Interdicting Low-Diameter Cohesive Subgroups in Large-Scale Social Networks

Niloufar Daemi, Juan S. Borrero, Balabhaskar Balasundaram
School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK 74078,
niloufar.daemi@okstate.edu, juan.s.borrero@okstate.edu, baski@okstate.edu

The *s-clubs* model cohesive social subgroups as vertex subsets that induce subgraphs of diameter at most s. In defender-attacker settings, for low values of s, they can represent tightly-knit communities whose operation is undesirable for the defender. For instance, in online social networks, large communities of malicious accounts can effectively propagate undesirable rumors. In this article we consider a defender that can disrupt vertices of the adversarial network to minimize its threat, which leads us to consider a maximum s-club interdiction problem where interdiction is penalized in the objective function. Using a new notion of H-heredity in s-clubs, we provide a mixed-integer linear programming formulation for this problem that uses far fewer constraints than the formulation based on standard techniques. We show that the linear programming relaxation of this formulation has no redundant constraints and identify facets of the convex hull of integral feasible solutions under special conditions. We further relate H-heredity to latency-s connected dominating sets and design a decomposition branch-and-cut algorithm for the problem. Our implementation solves benchmark instances with more than 10,000 vertices in a matter of minutes and is orders of magnitude faster than algorithms based on the standard formulation.

Key words: network interdiction, clubs, integer programming

1. Motivation

Cohesive subgroups in social networks can represent groups of individuals that share core beliefs, influence each other, and act together as a unit towards a common goal (Wasserman and Faust 1994). In more general networks, cohesive subgroup models provide formalizations of "tightly-knit clusters" (Balasundaram et al. 2011) and therefore have been used in applications beyond social network analysis, for example, to analyze complex biological networks (Pasupuleti 2008, Butenko and Wilhelm 2006, Balasundaram et al. 2005). The canonical optimization problem of identifying a particular type of clique relaxation of maximum cardinality (or weight) has received considerable attention in the literature (Pattillo et al. 2013, Balasundaram and Pajouh 2013) . The focus of this article is on the interdiction of a clique relaxation that models low-diameter clusters.

As a motivating example, consider the following stylized scenario. Suppose a social media network manager (NM) recognizes that disinformation is being spread with hashtags #badrumor and #fakenews, and suspects that a coordinated group of adversarial actors whose identities are unknown may be responsible. Although the NM could ban or deactivate accounts, it would not be effective to do so arbitrarily. The NM can consider the following graph model to capture this situation, let us refer to it as the *rumor graph*: the vertex set would include all user accounts using one of the offending hashtags in their posts; the edge set would include an edge $\{u,v\}$ if account u liked or reshared a post by account v that included one of the offending hashtags. We use an undirected edge to indicate that the accounts represented by the end-points are related, and not necessarily that one is *directing* the other. Under the assumption that the interaction patterns of such suspicious accounts in the rumor graph resembles a cohesive social subgroup that is capable of quick communication, one could arguably phrase the NM's decision problem as one of optimally interdicting (by disabling accounts) all large low-diameter cohesive subgroups in the rumor graph.

Although we are describing a stylized version of the decision problem faced by the NM, it can be a reasonable first step in analyzing such problems to devise effective interdiction policies in practice. To begin with, we choose to model cohesive subgroups of interest in this rumor network as *s-clubs* for low values of parameter s that ensure short pairwise distances inside the cohesive subgroup between members as a surrogate for quick communication between group members. We are also assuming that one of the maximum cardinality s-clubs contains the adversarial accounts and that diminishing its size can impact that group's ability to spread disinformation. Furthermore, the other maximum cardinality s-clubs (those not containing the adversarial actors) are unwittingly helping with the spread of the rumor and arguably also warrant deactivation.

1.1. Background and Notations

Consider an n-vertex graph $G=(V,E)$ with vertex set $V := \{1,\dots,n\}$ and edge set $E \subseteq \binom{V}{2} := \{\{u,v\} \mid u,v \in V, v \neq u\}$. We will assume throughout that G is not an empty graph, i.e., $E \neq \emptyset$. Denote by $N_G(v) := \{u \in V \mid \{u,v\} \in E\}$, the set of neighbors of vertex v and its cardinality by $\deg_G(v)$. We also use the notation $N_G[v] := N_G(v) \cup \{v\}$ to denote the *closed* neighborhood of a vertex v in G. We denote the subgraph induced by a set of vertices $S \subseteq V$ by $G[S] := \left(S, \binom{S}{2} \cap E\right)$. For convenience, we denote the deletion of a set of vertices T and incident edges as $G \setminus T := G[V \setminus T]$.

We use the following notations to describe concepts related to distances in graphs. Let $\text{dist}_G(u,v)$ denote the length of the shortest path between a connected pair for vertices u and v in G, where the length of a path is the number of edges in the path. The diameter of a connected graph is the maximum distance between a pair of vertices and we denote it by $\text{diam}(G) := \max\{\text{dist}_G(u,v) \mid u,v \in V\}$. If u and v are in different connected components of G, then the distance between them, and hence the diameter of that disconnected graph are taken to be infinite. When the graph G under consideration is known without any ambiguity, we drop the subscript G for convenience from all the notations.

Definition 1 (Mokken (1979); cf. Balasundaram et al. (2005)). Given a graph $G=(V,E)$ and a positive integer s, we call a subset of vertices $S \subseteq V$ an *s-club* if $\text{diam}(G[S]) \leq s$.

The largest cardinality of an s-club is called *the s-club number* of graph G, denoted by $\bar{\omega}_s(G)$. Detecting a maximum cardinality s-club, i.e., *the maximum s-club problem*, is NP-hard in general (Bourjolly et al. 2002) and in graphs of diameter $s+1$ (Balasundaram et al. 2005). The model is one of several types of clique relaxations that have been studied in the literature (Pattillo et al. 2013), as it reduces to a clique when $s=1$. With $s=2$, we obtain a formalization of the friend-of-a-friend cluster, as a 2-club S must satisfy at least one of the following conditions for every distinct pair of vertices $u,v \in S$: either $\{u,v\} \in E$, or $N_G(u) \cap N_G(v) \cap S \neq \emptyset$. In other words, every pair of members of a 2-club are either friends or they have a mutual friend in the group. In general, s-clubs for low values of parameter $s \in \{2,3\}$, can be used to represent clusters where quick communication between members is possible.

Interdiction by vertex deletion is the focus of this article. Suppose $T \subset V$ is the "deletion set." A fundamental difference between interdicting cliques in a graph (Furini et al. 2019) versus s-clubs in a graph is *heredity.* If $K \subset V$ is a clique in G then $K \setminus T$ is a clique in $G \setminus T$ because the clique property is preserved under vertex deletion. However, the s-club property is not hereditary under vertex deletion; see Figure 1 (Alba 1973). Consequently, if S is an s-club in G, we cannot claim that $S \setminus T$ is an s-club in $G \setminus T$ for every $T \subseteq S$. This fundamental difference drives all of the approaches taken in this article to model and solve the s-club interdiction problem, and differentiates it from the techniques recently proposed for interdicting cliques (Furini et al. 2019).

Typically, interdiction comes "at a cost." If there were no restrictions on T, the entire graph can be deleted. Shortest path and other network flow interdiction problems are often motivated by applications that justify using a budget b in a constraint that says the size of T cannot exceed b (Morton et al. 2007, Pan et al. 2003, Israeli and Wood 2002). The budget in these settings is derived from physical restrictions such as the number of patrol vehicles available to intercept smugglers, or the number of sensors that can be deployed in the network for monitoring purposes. In our setting, we avoid the use of a hard budget constraint as the NM can delete any number of vertices (e.g., by banning or temporarily disabling user accounts) and may be willing to delete a large number of accounts to stem the rumor (Spangler 2018). However, if there is no cost incurred by deleting vertices, we set up a pointless and trivial problem that would suggest deleting V. Our focus is on identifying and deleting "club-critical vertices" and we assume that we incur an interdiction penalty in doing so, as opposed to a hard budget constraint.

1.2. Problem Statement

We wish to solve the following optimization problem to find an *optimal interdiction policy*, that is, a subset of vertices T^* that achieves the following minimum:

$$\min_{T \subseteq V} \left\{ \bar{\omega}_s(G \setminus T) + \alpha |T| \right\}, \tag{1}$$

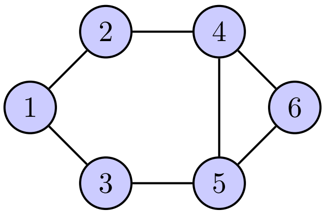

Figure 1 **The set $S=\{1,2,3,4,5\}$ is a 2-club. After deleting any vertex $i\in S$, the set $S\setminus\{i\}$ will not be a 2-club.**

where $\alpha > 0$ is the unit penalty cost of deleting a vertex. We could interpret this choice of penalty as follows. As the empty set is a feasible solution to problem (1), we have:

$$\begin{aligned} &\bar{\omega}_s(G\setminus T^*)+\alpha|T^*|\le\bar{\omega}_s(G)\\ \implies &\frac{\bar{\omega}_s(G)-\bar{\omega}_s(G\setminus T^*)}{|T^*|}\ge\alpha,\ \text{assuming } T^*\neq\emptyset. \end{aligned}$$

The ratio of the decrease in the s-club number upon interdiction to the size of an optimal deletion set (when non-empty) is at least α. In our models, we typically choose $\alpha\in\bigcup_{k\in\mathbb{N}}\{k,\frac{1}{k}\}$. By setting $\alpha=k$, the NM can use an operating policy that requires the s-club number decreases by at least k for each vertex deleted. In settings where we are prepared to delete a large number of vertices to decrease the s-club number, we can delete up to k times the decrease that we can produce by setting $\alpha=1/k$.

