which helps reduce the size of the instance solved by the BC algorithms.

6. Computational Study

The goal of the computational experiments is to assess the effectiveness of the cut-like IP formulations, preprocessing techniques, and the recursive block decomposition algorithm for solving the MHCP and the MRCP. We selected instances from the Tenth DIMACS Implementation Challenge on Clustering (Bader et al. 2013) that are frequently used as benchmarks for the maximum s-club problem. Numerical results are reported and discussed for the MRCP and the MHCP for the parameters $r \in \{2,3,4\}$ and $t \in \{2,3,4\}$, respectively. An instance in our DIMACS-10 test bed is thus defined by a graph from the collection and a value for the parameter r or t. Note that our approaches are applicable for any positive integer-valued parameter t or r, although we require $s \in \{2,3,4\}$ in our experiments. Formulation (1) for the MRCP is valid only for these values of s. Although Formulation (3) of the MHCP is valid for every positive integer s, our separation procedure that is based on computing $\rho_s(\cdot)$ is only valid for $s \in \{2,3,4\}$.

All algorithms evaluated in this computational study are implemented in C++, and Gurobi™ Optimizer 9.0.1 (Gurobi Optimization 2020) is used to solve the IP formulations in its default settings, other than the use of "lazy cuts" feature to implement the decomposition BC algorithm described in Section 5.4.

We impose a one-hour wall-clock time limit per instance for all solvers. If an instance was not solved to optimality within the time limit, we report the relative optimality gap calculated as (best bound − best objective)/best objective × 100%. In addition, we set the Gurobi cut-off parameter to the size of the largest r-robust (t-hereditary) s-club known at the time of calling the Gurobi optimization solver, informing the solver that we are only interested in solutions with better objective values. We conduct all numerical experiments on a 64-bit Linux® compute node running a dual Intel® Skylake 6130 processor with 32 cores, 2.10 GHz CPUs, and 96 GB RAM. We use parallel programming with the OpenMP library (Dagum and Enon 1998) when we implement the computation of length-bounded vertex-disjoint paths. Specifically, the tasks of finding the lower bound $\hat{\rho}_s(\cdot)$, the upper bound $\overline{\rho}_s(\cdot)$, and the exact value $\rho_s(\cdot)$ are parallelized. We observe eightfold speed-up with OpenMP using all 32 cores over a single-threaded implementation; Table 10 in Section 7 of the online appendix contains more details.

6.1. Assessing the Cut-Like Formulations, Recursive Block Decomposition, and Preprocessing When $s = 2$

In this section, we focus on the case $s = 2$ as it admits comparison between multiple competing mathematical programming approaches. Specifically, we assess the performance of the recursive block decomposition Algorithm 1 to solve the MRCP and the MHCP using the delayed constraint generation scheme from Section 5.4 (labeled as "BCUT" in the tables), by comparing it against an implementation of the delayed constraint generation scheme without block decomposition, preprocessing, or speed-ups achieved using $\rho_s(\cdot)$-bounds (labeled as "CUT" in the tables). This comparison serves to highlight the impact of the graph decomposition and IP model decomposition techniques that we introduce, along with preprocessing and other ideas for achieving better performance. A direct "monolithic" implementation of the common neighbor formulation also serves as a baseline solver in this study (labeled as "CN" in the tables). The comparison between CUT and CN serves to demonstrate the benefits of using delayed constraint generation for the MHCP and the MRCP even when the formulation is compact.

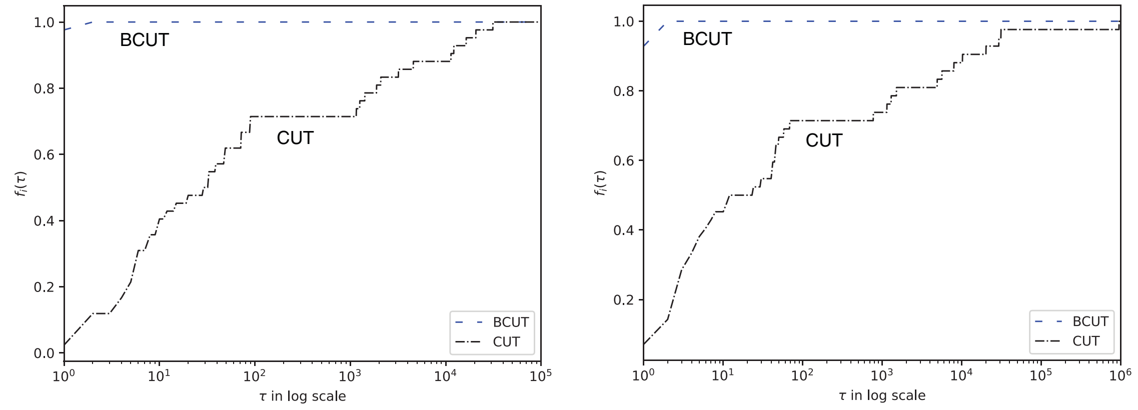
Table 4. Optimal Objective Values for the Maximum r-Robust 2-Club and t-Hereditary 2-Club Problems Found by BCUT