1.3. Our Contributions

In this paper we focus on solving problem (1) using mixed-integer linear programming (MILP) approaches. We first present an MILP formulation for the problem using standard interdiction techniques. The formulation has important drawbacks from an implementation perspective, which leads us to introduce the concept of *H-heredity* in s-clubs and study its implication for formulating and solving problem (1). We derive several interesting properties of the constraints based on this concept and use them to derive a more compact MILP reformulation of problem (1).

Specifically, this paper makes the following contributions to the literature of adversarial community disruptions, specifically, interdiction by deletion of vertices in maximum cardinality s-clubs.

(a) We introduce the new concept of *H-hereditary s-clubs*, which extends the notion of heredity to s-clubs. Based on H-heredity, we introduce an MILP formulation of the s-club interdiction problem that has fewer constraints than the naive MILP formulation that is based on standard interdiction formulation techniques.

(b) We show that the LP relaxation of the proposed formulation does not have redundant constraints. We also derive three types of facet defining inequalities for the convex hull of feasible solutions by further strengthening the new constraints based on H-heredity for special s-clubs.

(c) We establish a one-to-one correspondence between the sets inducing H-heredity in an s-club and latency-s connected dominating sets (latency-s CDSs) of the s-club (Validi and Buchanan 2020). We exploit this relationship in a decomposition branch-and-cut algorithm based on delayed constraint generation. This approach is able to solve several real-life and synthetic instances of the interdiction problem with more than 10,000 vertices in a matter of minutes. Moreover, our approach solves the problem orders of magnitude faster than using an analogous algorithm based on the naive MILP formulation.

The remainder of the paper is organized as follows. In Section 2 we present the naive MILP formulation of the interdiction problem. Section 3 introduces the concept of H-heredity and presents the proposed MILP reformulation developed based on this concept. In Section 4 we discuss the relationship between s-club interdiction and latency-s CDSs and in Section 5 we describe our decomposition algorithm. Section 6 reports the results of our numerical experiments and in Section 7 we present our conclusions. The proofs of all the technical results are included in Appendix A and additional experiments and results are discussed in Appendix B.

2. A Preliminary Formulation

An MILP formulation of problem (1) can be derived by using standard techniques in interdiction (Fischetti et al. 2018, 2019, Smith and Song 2020). To this end, we use vectors $x \in \{0,1\}^{|V|}$ as incidence vectors of a deletion set, thus $x_v = 1$ if v is deleted and zero otherwise. We let T^x denote the set of vertices deleted in solution x, thus $T^x = \{v \in V \mid x_v = 1\}$. Henceforth, we also use the convenient short form $x(S)$ in place of $\sum_{v\in S} x_v$ for $S \subseteq V$. In terms of x, problem (1) is given by:

$$z_{s,\alpha} = \min \left\{ \bar{\omega}_s(G \setminus T^x) + \alpha x(V) \,\middle|\, x \in \{0,1\}^{|V|} \right\}. \tag{2}$$

The bilevel optimization problem (2) can be reformulated as the following single-level MILP:

$$\begin{aligned} z_{s,\alpha} = \min\ & \theta + \alpha x(V) && \text{(3a)}\\ s.t.\quad & \theta \geq |S| - |S|x(S) \qquad \forall S \in \mathcal{S} && \text{(3b)}\\ & x \in \{0,1\}^{|V|}, \theta \in \mathbb{R}_+, && \text{(3c)} \end{aligned}$$

where $\mathcal{S}$ is the collection of all s-clubs in G. The right-hand side of constraint (3b) becomes redundant if a vertex in S is interdicted. Otherwise, the cardinality of the maximum s-club in the interdicted graph $G \setminus T^x$ should be at least $|S|$.

Although valid, the direct implementation of formulation (3) in an MILP solver is untenable for large instances as it requires enumerating exponentially many s-clubs in G in the worst case. Nevertheless, this formulation can be used in a delayed constraint generation algorithm as follows.

Let $\mathcal{S}^0 \subseteq \mathcal{S}$ be an initial collection of s-clubs. In iteration $i = 0, 1, \ldots$, the algorithm solves the master relaxation problem:

$$\min \left\{ \theta + \alpha x(V) \;\middle|\; \theta \geq |S| - |S| x(S) \;\; \forall S \in \mathcal{S}^i, x \in \{0,1\}^{|V|}, \theta \in \mathbb{R}_+ \right\} \tag{4}$$

and recovers an optimal solution (θ^i, x^i). If $\theta^i \geq \bar{\omega}_s(G \setminus T^{x^i})$ then it follows that (θ^i, x^i) is an optimal solution of problem (3), and the algorithm terminates. Otherwise, the algorithm identifies an s-club S' in $G \setminus T^{x^i}$ such that $|S'| > \theta^i$ and updates $\mathcal{S}^{i+1} := \mathcal{S}^i \cup \{S'\}$.

Clearly, this delayed constraint generation algorithm converges to an optimal solution of problem (3) in a finite number of steps because $\mathcal{S}$ is a finite set. However, it has two important limitations: First, each iteration requires solving an MILP master relaxation and the separation problem involves solving the NP-hard maximum s-club problem in the interdicted graph. Second, constraint (3b) will become redundant "easily" if any vertex in S is interdicted. (Contrast this with the clique interdiction counterpart studied by Furini et al. (2019); if S were a clique in G, the constraint would say $\theta \geq |S| - x(S)$ as the clique property is hereditary under vertex deletion.) This behavior can result in weak LP relaxations and it is exacerbated in the presence of numerous "nearly" identical s-clubs, each requiring the addition of a distinct constraint of the form (3b) to the master problem. There is empirical evidence that having a large number of similar s-clubs can be very detrimental for such delayed constraint generation approaches from a computational perspective, especially when the generated constraint is arguably not very strong (Lu et al. 2018, Moradi and Balasundaram 2018). In the following, we develop techniques that help alleviate the aforementioned concerns by exploiting graph-theoretic properties of s-clubs.

3. Exploiting Heredity in s-Clubs

In this section, we discuss an alternative formulation for the s-club interdiction problem that addresses the issues that arise from using constraints (3b) in a delayed constraint generation framework. The formulation is based on the observation that removing vertices of an s-club does not necessarily imply that the remaining vertices do not form an s-club. In other words, the formulation exploits the fact that some s-clubs can be *partially hereditary* in the following sense.

Definition 2. Given a graph $G = (V, E)$, an s-club S in G, and $H \subseteq S$, we say that S is an *H-hereditary s-club* if $\operatorname{diam}(G[S \setminus T]) \leq s$ for every $T \subseteq H$.

Observe that every s-club is trivially $\emptyset$-hereditary. Furthermore, an s-club S could be simultaneously H-hereditary and J-hereditary where J and H are incomparable subsets of S. Therefore, we are only interested in H-hereditary s-clubs of S for which H is maximal with respect to inclusion of vertices from $S \setminus H$. Given an H-hereditary s-club S, we refer to the partition $\{H, S \setminus H\}$ as the H-partition of S and it is said to be non-trivial if $H \neq \emptyset$. Figure 2 illustrates this idea using 2-clubs. Pertinently, an s-club S can be S-hereditary (i.e., truly hereditary) if and only if S is a clique.

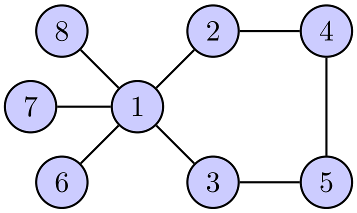

Figure 2 **The set $\hat{S} = \{1,2,3,4,5\}$ is a 2-club that does not admit any non-trivial H-partitions. The 2-club $\tilde{S} = \{1,2,3,6,7,8\}$ on the other hand is $\tilde{H}$-hereditary with $\tilde{H} = \{2,3,6,7,8\}$.**

3.1. Alternate Formulation Using Hereditary s-Clubs

Given an H-hereditary s-club S, define the following set:

$$\Lambda(S,H) := \Big\{(\theta,x) \in \mathbb{R}_+ \times \{0,1\}^{|V|} \mid \theta \geq |S| - x(H) - |S|x(S \setminus H)\Big\}, \tag{5}$$

and the following collection of subsets of vertices:

$$\mathcal{C}(S,H) := \{S \setminus T \mid T \subseteq H\}. \tag{6}$$

In other words, $\mathcal{C}(S,H)$ is the collection of all s-clubs generated from S by deleting every possible subset of H and $\mathcal{C}(S,\emptyset) = \{S\}$. The following two lemmas provide the elements that help us improve Formulation (3).

Lemma 1. *Let S be an H-hereditary s-club and suppose that $(\theta,x) \in \Lambda(S,H)$. Then (θ,x) satisfies the following inequalities:*

$$\theta \geq |U| - |U|x(U) \quad \forall U \in \mathcal{C}(S,H). \tag{7}$$

Based on Lemma 1, when we have two s-clubs U and S such that $U \in \mathcal{C}(S,H)$, we can replace constraint (3b) corresponding to U by the constraint defining the set $\Lambda(S,H)$ in (5) without compromising the correctness of formulation (3). Hence, $|\mathcal{C}(S,H)|$ constraints of type (3b) can be replaced by a single constraint. For example, the 2-club $\tilde{S} = \{1,2,3,6,7,8\}$ in Figure 2 is $\tilde{H}$-hereditary for $\tilde{H} = \{2,3,6,7,8\}$. Hence, we can replace constraints (3b) corresponding to all 2-clubs obtained by deleting subsets of $\tilde{H}$ by the single constraint $\theta \geq |\tilde{S}| - x(\tilde{H}) - |\tilde{S}|x(\tilde{S} \setminus \tilde{H})$.

Remark 1. It is important to contrast the aforementioned discussion against *incorrectly* reformulating (3) using $\Lambda(S,H)$-type constraints only for s-clubs that are maximal by inclusion. For example, consider the 2-clubs $\hat{S} = \{1,2,3,4,5\}$ and $\hat{U} = \{2,4,5\}$ in Figure 2. Although, $\hat{U} \subset \hat{S}$, we know that $\hat{U} \notin \mathcal{C}(\hat{S},H)$ for any non-empty $H \subseteq \hat{S}$ because $\hat{S}$ does not admit a non-trivial hereditary partition. Therefore, the omission of the constraint $\theta \geq |\hat{U}| - |\hat{U}|x(\hat{U})$ from the formulation would be a mistake because the resulting objective value of the solution defined by $x_v = 1$ for all $v \in V \setminus \hat{U}$ and $x_v = 0$ for all $v \in \hat{U}$ would be zero, rather than the correct objective value of $|\hat{U}|$.

The notion of H-heredity leads us to consider the following in regards to the strength of the $\Lambda(S,H)$-inequality. If the same s-club S is also J-hereditary, we obtain a different $\Lambda(S,J)$-inequality that is also valid. Is there a particular choice of H that makes the resulting constraint tighter? In this case, maximality of H with respect to inclusion of vertices from S is the answer. Given an s-club S, we define the set $\mathcal{H}(S)$ as follows:

$$\mathcal{H}(S) \coloneqq \{H \subseteq S \mid S \text{ is an } H\text{-hereditary } s\text{-club}\}. \tag{8}$$

Because $\emptyset \in \mathcal{H}(S)$ for every s-club S in G, by our definition $\mathcal{H}(S)$ is always non-empty.

Lemma 2. *Let S be an s-club such that $H, J \in \mathcal{H}(S)$. If $J \subset H$ then $\Lambda(S,H) \subseteq \Lambda(S,J)$.*

Based on Lemmas 1 and 2, we can replace constraint (3b) for an s-club U with the tighter constraint defining $\Lambda(S,H)$ if $U \in \mathcal{C}(S,H)$, and we only require the constraint for $H \in \mathcal{H}(S)$ that is maximal with respect to inclusion of vertices from S, in order to preserve the correctness of the MILP formulation. However, it should be noted that even if the collection $\mathcal{H}(S)$ is limited only to maximal sets, there could be several such maximal elements (see example in Figure 3).

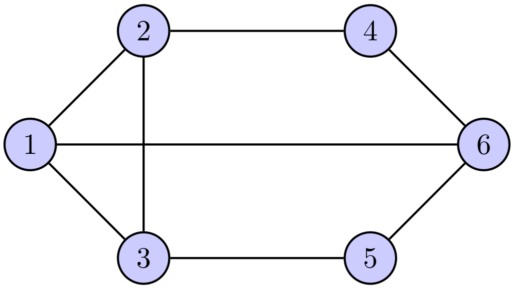

Figure 3 $H = \{1\}$ **and** $J = \{4,5\}$ **are maximal sets in** $\mathcal{H}(S) = \{\{1\},\{4\},\{5\},\{4,5\}\}$ **for the 2-club** $S = \{1,2,3,4,5,6\}$**.**

Let us define $\mathcal{H}^*(S)$ as the collection of maximal sets in $\mathcal{H}(S)$:

$$\mathcal{H}^*(S) \coloneqq \{H \in \mathcal{H}(S) \mid \text{there is no } J \in \mathcal{H}(S) \text{ such that } H \subset J\}. \tag{9}$$

Note that if S does not admit a non-trivial hereditary s-club description (e.g., $\hat{S}$ in Figure 2), $\mathcal{H}^*(S) = \mathcal{H}(S) = \{\emptyset\}$. We are now able to state the following result, which is an immediate consequence of the foregoing results and observations.

Lemma 3. *Given an s-club S in $G = (V,E)$, define $\mathcal{U}(S)$ as follows:*

$$\mathcal{U}(S) \coloneqq \bigcup_{H \in \mathcal{H}^*(S)} \mathcal{C}(S,H), \tag{10}$$

where $\mathcal{C}(S,H)$ is defined in (6)*, and define $\Lambda^*(S)$ as:*

$$\Lambda^*(S) := \bigcap_{H \in \mathcal{H}^*(S)} \Lambda(S,H). \tag{11}$$

If $(\theta,x) \in \Lambda^*(S)$, then $(\theta,x) \in \Lambda(S,H)$ for all $H \in \mathcal{H}(S)$ and, moreover, (θ,x) satisfies

$$\theta \geq |U| - |U|x(U) \qquad \forall U \in \mathcal{U}(S). \tag{12}$$

As a consequence of Lemma 3 we can replace all the constraints in Formulation (3) associated with all the s-clubs in $\mathcal{U}(S)$ by $|\mathcal{H}^*(S)|$ stronger constraints to obtain Formulation (14) described in Proposition 1 that follows. Depending on the particular s-club, such reduction in the number of constraints can be very significant as illustrated by the following remark.

Remark 2. A vertex v and its neighbors, the closed neighborhood $N_G[v]$, is an $N_G(v)$-hereditary s-club for every $s \geq 2$. Every possible subset of $N_G(v)$ is a deletion set T such that $N_G[v] \setminus T$ is an s-club, corresponding to exponentially many constraints in Formulation (3). These can all be replaced by a stronger constraint $\theta \geq \deg_G(v) + 1 - x(N_G(v)) - (\deg_G(v)+1)x_v$.

Proposition 1. *Define $\mathcal{C}^*$, the set of critical s-clubs in the graph $G = (V,E)$, as follows:*

$$\mathcal{C}^* = \{S \in \mathcal{S} \mid \textit{No } s\textit{-club } S' \supset S \textit{ exists such that } S \in \mathcal{U}(S')\}. \tag{13}$$

The following is an equivalent reformulation of problem (3)*:*

$$\begin{aligned} z_{s,\alpha} = \min\ & \theta + \alpha x(V) && && \text{(14a)}\\ s.t.\ \ & \theta \geq |S| - x(H) - |S|x(S \setminus H) && \forall H \in \mathcal{H}^*(S), \forall S \in \mathcal{C}^* && \text{(14b)}\\ & x \in \{0,1\}^{|V|}, \theta \in \mathbb{R}_+. && && \text{(14c)} \end{aligned}$$

Besides having significantly less constraints, Formulation (14) does not have redundancies in the sense that all constraints of the form (14b) are necessary in the description of the LP relaxation of (14); see Proposition 3 in Section 3.2.Two other questions that arise regarding Formulation (14) concern the strength of its LP relaxation and whether membership of an s-club in $\mathcal{C}^*$ is easily verifiable. Remark 3 that follows, shows that the LP relaxations of Formulations (3) and (14) are incomparable. (Hence, both formulations are investigated computationally in Section 6.) Proposition 2 that follows provides an alternate characterization of s-clubs in $\mathcal{C}^*$.

Remark 3. Let P and P' denote the LP relaxations of Formulations (14) and (3), respectively. There are instances where P is not contained in P' and vice versa. In general, for $s \geq 2$, neither

LP relaxation contains the other. To see that $P' \not\subseteq P$, consider an s-club $S \in \mathcal{C}^*$ and a non-empty $H \in \mathcal{H}^*(S)$ and construct the point $(\hat{\theta}, \hat{x})$ as follows:

$$\hat{x}_v = \begin{cases} 1, & \text{if } v \notin S \\ 0, & \text{if } v \in S \setminus H \\ 1/2, & \text{if } v \in H, \end{cases}$$

and $\hat{\theta} = |S| - |H|$. First we show that $(\hat{\theta}, \hat{x}) \in P'$. For any $U \in \mathcal{S}$, define $q(U,x) := |U|(1 - x(U))$, the right-hand side of constraint (3b). If $U \setminus S$ is not empty, then $q(U,\hat{x}) \leq 0$. On the other hand, if $U \subseteq S$, we have $q(U,\hat{x}) = |U|(1 - |U \cap H|/2)$. It follows that the maximum value of $q(U,\hat{x})$ over $U \in \mathcal{S}$ is $|S| - |H|$, achieved when $U = S \setminus H$. Hence, the point $(\hat{\theta}, \hat{x}) \in P'$. Furthermore, $(\hat{\theta}, \hat{x}) \notin P$ as it violates constraint (14b) for the chosen S and H when $|H| \geq 1$.

To see that $P \not\subseteq P'$, we consider a more specific counter-example applicable for any $s \geq 2$. Suppose that $G = (V,E)$ is a five-vertex star with center 1 and leaves {2,3,4,5}. In this case, $\mathcal{C}^* = \big\{V\big\}$ with $\mathcal{H}^*(V) = \big\{V \setminus \{1\}\big\}$. The LP relaxation of Formulation (14) becomes:

$$\begin{aligned} &\min \theta + \alpha x(V) \\ s.t. \quad &\theta \geq 5 - x_2 - x_3 - x_4 - x_5 - 5x_1, \\ &x \in [0,1]^5, \theta \geq 0. \end{aligned}$$

Consider the point $\bar{\theta} = 13/12$, $\bar{x}_1 = 1/3$, $\bar{x}_2 = 0$, $\bar{x}_3 = \bar{x}_4 = \bar{x}_5 = 1$. Observe that $(\bar{\theta}, \bar{x})$ belongs to P but does not belong to P' because Formulation (3) includes the constraint $\theta \geq 2(1 - x_1 - x_2)$ corresponding to the s-club $\{1,2\}$ that is violated by $(\bar{\theta}, \bar{x})$.