			MRCP, $s = 2$			MHCP, $s = 2$		
Graph	n	m	$r = 2$	$r = 3$	$r = 4$	$t = 2$	$t = 3$	$t = 4$
karate	34	78	12	6	6	12	6	6
dolphins	62	159	9	7	6	9	7	6
lesmis	77	254	18	14	13	18	14	13
polbooks	105	441	20	15	**12**	20	15	**13**
adjnoun	112	425	23	12	**6**	23	12	**9**
football	115	613	14	13	**12**	14	13	**13**
jazz	198	2,742	79	73	65	79	73	65
celegans	453	2,025	104	54	30	104	54	30
email	1,133	5,451	27	23	**19**	27	23	**20**
polblogs	1,490	16,715	232	182	**158**	232	182	**159**
netscience	1,589	2,742	22	21	20	22	21	20
power	4,941	6,594	9	7	6	9	7	6
hep-th	8,361	15,751	33	24	24	33	24	24
PGP	10,680	24,316	96	71	64	96	71	64

Note. Instances for which the size of a maximum r-robust 2-club and t-hereditary 2-club differ are in bold.

Figure 5 shows performance profiles (Dolan and Moré 2002, Gould and Scott 2016) based on the wall-clock running times of solvers CUT and BCUT for the maximum r-robust 2-club and t-hereditary 2-club problems across all instances in our test bed. For each solver i we plot $f_i(\tau)$—the fraction of the test instances for which the running time required by solver i is at most a factor τ of the running time of the fastest solver for that instance. Following convention, we take the solution time to be equal to the time limit for instances that terminated by reaching the time limit (Dolan and Moré 2002). The

Figure 5. (Color online) Performance Profiles of Solvers CUT and BCUT for the Maximum r-Robust 2-Club (Left) and t-Hereditary 2-Club (Right) Problems



performance profiles reflect the dominant performance of BCUT, which solved all the instances of the MRCP and the MHCP in this test bed to optimality when $s = 2$. Table 4 reports the optimal objective values of the MRCP and the MHCP found by BCUT.

From the details reported in Tables 5 and 6 for the MRCP and the MHCP, respectively, we can see that BCUT outperforms CN and CUT on all instances with more than 150 vertices, demonstrating the effectiveness of the recursive block decomposition algorithm and preprocessing. For example, for the maximum r-robust 2-club problem, BCUT solves instance `PGP` with $r = 4$ to optimality in 0.10 seconds, whereas CUT takes 3,074.4 seconds, and CN fails to solve this instance under the time limit. Similarly, for the maximum t-hereditary 2-club problem, BCUT solves instance `PGP` with $t = 4$ to optimality in 0.12 seconds, whereas both CUT and CN fail to solve this instance under the time limit.

For the maximum r-robust 2-club problem, we also compare BCUT against the fastest of the four solvers recently introduced by Veremyev et al. (2022). Table 24 in Section 7 of the online appendix shows that, for most instances, BCUT performs better in terms of running times when solving the maximum r-robust 2-club problem for $r \in \{2, 3\}$. However, the comparison is limited by the fact that the implementations of Veremyev et al. (2022) use a different programming language. Komusiewicz et al. (2019) also study the MRCP and the MHCP for $s = 2$ as mentioned earlier. Although direct comparison with the numerical results in Komusiewicz et al. (2019) is limited by differences in hardware/software, it is safe to say that the performance of BCUT is comparable to their solvers based on a direct comparison of running times over the instances that are common between the test beds.