Remark 4. For non-empty H, constraint (14b) can be tightened using a smaller 'big-M' coefficient as $\theta \geq |S| - x(H) - (|S| - 1)x(S \setminus H)$ resulting in a valid formulation with a tighter LP relaxation. However, the conclusion of Remark 3 that the LP relaxations are incomparable continues to hold even using the modified constraint. This can be verified using the same counter-examples as in Remark 3. As this modification did not improve the computational performance significantly in our preliminary numerical experiments, we use constraint (14b) with the 'big-M' coefficient of $|S|$ for simplicity in the subsequent discussions and in our computational studies.

Another question of interest related to Formulation (14) is about the relationship between criticality of an s-club as defined in Proposition 1 and maximality of an s-club (by vertex inclusion). Proposition 2 we establish next shows that maximality is a stricter condition than criticality, that is, every maximal s-club is also a critical s-club although the converse is not true. Consider the example used earlier in Remark 1. The 2-club $\hat{S} = \{1,2,3,4,5\}$ in Figure 2 strictly contains the 2-club $\hat{U} = \{2,4,5\}$. The 2-club $\hat{S}$ is both critical and maximal, while $\hat{U}$ is clearly not maximal by inclusion. However, $\hat{U}$ is critical according to the definition in Proposition 1 because $\hat{S}$, which is the *unique* 2-club strictly containing $\hat{U}$, does not admit any non-trivial H-partitions. Indeed, criticality is equivalent to a weaker requirement that we refer to as *one-step maximality* for convenience.

Definition 3. We say that an s-club S in graph $G=(V,E)$ is one-step maximal if and only if $S\cup\{v\}$ is not an s-club for any $v\in V\setminus S$.

Observe that if an s-club is maximal then it is also one-step maximal, but the converse is not true. The 2-club $\hat{U}$ is one-step maximal but it is not maximal by inclusion in the conventional sense. It is also easy to see that for cliques and other hereditary properties, one-step maximality is equivalent to inclusionwise maximality.

Proposition 2. *Consider an s-club S in graph $G=(V,E)$. Then, $S\in\mathcal{C}^*$ if and only if S is one-step maximal.*

Although deciding if an s-club is maximal by inclusion is coNP-complete (Pajouh and Balasundaram 2012), Proposition 2 enables us to verify whether a given s-club S is critical in polynomial time. Nonetheless, using Formulation (14) directly is not expected to be computationally viable because it requires the enumeration of all s-clubs in $\mathcal{C}^*$ and their maximal hereditary partitions based on $\mathcal{H}^*(\cdot)$. Pertinently, given an s-club S, the complexity of enumerating $\mathcal{H}^*(S)$ or identifying a member in it is also unclear.

However, recall the discussion in Section 2 on a delayed constraint generation algorithm. In each iteration i, such a sequential cutting plane method would maintain a collection of s-clubs $\mathcal{S}^i\subset\mathcal{S}$ and for each $S\in\mathcal{S}^i$ it would also maintain collections $\widetilde{\mathcal{H}}(S)\subset\mathcal{H}(S)$. Then, the algorithm solves the following master relaxation MILP (compare with master problem (4)):

$$z^i_{s,\alpha}=\min_{\substack{x\in\{0,1\}^{|V|}\\ \theta\in\mathbb{R}_+}}\left\{\theta+\alpha x(V)\;\middle|\;\theta\geq|S|-x(H)-|S|x(S\setminus H)\ \forall S\in\mathcal{S}^i, H\in\widetilde{\mathcal{H}}(S)\right\}. \tag{15}$$

Denote the optimal solution found by (θ^i,x^i), we proceed similarly by identifying an s-club S' in the interdicted graph $G\setminus T^{x^i}$ such that $|S'|>\theta^i$, if it exists; otherwise, the solution is feasible and optimal. If found, an important difference is that now, instead of adding the constraint $\theta\geq|S'|-|S'|x(S')$, we will seek to identify a member $H'\in\mathcal{H}^*(S')$ (if that is not possible, find a member $H'\in\mathcal{H}(S')$). Then, we can add the constraint $\theta\geq|S'|-x(H')-|S'|x(S'\setminus H')$, update $\mathcal{S}^{i+1}$ with $\mathcal{S}^i\cup\{S'\}$, update $\widetilde{\mathcal{H}}(S')$ with $\widetilde{\mathcal{H}}(S')\cup\{H'\}$, and then re-solve the master relaxation. Alternately, we could add a round of constraints by enumerating multiple members of $\mathcal{H}(S')$. Nonetheless, the $\Lambda(S',H)$ inequality is violated by (θ^i,x^i) for every $H\in\mathcal{H}(S')$ as $x^i(S')=0$; recall that the s-club S' was found in the interdicted graph.

The foregoing discussion highlights the important considerations when separating $\Lambda(S,H)$-inequalities. In particular, how can we detect an $H\in\mathcal{H}^*(S)$? We address this question in Section 4. We close this section by discussing polyhedral properties of the LP relaxation and of the convex hull of feasible solutions of Formulation (14).

3.2. Facial Structure of Associated Polyhedra

First, we show that the LP relaxation of Formulation (14) has no redundant constraints, then we show three types of facets of the convex hull of the formulation based on maximal cliques, critical stars, and critical edge stars of G under an additional assumption of independence among some vertices in the s-club.

PROPOSITION 3. *Every constraint* (14b) *induces a facet of the LP relaxation polyhedron of* (14)*.*

The result in Proposition 3 indicates the importance of critical s-clubs in $\mathcal{C}^*$ (and the maximal members in $\mathcal{H}^*(S)$ for every critical s-club S) in formulating this problem. It further emphasizes the fact that no constraint of type (14b) is dominated by another of this type in the associated LP relaxation. This result also motivates the facets of the *convex hull* of feasible solutions to Formulation (14) we derive based on specially structured s-clubs. These results are presented next.

Let $\mathcal{P}$ denote the convex hull of the set of feasible solutions of Formulation (14). As it is to be expected, constraints (14b) do not yield facets of $\mathcal{P}$ in general because of the 'big-M' type constant $|S|$ in the constraint. To identify facets of $\mathcal{P}$, we begin with an inequality that is known to induce a facet of the clique interdiction counterpart. Furini et al. (2019) formulate the clique interdiction problem (with an interdiction budget instead of a penalty) using the following constraints:

$$\theta \geq |K| - x(K) \quad \forall K \in \mathcal{K}, \tag{16}$$

where $\mathcal{K}$ is the collection of all cliques in G. Because the clique property is hereditary, there is no need for a 'big-M' coefficient in constraint (16) to make the constraint redundant if a vertex in K is interdicted. Furini et al. (2019) further show that inequality (16) induces a facet of the convex hull of feasible solutions to their budget-constrained clique interdiction problem under suitable conditions, one of which is the maximality of clique K.

Cliques are s-clubs for every $s \geq 2$ and remain so if some vertices are interdicted. So the inequality (16) is valid for the s-club interdiction problem as well, and it is reasonable to ask if they induce facets when the clique K satisfies some additional requirement (like maximality). Next, we provide a result that generalizes these facets to s-clubs, for any $s \geq 2$.

For a given subset of vertices $Q \subseteq V$, let $\mathcal{P}^Q$ denote the face of the convex hull $\mathcal{P}$ in which the vertices of Q are not interdicted, that is, $\mathcal{P}^Q = \mathcal{P} \cap \{(\theta, x) \mid x_v = 0 \ \forall v \in Q\}$ (see Appendix A.7 regarding the "zero facets" of $\mathcal{P}$). One can consider $\mathcal{P}^Q$ as the convex hull of interest at a node of a branch-and-cut tree where the variables corresponding to Q have been fixed to zero. However, we are more interested in the case where $Q = S \setminus H$ for some H-hereditary s-club S when certain facets of $\mathcal{P}^Q$ can be readily derived, as shown next.

Theorem 1. *Let $S \in \mathcal{C}^*$ be an H-hereditary s-club. Then the following inequality is valid for $\mathcal{P}^{S\setminus H}$ and induces a facet of $\mathcal{P}^{S\setminus H}$ for any positive integer s:*

$$\theta \geq |S| - x(H). \tag{17}$$

Although H is not required to be a maximal member of $\mathcal{H}(S)$ for Theorem 1 to hold, it is relevant in the following sense. Such an inequality is valid (without lifting the variables in $S \setminus H$) only locally in the nodes of a branch-and-cut tree where the corresponding variables have been fixed to zero. It could therefore be argued that larger $H \in \mathcal{H}(S)$ will make this inequality usable higher up in the branch-and-cut tree where it could be even more effective.

This observation leads us to consider the special case $S = H$, where (17) is valid for $\mathcal{P}$ and induces a facet if $S \in \mathcal{C}^*$. Recall from the discussions following Definition 2 that S is S-hereditary only if it is a clique, in which case inequality (17) is precisely inequality (16) for clique S. If $s = 1$ and we consider clique interdiction, this inequality induces a facet of $\mathcal{P}$ if the clique $S \in \mathcal{C}^*$. We also know from Proposition 2 that the 1-club (clique) $S \in \mathcal{C}^*$ if and only if it is one-step maximal. As clique is a hereditary property, this is equivalently saying that the clique S must be inclusionwise maximal for inequality (17) to induce a facet of $\mathcal{P}$. Therefore, the special case of Theorem 1 with $H = S$ and $s = 1$ extends the result of Furini et al. (2019) to our setting with interdiction penalty.