6.2. Assessing the Cut-Like Formulation for the MRCP When $s = 3$

For the case $s = 3$, although no competing formulation is available for the MHCP, the formulation introduced by Almeida and Carvalho (2014) for the MRCP can be compared against the cut-like formulation (1). To isolate the effect of the formulation used in the recursive block decomposition algorithm, we only change the exact approach used in line 11 of Algorithm 1 and compare running times using the AC formulation (9) (see Section 3 of the online appendix) and the delayed constraint generation approach from Section 5.4. The performance profiles in Figure 6 based on the wall-clock running times of these solvers show that the recursive block decomposition algorithm (with the same heuristic and preprocessing for both solvers) performs better when the cut-like formulation is used compared with the AC formulation (9). For example, BCUT solves the instance `hep-th` for $r = 2$ to optimality in 16.8 seconds, whereas AC fails to solve this instance under the time limit. For some challenging instances, the optimality gap using BCUT is smaller when both solvers fail to solve to optimality. Detailed results are reported in Table 17 in Section 7 of the online appendix.

6.3. Assessing the Impact of Fault-Tolerance on Solution Size

At this time, no numerical results for the MRCP and the MHCP are available in the literature for $s \geq 3$. In Table 7, we report results obtained using the BCUT

Table 5. Wall-Clock Running Time (in Seconds) for Solving the Maximum r-Robust 2-Club Problem

			Wall-clock running time								
			$r = 2$			$r = 3$			$r = 4$		
Graph	n	m	CN	CUT	BCUT	CN	CUT	BCUT	CN	CUT	BCUT
karate	34	78	0.05	0.03	0.03	0.04	0.03	0.00	0.03	0.02	0.02
dolphins	62	159	0.15	0.06	0.03	0.08	0.06	0.00	0.06	0.05	0.04
lesmis	77	254	0.15	0.08	0.00	0.83	0.08	0.00	0.18	0.09	0.00
polbooks	105	441	0.36	0.14	0.03	3.25	0.13	0.03	0.33	0.20	0.03
adjnoun	112	425	0.29	0.15	0.03	0.16	0.18	0.02	0.09	0.39	0.03
football	115	613	0.31	0.10	0.11	0.31	0.07	0.00	0.26	0.07	0.00
jazz	198	2,742	0.87	0.19	0.06	1.24	0.20	0.05	1.00	0.20	0.04
celegans	453	2,025	3.30	1.41	0.02	71.68	1.38	0.02	30.02	1.11	0.03
email	1,133	5,451	318.11	109.48	7.38	121.89	38.12	0.53	37.63	13.40	0.28
polblogs	1,490	16,715	1,382.21	22.39	5.25	**4.52%**	56.15	7.69	**3.21%**	61.49	6.61
netscience	1,589	2,742	34.78	22.64	0.00	37.21	19.97	0.01	39.24	15.23	0.01
power	4,941	6,594	**240.00%**	625.26	0.50	1,363.12	53.24	0.02	790.46	41.31	0.00
hep-th	8,361	15,751	**136.84%**	1,299.56	0.69	**144.44%**	1,284.72	0.28	**70.83%**	897.76	0.07
PGP	10,680	24,316	LPNS	1,479.27	0.71	LPNS	LPNS	0.22	LPNS	3,074.40	0.10

Notes. If an instance was not solved to optimality under the time limit, the optimality gap is reported (highlighted in bold). The entry "LPNS" means that the root LP relaxation was not solved to optimality under the one-hour time limit. An entry of 0.00 means the run took less than 0.005 seconds.

Table 6. Wall-Clock Running Times (in Seconds) for Solving the Maximum t-Hereditary 2-Club Problem

	Wall-clock running time											
			$t = 2$			$t = 3$			$t = 4$			
Graph	n	m	CN	CUT	BCUT	CN	CUT	BCUT	CN	CUT	BCUT	
karate	34	78	0.04	0.05	0.05	0.03	0.06	0.04	0.03	0.04	0.04	
dolphins	62	159	0.12	0.05	0.05	0.21	0.05	0.02	0.12	0.06	0.03	
lesmis	77	254	0.15	0.07	0.00	0.12	0.06	0.00	0.10	0.06	0.00	
polbooks	105	441	0.27	0.09	0.02	1.15	0.10	0.03	1.16	0.09	0.04	
adjnoun	112	425	0.29	0.09	0.03	1.25	0.38	0.05	1.08	0.31	0.15	
football	115	613	0.21	0.09	0.10	0.17	0.09	0.00	0.25	0.09	0.09	
jazz	198	2,742	0.58	0.18	0.06	0.98	0.21	0.05	0.94	0.24	0.04	
celegans	453	2,025	3.30	1.26	0.02	65.61	1.08	0.02	41.52	1.10	0.05	
email	1,133	5,451	460.84	109.72	9.79	260.83	45.89	1.03	67.91	14.50	0.36	
polblogs	1,490	16,715	1,484.41	18.38	5.08	**55.71%**	53.69	8.57	**54.13%**	76.73	7.18	
netscience	1,589	2,742	38.74	18.53	0.00	34.82	18.43	0.02	40.08	19.72	0.01	
power	4,941	6,594	1,762.25	444.17	0.58	2,411.53	656.91	0.02	2,351.34	1,919.99	0.00	
hep-th	8,361	15,751	**130.00%**	1,034.27	0.80	**136.84%**	1,832.96	0.37	**131.58%**	1,679.40	0.08	
PGP	10,680	24,316	LPNS	LPNS	0.64	LPNS	LPNS	0.35	LPNS	LPNS	0.12	

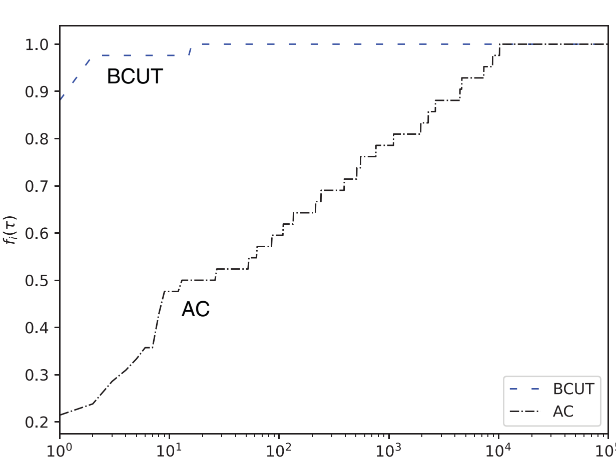
Notes. If an instance was not solved to optimality, the optimality gap is reported (highlighted in bold). The entry "LPNS" means that the root LP relaxation was not solved to optimality under the one-hour time limit. An entry of 0.00 means the run took less than 0.005 seconds.

solver for the MRCP and the MHCP for $r, t \in \{2, 3, 4\}$ and $s \in \{3, 4\}$. All instances in the test bed except the graph `email` for $s = r = t = 3$ were solved to optimality.

To understand the impact of the fault-tolerance requirements of the MRCP and the MHCP on the solution, we compare the size of the largest t-hereditary s-club, r-robust s-club, and the relaxation of r-robust s-club requiring r *distinct* paths of length at most s for all vertex pairs considered by Veremyev and Boginski (2012). The "distinct-path relaxation" of the MRCP is found by directly solving the formulation of Veremyev and Boginski (2012).

The best objective values of the MRCP and the MHCP are the same for most instances, and when they are different, the t-hereditary s-club is larger than the r-robust s-club (for $r = t$), consistent with Lemma 1. However, it can also be seen from Table 7 that the r-distinct-path s-clubs found are on average 60.64% larger than the r-robust s-clubs found and on average 19.71% smaller than the maximum s-club size. These observations indicate that requiring r distinct length-s paths between vertices in s-clubs may not be sufficient if we seek fault-tolerant s-clubs. For example, the maximum 3-robust and 3-hereditary 3-clubs found by our solver in the instance `lesmis` are the same 25 vertices, whereas the optimal solution for the 3-distinct-path relaxation contains 45 vertices. We also verified computationally that the optimal 3-distinct-path 3-club we found in `lesmis` can be disconnected by deleting a single vertex.