Now consider the same special case $S = H$ but with $s \geq 2$. For a clique S to be a critical s-club, i.e., a one-step maximal s-club, no vertex in $V \setminus S$ can be adjacent to a vertex in S; otherwise, such a vertex along with vertices in S forms an s-club for any $s \geq 2$. Thus, we can conclude that if S is clique that induces a maximal connected component of the graph G, inequality (17) induces a facet of $\mathcal{P}$ for any $s \geq 2$. We can now see Theorem 1 as a generalization of the result of Furini et al. (2019) to s-club interdiction under interdiction penalty *for any* $s \geq 2$. It should be noted, however, that the criticality requirement on the clique is a very restrictive condition when $s \geq 2$, as the clique must form a connected component by itself. It turns out, as the following theorem established by a direct proof shows, that it is sufficient for the clique S to be maximal with respect to the clique property (and not necessarily critical with respect to the s-club property) for inequality (16) to induce a facet of $\mathcal{P}$ for any $s \geq 2$.

Theorem 2. *Given a graph $G = (V, E)$, a positive integer s, and an inclusionwise maximal clique S in G, the following inequality is valid for $\mathcal{P}$ and induces a facet of $\mathcal{P}$:*

$$\theta \geq |S| - x(S). \tag{18}$$

Because enumerating maximal cliques is not computationally desirable given that there could be exponentially many in a graph (Moon and Moser 1965), we do not explicitly make use of this facet in our computational studies. However, the next two results—based on specially structured s-clubs—are interesting to us from a computational perspective.

Theorem 3. *Given a graph $G=(V,E)$ and an integer $s \geq 2$, suppose that for some vertex $v \in V$ the closed neighborhood of v forms a critical s-club. That is, $N_G[v] \in \mathcal{C}^*$, a critical star centered at v. If $N_G(v)$ is an independent set, the following inequality is valid and induces a facet of $\mathcal{P}$:*

$$\theta \geq \deg_G(v) + 1 - x(N_G(v)) - \deg_G(v) x_v. \tag{19}$$

This inequality can be viewed as a strengthening of the coefficient of x_v in constraint (14b) with $S=N_G[v]$ and $H=N_G(v)$. Theorem 4 that follows is similar to Theorem 3, and is based instead on critical *edge stars*, i.e., sets of the form $N_G(u) \cup N_G(v)$ where $\{u,v\} \in E$. Due to the asymmetry in the coefficients of x_u and x_v, in general Theorem 4 corresponds to two facet-inducing inequalities obtained by interchanging vertices u and v.

Theorem 4. *Given a graph $G=(V,E)$ and an integer $s \geq 3$, consider adjacent vertices u and v such that $N_G(u) \cup N_G(v) \setminus \{u,v\}$ is a non-empty independent set. If $N_G(u) \cup N_G(v) \in \mathcal{C}^*$, then*

$$\theta \geq \deg_G(u) + \deg_G(v) - c_{uv} - x(L_{uv}) - [\deg_G(u) - c_{uv}]x_u - [\deg_G(v) - \min(c_{uv}, 1)]x_v \tag{20}$$

is valid and induces a facet of $\mathcal{P}$, where $L_{uv} := N_G(u) \cup N_G(v) \setminus \{u,v\}$ and $c_{uv} := |N_G(u) \cap N_G(v)|$.

Many real-life social and biological networks demonstrate a power law degree distribution and are also extremely sparse in terms of edge density (Chung and Lu 2006, Newman 2003, Barabási and Albert 1999). So it is not uncommon in practice to find vertices and edges with a large number of independent neighbors in sparse real-life graphs, such as those used in our computational study. Nonetheless, we also do not recommend strictly testing the satisfaction of the sufficient conditions in order to add the critical vertex and edge star facets during computations. These two results essentially serve to motivate our emphasis on vertex and edge stars in building the master relaxation of Formulation (14) used in our delayed constraint generation algorithm discussed in Section 5.

4. Hereditary s-Clubs and Latency-s Connected Dominating Sets

Given a graph $G=(V,E)$, we say that $D \subseteq V$ is a *dominating set* if every vertex outside D has a neighbor in D. We say that D is a *connected* dominating set if in addition, $G[D]$ is connected. In essence, a connected dominating set ensures that every pair of distinct vertices outside the dominating set have a path connecting them (whose internal vertices are contained in the dominating set). The connection to hereditary s-clubs, while not obvious, arises when we require distance bounds in addition to connected domination. Definition 4 below is adapted from its counterpart for directed graphs introduced by Validi and Buchanan (2020).

Definition 4 (Validi and Buchanan (2020)). Given a graph $G=(V,E)$, a subset of vertices D is called a *latency-s connected dominating set* (latency-s CDS) if it is a dominating set in G and for every pair of distinct vertices in V there exists a path of length at most s whose internal vertices belong to D.

If D is a latency-s CDS, then it is a dominating set that is also an s-club. Note that the length-bounded path requirement applies to vertex-pairs inside D as well. Clearly, a dominating s-club is not necessarily a latency-s CDS (see Figure 4a). It is also easy to see that a dominating $(s-2)$-club is a latency-s CDS. However, a latency-s CDS is not necessarily a dominating $(s-2)$-club (see Figure 4b).

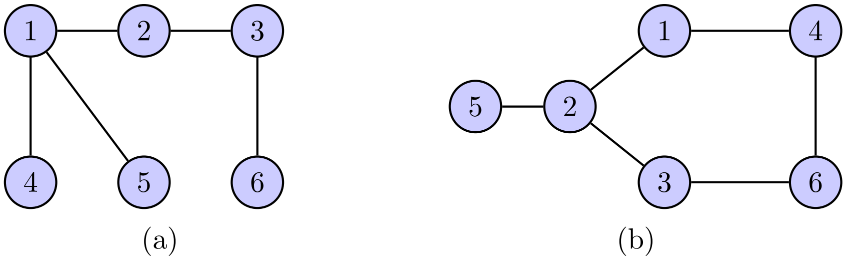

Figure 4 **(a) Set $\{1,2,3\}$ forms a dominating 2-club, but it is not a latency-2 CDS since the length of the path between vertices 5 and 6 is 4. (b) Set $\{1,2,3\}$ forms a latency-3 CDS. (Note that vertices 4 and 6 are adjacent and vacuously satisfy the requirement.) Clearly, it is not a 1-club (clique).**

Given an s-club S in graph $G=(V,E)$, we say that D is a "latency-s CDS over S" if and only if D is a latency-s CDS in the induced subgraph $G[S]$. In general, a graph G has a latency-s CDS if and only if $\operatorname{diam}(G) \leq s$. The necessity can be deduced from the fact that every pair of vertices must be connected by a path of length at most s, in order for a latency-s CDS to exist. Conversely, if $\operatorname{diam}(G) \leq s$ we know that V is a latency-s CDS. A meaningful optimization problem therefore is to find a latency-s CDS of minimum cardinality. The notion of a latency-s CDS is relevant to s-club interdiction because of its close relationship to hereditary s-clubs as crystallized in the following result.

PROPOSITION 4. *Consider a graph $G=(V,E)$ in which S is an s-club and $H \in \mathcal{H}(S)$ such that $H \neq S$. Then $S \setminus H$ is a latency-s CDS over S. Conversely, suppose that a non-empty $D \subseteq S$ is a latency-s CDS over S. Then $S \setminus D \in \mathcal{H}(S)$.*

Proposition 4 allows us to find large subsets $H \in \mathcal{H}(S)$ by equivalently finding small latency-s CDSs. Hence, when identifying violated constraints in our delayed constraint generation approach, we can replace the problem of finding a large $H \in \mathcal{H}(S)$ by finding a minimum cardinality latency-s CDS sets in S. By framing the problem in this manner we can exploit existing methods to solve the minimum latency-s CDS problem (Validi and Buchanan 2020).

5. Implementing a Decomposition Branch-and-Cut Algorithm

Based on the results of Sections 3 and 4, our approach to solve Formulation (14) employs delayed constraint generation in a decomposition and branch-and-cut (DBC) framework. This DBC algorithm starts by solving a master relaxation of Formulation (14) where $\mathcal{C}^*$ in constraint (14b) is replaced by an initial collection of s-clubs $\mathcal{S}^0 \subseteq \mathcal{S}$. As this master relaxation is solved using an LP relaxation based branch-and-cut (BC) algorithm, nodes are pruned as usual by infeasibility or by bound. However, if the LP relaxation optimum (θ^i, x^i) at some BC node i is integral, we must verify its feasibility.