Figure 6. (Color online) Performance Profiles of Solvers AC and BCUT for the Maximum r-Robust 3-Club Problem



Finally, we report on the performance of BCUT on graph instances generated using models proposed by Watts and Strogatz (1998) and Gendreau et al. (1993). Specifically, we choose the Watts-Strogatz (WS) instances used in (Veremyev et al. 2022) and Gendreau instances used in (Veremyev and Boginski 2012). WS instances range from 100 to 1,000 vertices, whereas Gendreau instances have 100, 200, or 300 vertices with different edge densities. The average running times for the MRCP and the MHCP on WS instances are 0.36 and 3.42 seconds, respectively. When $r = t = 4$ and $s \in \{2, 3, 4\}$, the maximum r-robust and t-hereditary s-clubs on WS graphs are trivial (singletons). For Gendreau instances, the average running times for the MRCP and the MHCP are 0.9 and 1.06 seconds, respectively. Detailed numerical results including optimal objective values and wall-clock running times are presented in Tables 18–21 in Section 7 of the online appendix.

Table 7. Comparison of the Best Objective Values Found Using the BCUT Solver for the MRCP and MHCP and by Directly Solving the Formulation for the Maximum r-Distinct-Path s-Club Problem Proposed by Veremyev and Boginski (2012)

			Best objective			Wall-clock time		Best objective			Wall-clock time	
Graph	n	m	Robust	Hereditary	Distinct paths	Robust	Hereditary	Robust	Hereditary	Distinct paths	Robust	Hereditary
			$r=t=2, s=3$					$r=t=2, s=4$				
karate	34	78	21	21	22	0.01	0.07	26	26	33	0.01	0.07
dolphins	62	159	22	22	24	0.04	0.03	32	32	36	0.01	0.01
lesmis	77	254	35	35	49	0.00	0.00	51	51	65	0.01	0.01
polbooks	105	441	39	39	46	0.03	0.03	58	58	64	0.11	0.08
adjnoun	112	425	63	63	73	0.04	0.04	94	94	104	0.08	0.06
football	115	613	40	40	43	0.70	0.60	113	115	115	0.32	0.02
jazz	198	2,742	158	158	165	0.06	0.06	186	186	192	0.10	0.11
celegans	453	2,025	234	234	353	0.16	0.18	378	378	429	0.69	0.77
email	1,133	5,451	138	138	168	403.02	296.03	505	505	≥582	42.86	37.72
polblogs	1,490	16,715	672	672	715	1.36	1.30	1,000	1,000	≥1,000	18.59	18.74
netscience	1,589	2,742	24	24	36	0.32	0.10	29	29	68	0.10	0.01
power	4,941	6,594	17	17	22	0.32	0.33	29	29	41	0.81	0.84
hep-th	8,361	15,751	52	52	76	16.80	9.99	177	177	≥177	66.77	80.39
PGP	10,680	24,316	239	239	≥239	1.12	1.14	446	446	≥446	34.73	30.67
			$r=t=3, s=3$					$r=t=3, s=4$				
karate	34	78	11	11	19	0.01	0.01	13	13	32	0.05	0.39
dolphins	62	159	14	17	23	0.06	0.00	24	24	34	0.11	0.68
lesmis	77	254	25	25	45	0.00	0.00	34	34	64	0.01	0.17
polbooks	105	441	31	31	41	0.05	0.06	44	44	60	0.21	0.69
adjnoun	112	425	47	47	67	0.03	0.04	81	81	101	0.14	0.36
football	115	613	27	27	36	0.89	1.24	99	103	115	0.96	1.42
jazz	198	2,742	145	145	162	0.06	0.06	181	181	191	0.32	0.46
celegans	453	2,025	141	141	321	0.16	0.18	291	291	426	3.23	3.69
email	1,133	5,451	88	88	130	**9.09%**	**11.49%**	407	407	558	83.68	87.10
polblogs	1,490	16,715	605	605	677	1.58	1.72	913	913	≥913	22.50	22.27
netscience	1,589	2,742	21	21	35	0.02	0.02	21	21	63	0.16	0.34
power	4,941	6,594	12	12	17	0.02	0.02	17	17	37	0.07	0.52
hep-th	8,361	15,751	38	38	65	0.71	0.87	109	109	≥109	174.38	262.87
PGP	10,680	24,316	170	170	251	0.42	0.44	308	308	≥308	14.29	19.30
			$r=t=4, s=3$					$r=t=4, s=4$				
karate	34	78	9	9	16	0.01	0.07	10	10	31	0.00	0.01
dolphins	62	159	7	7	20	0.13	0.17	17	17	33	0.06	0.06
lesmis	77	254	21	21	40	0.00	0.00	25	25	63	0.00	0.01
polbooks	105	441	24	24	40	0.03	0.04	35	35	59	0.06	0.06
adjnoun	112	425	31	32	64	0.14	0.11	67	67	100	0.47	0.49
football	115	613	17	17	30	1.34	1.41	65	65	115	7.23	13.07
jazz	198	2,742	136	136	158	0.06	0.08	174	174	191	0.51	0.55
celegans	453	2,025	99	99	295	0.13	0.13	207	207	424	5.38	3.08
email	1,133	5,451	66	66	116	1,130.71	2,563.28	340	340	≥522	92.21	83.22
polblogs	1,490	16,715	557	558	657	1.85	1.79	852	852	≥852	50.94	47.81
netscience	1,589	2,742	20	20	33	0.01	0.21	20	20	56	0.06	0.02
power	4,941	6,594	12	12	16	0.00	0.00	13	13	35	0.00	0.00
hep-th	8,361	15,751	32	32	51	0.18	0.20	70	70	≥70	118.55	98.71
PGP	10,680	24,316	124	124	229	0.41	0.45	227	227	≥227	12.72	13.36

Notes. Wall-clock running times (in seconds) are also reported for the BCUT solver. If an instance was not solved to optimality under the time limit, the optimality gap is reported (highlighted in bold). An entry of 0.00 means the run took less than 0.005 seconds.

7. Conclusion

The r-robust s-club and t-hereditary s-club models formalize the notion of fault-tolerance in s-clubs, a desirable property when seeking reliable low-diameter clusters. In this article, we establish the NP-hardness of the associated optimization problems on arbitrary and restricted graph classes for parameters $r, t, s \geq 2$. Furthermore, we show that it is NP-complete to verify if a vertex subset is an r-robust s-club when $r \geq 2$ is fixed and s is part of the input, and so is its counterpart for fixed $s \geq 5$ and r part of the input. We also show that it is coNP-complete to verify if a vertex subset is a t-hereditary s-club when $s \geq 5$ is fixed and t is a part of the input.

We propose cut-like formulations for the MRCP for $s \in \{2, 3, 4\}$ and the MHCP for every integer $s \geq 2$ based on length-bounded vertex separators. This is the first

IP formulation of the maximum r-robust 4-club problem to appear in the literature. For $s \in \{2,3,4\}$, we establish the polynomial-time solvability of the associated separation problems. For each $s \geq 5$, we show that it is coNP-complete to determine whether a given solution satisfies all length-bounded vertex separator inequalities used to formulate the MHCP.

We introduce a graph decomposition approach based on finding maximal biconnected components (blocks) that enables us to solve the IP on several smaller subgraphs of the input graph. This block decomposition algorithm is recursive and incorporates preprocessing techniques based on a heuristic solution that aims to further reduce the size of the subgraph on which the IP is solved. We also propose lower and upper bounds on the number of length-bounded vertex-disjoint paths between any given a pair of vertices, which enables us to avoid the exact computation of this quantity used frequently in several steps of the overall algorithm. We devise a decomposition BC algorithm to solve the cut-like IP formulations of the MRCP and the MHCP when $s \in \{2,3,4\}$. The computational gains are empirically evaluated on a test bed of real-life instances from the Tenth DIMACS Implementation Challenge. Our computational studies include the first reported numerical results for the MRCP and the MHCP when $s \in \{3,4\}$.

This line of research could be continued by targeting the cases involving $s \geq 5$, where the length-bounded counterpart of Menger's theorem no longer applies. Our results concerning the unlikelihood of convenient IP formulations of the MRCP and the MHCP for $s \geq 5$ can be informative in that regard. The foundations laid in this article, such as the recursive block decomposition algorithm, can be helpful to any exact approach to these problems.

Acknowledgment

The authors thank Alexander Veremyev for sharing test instances, codes for graph visualization, and solvers for the maximum r-robust 2-club problem.

Endnote

[1] A cut-vertex and a bridge and its end-vertices are considered biconnected subgraphs.

References

Alba RD (1973) A graph-theoretic definition of a sociometric clique. *J. Math. Sociol.* 3(1):113–126.

Almeida MT, Carvalho FD (2014) An analytical comparison of the LP relaxations of integer models for the *k*-club problem. *Eur. J. Oper. Res.* 232(3):489–498.

Alon U (2007) Network motifs: theory and experimental approaches. *Nature Rev. Genetics* 8:450–461.