Algorithm 1: Separation procedure

```
Input: Integral LP optimum $(\hat{\theta}, \hat{x})$ at the DBC node.
Output: $(S, H)$, where $S$ is an $H$-hereditary $s$-club corresponding to a violated constraint,
        if one exists.
$S \leftarrow$ a maximum $s$-club in $G \setminus T^{\hat{x}}$
if $|S| > \hat{\theta}$ then
    $D \leftarrow$ a minimum latency-$s$ CDS in $G[S]$
    return $(S, S \setminus D)$
else
    $(\hat{\theta}, \hat{x})$ is feasible
```

To this end, we can solve a separation subproblem in order to verify if a constraint of type (14b) corresponding to some H-hereditary s-club S is violated. First, we find a maximum s-club in the interdicted graph, say S. If $\bar{\omega}_s(G \setminus T^{x^i}) = |S| > \theta^i$, we must add a violated constraint to eliminate this solution. In order to find an $H \in \mathcal{H}(S)$, based on Proposition 4, we can solve the minimum latency-s CDS problem on the subgraph $G[S]$. If $\mathcal{H}(S)$ is empty, then the minimum latency-s CDS will be S itself, and we add constraint (3b) for S. After the violated constraint is added, the LP relaxation at node i is re-solved. If $\bar{\omega}_s(G \setminus T^{x^i}) = |S| \leq \theta^i$, no violated constraint exists, we can certify that the integral solution (θ^i, x^i) is feasible to the original problem and prune that BC node. This separation routine is described in Algorithm 1.

The separation subproblem ensures the correctness of the overall algorithm despite starting with a relaxation of the original problem. From our experiments, we found that the DBC algorithm typically generates far fewer constraints than all possible constraints of type (14b). In the following we discuss how we initialize the master relaxation in our computational study described in Section 6, as well as specify some additional implementation details of the heuristic separation procedure used in our experiments when $s = 2$ and $s = 3$.

5.1. Implementation Details for 2-club Interdiction

When solving the 2-club interdiction problem, we initialize the master relaxation with constraints based on stars in G (see Remark 2). We write the constraints for the star $N_G[v]$ centered at v, with the hereditary subset $H = N_G(v)$. This choice of H is maximal as long as $N_G[v]$ is not a clique. Hence, the master relaxation constraints have the following form:

$$\theta \geq \deg_G(v) + 1 - x(N_G(v)) - (\deg_G(v) + 1)x_v. \tag{21}$$

In our experiments, we add constraint (21) only for those vertices that correspond to the top 20% of the largest degrees in G.

Once a maximum 2-club S that corresponds to a violated constraint is found, we use a simpler heuristic approach to identify a hereditary subset for the case of $s = 2$, instead of finding a minimum latency-s CDS inside $G[S]$ (line 3 of Algorithm 1). This simplification is based on the observation that if $G[S]$ contains a dominating vertex v, then the set $\{v\}$ is a latency-2 CDS and S is a $S \setminus \{v\}$-hereditary 2-club. In fact, if S is not a clique, then $\{v\}$ is a *minimum* latency-2 CDS of $G[S]$.

Algorithm 2 outlines the pseudocode of a heuristic separation procedure for $s = 2$ that does not rely on solving the minimum latency-s CDS problem. If we find any vertex v that dominates $G[S]$, we return immediately having identified a strong violated constraint. Otherwise, we find all the leaves L in $G[S]$ and $S \setminus L$ is a feasible latency-2 CDS. If no leaves exists, then L is empty, and we effectively add a constraint of type (3b). In all three cases, note that the constraint identified will be violated by $(\hat{\theta}, \hat{x})$. This heuristic separation procedure was found to be effective for the case $s = 2$ in our computational studies.

5.2. Implementation Details for 3-club Interdiction

In this case, the initial set $\mathcal{S}^0$ of 3-clubs includes the constraints associated with edge stars corresponding to $S := N_G(u) \cup N_G(v)$ for each $\{u, v\} \in E$, with $H = N_G(u) \cup N_G(v) \setminus \{u, v\}$. Clearly, $H \in \mathcal{H}(S)$, and H would belong to $\mathcal{H}^*(S)$ unless $H \cup \{u\} \in \mathcal{H}^*(S)$, $H \cup \{v\} \in \mathcal{H}^*(S)$, or $H \cup \{u, v\} \in \mathcal{H}^*(S)$. In other words, H is at most two elements short of a maximal member of $\mathcal{H}(S)$ in case it is not in $\mathcal{H}^*(S)$. The constraint of type (14b) specializes to the following for edge stars:

$$\theta \geq |S| - x(S \setminus \{u, v\}) - |S|(x_u + x_v) \quad \forall \{u, v\} \in E. \tag{22}$$

As every 2-club is also a 3-club, we also add constraint (21) for all the vertices in G. In general, the star constraints are not dominated by edge star constraints (22).

Algorithm 2: Separation algorithm for $s = 2$

```
Input: Integral LP optimum $(\hat{\theta}, \hat{x})$ at the DBC node.
Output: $(S, H)$, where $S$ is a $H$-hereditary 2-club corresponding to a violated constraint, if
        one exists.
$S \leftarrow$ a maximum 2-club in $G \setminus T^{\hat{x}}$
if $|S| > \hat{\theta}$ then
    $L \leftarrow \emptyset$
    for $v \in S$ do
        if $|N_{G[S]}(v)| = |S| - 1$ then
            return $(S, S \setminus \{v\})$
        if $|N_{G[S]}(v)| = 1$ then
            $L \leftarrow L \cup \{v\}$
    return $(S, L)$
else
    $(\hat{\theta}, \hat{x})$ is feasible
```

6. Computational Experiments

In this section, we report on the results of our numerical experiments designed to assess the capabilities of the proposed DBC algorithm to solve the s-club interdiction problem on real and synthetic benchmark instances. All experiments are conducted on a 64-bit Windows® 10 Pro machine with 16GB of RAM and 1.8 GHz processor with 7 cores. All algorithms are implemented in C++, compiled using Microsoft® Visual Studio® 2017, and Gurobi™ Optimizer v9.0.2 is used to solve the MILPs (Gurobi Optimization, LLC 2021). Our codes are publicly available (Daemi et al. 2021a,b).

Our testbed consists of two groups of instances. Group-1 contains 22 graphs from the Tenth DIMACS Implementation Challenge (DIMACS-10), see (Dimacs 2012). Group-2 contains 18 graphs taken from the following online repositories: Stanford Large Network Dataset Collection (SNAP) (Leskovec and Krevl 2014), the BGU Social Networks Security Research Group (BGU) (Lesser et al. 2013), the Koblenz Network Collection (KONECT) (Kunegis 2013) and the Network Repository (NR) (Rossi and Ahmed 2015). Most of the instances in our testbed come from real-world networks. Further, the instances `Gplus, Facebook1, Facebook2,` and `Douban` in Group-2 represent snapshots of real online social networks. The instances in Group-2 were also used in the computational studies in Raghavan and Zhang (2019).

Tables 1 and 2 list all the graphs in our testbed. We converted the directed graphs in our testbed to undirected graphs by ignoring the orientation on the arcs. For each instance we list the number of vertices, edges, and the edge density $\rho(G) = |E|/\binom{|V|}{2}$. To solve the maximum s-club problem during separation, we use the "ICUT" algorithm introduced by Salemi and Buchanan (2020), the code for which has been made publicly available by the authors. ICUT is an effective integer programming based exact solver for the maximum s-club problem for general values of s on the instances we use in our testbed. It sequentially solves the maximum s-club problem on several smaller subgraphs using a delayed constraint generation framework. Tables 1 and 2 report the time it takes to find $\bar{\omega}_2(G)$ and $\bar{\omega}_3(G)$ using the ICUT solver. TL in the Time column indicates that the solver terminated by reaching the time limit.

Using the Gurobi parameter `GRB_DoubleParam_Timelimit`, we impose a time limit of 3600 seconds on the solve time of the master problem, and the same time limit on each call to solve the maximum s-club subproblems in ICUT and the minimum latency-s CDS problem during the separation procedure. Reaching the time limit while solving any of these problems will terminate the overall algorithm (usually quickly), in which case we have failed to solve the problem to optimality on that instance. We also use the Gurobi parameter `LazyConstraint` to add the the violated constraints found in the separation procedure on-the-fly.

As discussed in Section 5, the DBC algorithm requires solving the maximum s-club problem several times, once for every integral solution $(\hat{\theta}, \hat{x})$ that is found in the BC tree to verify its feasibility. Therefore, if solving the maximum s-club problem requires a significant amount of time for a given graph, then we do not expect the interdiction problem to be solved in a reasonable amount of time. More critically, reaching the time limit on the maximum s-club solver without producing a violated s-club affects the overall correctness. For this reason, we only consider those instances in the larger test bed described next on which we can find a large enough s-club in reasonable time using our chosen solver. As it can be seen in Tables 1 and 2, all instances in Group-1 are solved within a reasonable time (less than 5 minutes) for both $s = 2$ and $s = 3$. For Group-2, all the instances except `Douban` are solved to optimality when $s = 2$. However, when $s = 3$, only 9 out of 18 instances are solved to optimality within the time limit, and among these instances, `Advogato` and `Facebook1` requires a significant amount of time. For this reason, when $s = 2$, we do not include instance `Douban` in our experiments and when $s = 3$, for instances in Group-2, we use heuristic approaches to find the maximum 3-club and the minimum latency-3 CDS instead of using the exact methods we implement in other cases.

In Section 6.1, we use the Group-1 instances to show how the naive Formulation (3) and Formulation (14) based on hereditary s-clubs compare when each is used in the DBC algorithm. In Sections 6.2 and 6.3, we present the results of our experiments with both groups of instances using the best performing DBC algorithm and heuristic approaches.

6.1. The Impact of Using the H-hereditary s-club Formulation

In Section 3.1, we introduced the idea of partitioning an s-club S into a hereditary subset H and $S \setminus H$ in order to generate a constraint of type (14b). Here, we assess the impact of using these

Table 1 DIMACS-10 instances in Group-1 and the time taken to solve the maximum s-club problem for $s = 2, 3$ using the ICUT algorithm. Instances `celegansneural`, `celegans-metabolic`, and `PGPgiantcompo` are shortened to `celegansn`, `celegansm`, and `PGP`, respectively, in the other tables.

Graph G	$\|V\|$	$\|E\|$	$\rho(G)$ (%)	$\bar{\omega}_2(G)$	Time (s)	$\bar{\omega}_3(G)$	Time (s)
karate	34	78	13.90	18	0.01	25	0.00
dolphins	62	159	8.41	13	0.14	29	0.02
lesmis	77	254	8.68	37	0.00	58	0.00
polbooks	105	441	8.08	28	0.09	53	0.00
adjnoun	112	425	6.84	50	0.00	82	0.19
football	115	613	9.35	16	0.84	58	1.52
jazz	198	2,742	14.06	103	0.42	174	0.05
celegansneural	297	2,148	4.89	135	0.02	243	0.37
celegans-metabolic	453	2,025	1.98	238	0.02	371	0.10
email	1,133	5,451	0.85	72	6.89	212	65.69
polblogs	1,490	16,715	1.51	352	30.82	776	31.43
netscience	1,589	2,742	0.22	35	0.02	54	0.02
add20	2,395	7,462	0.26	124	0.17	671	0.23
data	2,851	15,093	0.37	18	13.27	32	15.51
uk	4,824	6,837	0.06	5	12.32	8	13.86
power	4,941	6,593	0.05	20	0.68	30	0.69
add32	4,960	9,462	0.08	32	0.48	99	0.50
hep-th	8,361	15,751	0.05	51	1.34	120	41.66
whitaker3	9,800	28,989	0.06	9	66.50	15	90.78
crack	10,240	30,380	0.06	10	81.95	17	96.06
PGPgiantcompo	10,680	24,316	0.04	206	4.07	422	4.30
cs4	22,499	43,858	0.02	6	165.26	12	236.51

Table 2 Instances in Group-2 and the time taken to solve the maximum s-club problem for $s = 2, 3$ using the ICUT algorithm.

Graph G	Source	$\|V\|$	$\|E\|$	$\rho(G)$(%)	$\bar{\omega}_2(G)$	Time (s)	$\bar{\omega}_3(G)$	Time (s)
G04	SNAP	10,876	39,994	0.07	104	4.89	$\geq$ 181	TL
G05	SNAP	8,846	31,839	0.08	89	9.96	$\geq$ 258	TL
G06	SNAP	8,717	31,525	0.08	116	3.64	$\geq$ 243	TL
G08	SNAP	6,301	20,777	0.10	98	23.08	453	464.72
G09	SNAP	8,114	26,013	0.08	103	20.93	449	945.33
B-Alpha	SNAP	3,783	14,124	0.20	512	0.66	1,294	626.06
B-OTC	SNAP	5,881	21,492	0.12	796	1.36	$\geq$ 1,969	TL
AS01	SNAP	10,670	22,002	0.04	2,313	15.25	4,997	613.26
AS02	SNAP	10,900	31,180	0.05	2,344	15.89	5,352	202.68
Ning	BGU	10,298	40,887	0.09	688	4.29	$\geq$ 2,294	TL
Hamsterster	Konect	1,858	12,534	0.78	273	0.18	680	89.18
Escorts	Konect	10,106	39,016	0.08	312	4.32	$\geq$ 679	TL
Anybeat	N.R.	12,645	49,132	0.06	4,801	9.17	$\geq$ 7,752	TL
Advogato	N.R.	6,551	39,432	0.31	808	1.64	2,193	1,937.74
Gplus	Konect	23,613	39,194	0.01	2,762	9.99	$\geq$ 4,767	TL
Facebook1	BGU	39,446	50,228	0.01	1,366	27.45	11,542	2,136.21
Facebook2	Konect	2,888	2,981	0.07	770	0.13	1,241	0.18
Douban	N.R.	154,908	327,162	0.00	$\geq$ 288	TL	$\geq$ 911	TL

constraints by comparing three different methods. In the first method a constraint of type (3b) is used in the initialization and during separation (Method 1). The second method uses the H-hereditary s-club constraint (14b) in the initialization of the master relaxation and constraint (3b) during separation (Method 2). The third method uses constraint (14b) during initialization and separation (Method 3). In all three methods we initialize $\mathcal{S}^0$ by creating a set of s-clubs in the form of stars (when $s=2,3$) or edge stars (when $s=3$), and add a constraint for each s-club in $\mathcal{S}^0$ to initialize the master relaxation. Note that the type of the constraint we add for each s-club in $\mathcal{S}^0$ depends on the method, as explained before. We compare the performance of these three methods in terms of running time and visualize the comparison using performance profiles (Dolan and Moré 2002).

In order to construct a performance profile, we define $\mathcal{I}$ as the set of the instances in our testbed, $\mathcal{M}$ as the set of methods, and $t_{i,m}$ as the running time of solving the instance i by method m. The baseline of the comparison is the shortest running time among three methods for every instance, and we compute the performance ratio as $r_{i,m} = t_{i,m}/t_i^*$, where $t_i^* = \min\{t_{i,m} : m \in \mathcal{M}\}$. Then, for every method m, we define $f_m(\tau)$ as the empirical cumulative distribution function of the performance ratio $r_{i,m}$. As stated in Equation (23), $f_m(\tau)$ is the fraction of the instances in our testbed that were solved by method m within a factor τ of the fastest solve-time for that instance.

$$f_m(\tau) = \frac{|\{i \in \mathcal{I} : t_{i,m} \le \tau t_i^*\}|}{|\mathcal{I}|}. \tag{23}$$

If we observe that $f_m(\tau) \ge f_{m'}(\tau)$ for all $\tau \ge 1$, then there is evidence to suggest that method m is better than m' on this testbed. In particular, $f_m(1.0)$ is the fraction of the instances in the testbed for which method m is the fastest.

Figure 5 shows the performance profiles of Method 1, Method 2, and Method 3 for all the instances in Group-1 for $s=2$. We selected $\alpha = 0.5$ and $\alpha = 2$ for these experiments, meaning that in the former setting it is cheap to interdict vertices (i.e., for every two interdicted vertices the maximum s-club in the interdicted graph should reduce by at least one) while in the latter setting it is expensive to interdict vertices (i.e., for every interdicted vertex the maximum s-club in the interdicted graph should reduce by at least two).

It can be seen that for both values of α, Method 3 is significantly better than Methods 1 and 2 on this testbed for $s=2$. The performance of Method 2 is generally within 10 times the fastest running time, while Method 1 has a far worse performance overall, achieving 10 times the fastest running time only for for less that 50% of the instances.

The performance profile for $s=3$ is shown in Figure 6. As before, Method 3 has the best performance on this testbed. Method 2 performs worse than it did when $s=2$, because there are about 5% of the instances whose solution times are not within 100 times the fastest running time when $\alpha = 2$. Method 1, on the other hand, has a similarly poor performance now, as it was the case with $s=2$.

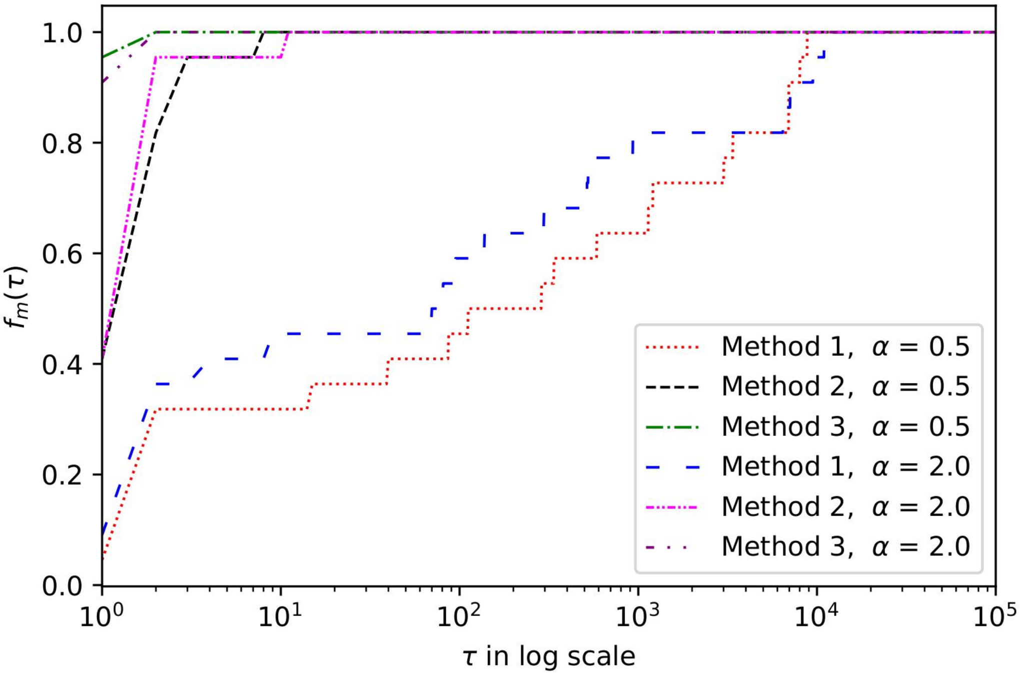

Figure 5 **Performance profile based on the running time of methods for $s = 2$ and $\alpha \in \{0.5, 2\}$.**

These comparisons show that, in general, Method 3 outperforms the other two methods. This observation confirms that using constraints based on H-hereditary s-clubs at initialization and during separation can significantly improve the performance of our DBC algorithm. Therefore, we use this method in the remaining computational experiments in Sections 6.2 and 6.3.