Asahiro Y, Miyano E, Samizo K (2010) Approximating maximum diameter-bounded subgraphs. López-Ortiz A, ed. *LATIN 2010: Theoretical Informatics* (Springer, Berlin), 615–626.

Asahiro Y, Doi Y, Miyano E, Samizo K, Shimizu H (2018) Optimal approximation algorithms for maximum distance-bounded subgraph problems. *Algorithmica* 80(6):1834–1856.

Bader DA, Meyerhenke H, Sanders P, Wagner D, eds. (2013) *Graph Partitioning and Graph Clustering: Tenth Dimacs Implementation Challenge Workshop*, February 13–14, Georgia Institute of Technology, Atlanta, GA. Contemporary Mathematics 588 (American Mathematical Society and Center for Discrete Mathematics and Theoretical Computer Science, Providence, RI).

Baier G, Erlebach T, Hall A, Köhler E, Kolman P, Pangrác O, Schilling H, et al. (2010) Length-bounded cuts and flows. *ACM Trans. Algorithms* 7(1):4.

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

Batagelj V, Zaveršnik M (2011) Fast algorithms for determining (generalized) core groups in social networks. *Adv. Data Anal. Classification* 5:129–145.

Botton Q, Fortz B, Gouveia L (2015) On the hop-constrained survivable network design problem with reliable edges. *Comput. Oper. Res.* 64:159–167.

Botton Q, Fortz B, Gouveia L, Poss M (2013) Benders decomposition for the hop-constrained survivable network design problem. *INFORMS J. Comput.* 25(1):13–26.

Bourjolly JM, Laporte G, Pesant G (2000) Heuristics for finding *k*-clubs in an undirected graph. *Comput. Oper. Res.* 27:559–569.

Bourjolly JM, Laporte G, Pesant G (2002) An exact algorithm for the maximum *k*-club problem in an undirected graph. *Eur. J. Oper. Res.* 138:21–28.

Caccetta L (1979) On extremal graphs with given diameter and connectivity. *Ann. New York Acad. Sci.* 328(1):76–94.

Carmesin J, Diestel R, Hamann M, Hundertmark F (2014) *k*-blocks: A connectivity invariant for graphs. *SIAM J. Discrete Math.* 28(4):1876–1891.

Carr RD, Lancia G (2002) Compact vs. exponential-size LP relaxations. *Oper. Res. Lett.* 30(1):57–65.

Carvajal R, Constantino M, Goycoolea M, Vielma JP, Weintraub A (2013) Imposing connectivity constraints in forest planning models. *Oper. Res.* 61(4):824–836.

Chang MS, Hung LJ, Lin CR, Su PC (2013) Finding large *k*-clubs in undirected graphs. *Computing* 95(9):739–758.

Chen YP, Liestman AL, Liu J (2005) Clustering algorithms for ad hoc wireless networks. Pan Y, Xiao Y, eds. *Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing* (Nova Science Publishers, New York), 145–164.

Cook DJ, Holder LB (2006) *Mining Graph Data* (John Wiley & Sons, Hoboken, NJ).

Dagum L, Enon R (1998) OpenMP: An industry standard API for shared-memory programming. *IEEE Comput. Sci. Engrg.* 5(1):46–55.

Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. *Math. Programming* 91(2):201–213.

Faudree RJ, Gould RJ, Powell JS (2012) Property $P_{d,m}$ and efficient design of reliable networks. *Networks* 60(3):167–178.

Ford L Jr, Fulkerson DR (1956) Maximal flow through a network. *Canadian J. Math.* 8:399–404.

Gendreau M, Soriano P, Salvail L (1993) Solving the maximum clique problem using a tabu search approach. *Ann. Oper. Res.* 41(4):385–403.

Golovach PA, Heggernes P, Kratsch D, Rafiey A (2014) Finding clubs in graph classes. *Discrete Appl. Math.* 174:57–65.

Gould N, Scott J (2016) A note on performance profiles for benchmarking software. *ACM Trans. Math. Software* 43(2):1–5.

Gouveia L, Leitner M (2017) Design of survivable networks with vulnerability constraints. *Eur. J. Oper. Res.* 258(1):89–103.

Grötschel M, Lovász L, Schrijver A (1993) *Geometric Algorithms and Combinatorial Optimization*, 2nd ed. (Springer-Verlag, Berlin).

Grötschel M, Monma CL, Stoer M (1992) Facets for polyhedra arising in the design of communication networks with low-connectivity constraints. *SIAM J. Optim.* 2(3):474–504.

Gupta G, Younis M (2003) Fault-tolerant clustering of wireless sensor networks. *Proc. IEEE Wireless Communications and Networking* (IEEE, New York), 1579–1584.

Gurobi Optimization L (2020) Gurobi optimizer reference manual. Accessed August 29, 2022, http://www.gurobi.com.

Hartung S, Komusiewicz C, Nichterlein A, Suchý O (2015) On structural parameterizations for the 2-club problem. *Discrete Appl. Math.* 185:79–92.

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

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

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

Langguth J, Manne F, Sanders P (2010) Heuristic initialization for bipartite matching problems. *J. Experiment. Algorithmics* 15:1–1.

Lawler E (1976) *Combinatorial Optimization: Networks and Matroids* (Holt, Rinehart, and Winston, New York).

Lewis JM, Yannakakis M (1980) The node-deletion problem for hereditary properties is NP-complete. *J. Comput. System Sci.* 20(2):219–230.

Lovász L, Neumann-Lara V, Plummer M (1978) Mengerian theorems for paths of bounded length. *Period Math. Hungary* 9(4):269–276.

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

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

Magun J (1998) Greedy matching algorithms, an experimental study. *ACM J. Experiment. Algorithmics* 3:6.

Martin RK (1991) Using separation algorithms to generate mixed integer model reformulations. *Oper. Res. Lett.* 10(3):119–128.

Matula DW (1978) *k*-blocks and ultrablocks in graphs. *J. Combinatorial Theory Ser. B* 24(1):1–13.

Matula DW, Beck LL (1983) Smallest-last ordering and clustering and graph coloring algorithms. *J. ACM* 30(3):417–427.

Menger K (1927) Zur allgemeinen kurventheorie. *Fundamentals Math.* 10:95–115.

Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U (2002) Network motifs: Simple building blocks of complex networks. *Science* 298:824–827.

Möhring R, Müller-Hannemann M (1995) Cardinality matching: Heuristic search for augmenting paths. Technical report 439, Technische Universität, Berlin.

Mokken RJ (1979) Cliques, clubs and clans. *Qualitative Quantitative* 13(2):161–173.

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

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

Pattillo J, Youssef N, Butenko S (2013) On clique relaxation models in network analysis. *Eur. J. Oper. Res.* 226(1):9–18.

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

Salemi H, Buchanan A (2020) Parsimonious formulations for low-diameter clusters. *Math. Programming Comput.* 12(3):493–528.

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

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

Veremyev A, Boginski V (2012) Identifying large robust network clusters via new compact formulations of maximum *k*-club problems. *Eur. J. Oper. Res.* 218(2):316–326.

Veremyev A, Boginski V, Pasiliao EL, Prokopyev OA (2022) On integer programming models for the maximum 2-club problem and its robust generalizations in sparse graphs. *Eur. J. Oper. Res.* 297(1):86–101.

Vijayan K, Murty USR (1964) On accessibility in graphs. *Sankhya Ser. A* 26(2–3):299–302.

Walteros JL, Buchanan A (2020) Why is maximum clique often easy in practice? *Oper. Res.* 68(6):1866–1895.

Wang Y, Buchanan A, Butenko S (2017) On imposing connectivity constraints in integer programs. *Math. Programming* 166(1):241–271.

Watts D, Strogatz S (1998) Collective dynamics of "small-world" networks. *Nature* 393:440–442.

West D (2001) *Introduction to Graph Theory* (Prentice-Hall, Upper Saddle River, NJ).

Yezerska O, Pajouh FM, Butenko S (2017) On biconnected and fragile subgraphs of low diameter. *Eur. J. Oper. Res.* 263(2): 390–400.

Zhang LV, King OD, Wong SL, Goldberg DS, Tong AH, Lesage G, Andrews B, et al. (2005) Motifs, themes and thematic maps of an integrated saccharomyces cerevisiae interaction network. *J. Biology* 4(2):6.

Zheng J, Wang P, Li C, Mouftah HT (2008) An efficient fault-prevention clustering protocol for robust underwater sensor networks. *Proc. IEEE Internat. Conf. on Comm.*, 2802–2807.