Before discussing the results of our main experiments with Method 3, we should mention that we evaluated its performance by conducting two other experiments reported in greater detail in Appendix B. First, a root node performance comparison between Method 1 and Method 3. The results show that Method 3 outperforms Method 1 by providing the same or smaller gaps and objective values for nearly all the instances (see Appendix B.1). We also evaluated the dependency of Method 3 on primal heuristics built into the Gurobi solver, comparing its performance with and without these heuristics. Neither choice consistently offers superior performance, and we discuss this in greater detail in Appendix B.2.

6.2. Results for Group-1 Instances

We report on the results obtained for the instances in Group-1 for $s \in \{2, 3\}$ and $\alpha \in \{0.5, 1, 2\}$ using Method 3 in this section. For each instance we report the number of interdicted vertices under $x(V)$, the s-club number of the interdicted graph under θ, the total number of BC nodes explored, the total number of separation callbacks under #CB, the total number of violated constraints

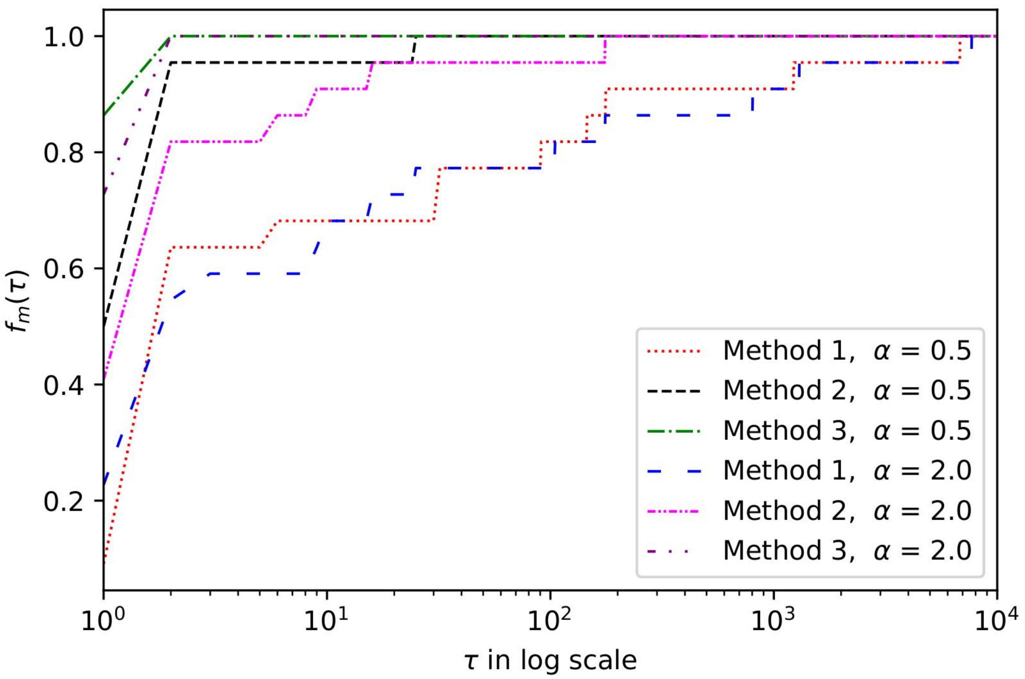

Figure 6 **Performance profile based on the running time of methods for $s = 3$ and $\alpha \in \{0.5, 2\}$**

added under #Cuts (broken down by each type when $s = 2$ under Star, Leaf, Regular), the total running time, the total time taken to solve the maximum s-club problem, the total time taken to solve the minimum latency-s CDS problem (when $s = 3$), and the relative optimality gap at termination.

Tables 3, 4, and 5 show the results for $\alpha = 2, 1,$ and 0.5, respectively, with $s = 2$. All the instances are solved to optimality under a one hour time limit with the exception of `jazz` and `polblogs` that are not solved to optimality for any value of α. We can observe in Table 1 that for most of the instances the 2-club number of the original graph $\bar{\omega}_2(G)$ tends to be much larger than the 2-club number after interdiction (i.e., θ) for all values of α we consider. For example, the values of $\bar{\omega}_2(G)$ in the original graph for `celegans-metabolic` and `PGPgiantcompo` are respectively 238 and 206, while they decrease to 32 and 76 after interdiction when $\alpha = 2$. These values further decrease to 10 and 47 as $\alpha = 0.5$ because interdiction is cheaper in this case. However, when $\bar{\omega}_2(G)$ is very small compared to $|V|$, we find θ to be almost equal to $\bar{\omega}_2(G)$. Consider the instance `cs4` as an example, with $\bar{\omega}_2(G) = 6$. The 2-club number of this graph remains the same after interdiction for all values of α we considered (note that this instance has 22,449 vertices).

Another observation is that for most of the instances, decreasing the value of α from 2 to 0.5 makes the instance more difficult to solve and as a result, the number of BC nodes explored and running times increase. For example, when $\alpha = 2$, `football` is solved in the root node in 3.07 seconds, while for $\alpha = 0.5$, the number of explored nodes is $973,384$ and the running time increases to 92.14 seconds. This behavior could be due to the fact that as α decreases, interdiction is cheaper and there are many more feasible solutions of high quality distributed across the BC tree, thereby resulting in far fewer BC nodes being pruned.

Table 3 Results for Group-1 instances with $s = 2$ and $\alpha = 2$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Leaf	Regular	Total time (s)	s-club time (s)	Gap (%)
karate	3	9	1	5	0	0	3	0.08	0.01	0.00
dolphins	0	13	1	2	1	0	0	0.10	0.06	0.00
lesmis	2	18	1	4	1	0	0	0.38	0.01	0.00
polbooks	1	25	1	14	4	0	8	0.70	0.45	0.00
adjnoun	6	14	1	5	1	0	0	0.56	0.46	0.00
football	0	16	1	12	0	0	10	3.07	3.01	0.00
jazz	5	71	37,284	4,735	1	0	4,731	TL	3570.47	16.17
celegansn	12	36	63	8	1	0	3	7.18	6.87	0.00
celegansm	13	32	21	4	1	0	0	0.33	0.03	0.00
email	1	52	1	3	1	0	0	12.37	12.11	0.00
polblogs	21	154	1,180	100	2	0	92	TL	3607.45	18.67
netscience	3	21	1	3	1	0	0	0.16	0.04	0.00
add20	14	68	32	5	1	0	0	6.88	0.81	0.00
data	0	18	0	2	0	0	1	7.64	7.58	0.00
uk	0	5	1	8	0	0	7	29.93	29.76	0.00
power	2	15	1	3	0	0	1	1.74	1.60	0.00
add32	0	32	1	2	1	0	0	0.76	0.63	0.00
hep-th	3	40	1	3	1	0	0	3.87	3.05	0.00
whitaker3	0	9	0	2	1	0	0	48.23	47.96	0.00
crack	0	10	1	2	1	0	0	38.09	37.86	0.00
PGP	11	76	79	6	4	0	0	26.46	20.02	0.00
cs4	0	6	1	6	0	0	5	407.60	405.63	0.00

Tables 6, 7, and 8 report our results for 3-club interdiction with $\alpha = 2, 1$, and 0.5, respectively. The number of instances that are solved to optimality within the time limit are 14, 13, and 12 for $\alpha = 2, 1$, and 0.5, respectively. (By contrast, 20 out of the 22 graphs for all three values of α were solved to optimality for 2-club interdiction.) In general, we observe that the 3-club interdiction problem is significantly more difficult to solve than its 2-club counterpart. When solving the 3-club interdiction problem using Method 3, we invoke separation more frequently and each callback to the separation problem takes more time to finish.

During separation, the maximum 3-club problem takes more time to solve than the maximum 2-club problem on our testbed (see Table 1 for instances where the difference is significant). But more importantly, on each maximum 3-club we find, the algorithm now solves the latency-3 CDS problem as opposed to the heuristic used for $s = 2$. We find that the instances that were not solved to optimality also typically have significantly larger running times for finding a latency-3 CDS, compared to those instances that we do solve to optimality.

Table 4 Results for Group-1 instances with $s = 2$ and $\alpha = 1$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Leaf	Regular	Total time (s)	s-club time (s)	Gap (%)
karate	5	5	1	9	3	0	4	0.12	0.01	0.00
dolphins	0	13	34	4	2	0	1	0.27	0.12	0.00
lesmis	8	10	57	10	4	0	3	0.16	0.01	0.00
polbooks	13	12	144	12	6	0	3	0.42	0.14	0.00
adjnoun	6	14	23	6	1	0	0	1.01	0.80	0.00
football	0	16	1	14	0	0	11	3.52	3.42	0.00
jazz	21	45	74,282	6,389	22	0	6,361	TL	3598.92	10.42
celegansn	21	23	175	6	1	0	0	4.24	3.92	0.00
celegansm	21	18	54	5	1	0	0	0.35	0.05	0.00
email	8	42	49	3	1	0	0	12.72	11.38	0.00
polblogs	113	55	972	140	1	0	135	TL	3737.62	30.78
netscience	3	21	1	3	1	0	0	0.17	0.04	0.00
add20	30	49	281	5	2	0	0	8.98	0.64	0.00
data	0	18	1	2	0	0	1	7.83	7.69	0.00
uk	0	5	1	8	0	0	7	30.01	29.85	0.00
power	2	15	1	3	0	0	1	1.83	1.65	0.00
add32	0	32	1	2	1	0	0	1.12	0.64	0.00
hep-th	5	36	1	3	1	0	0	4.79	3.55	0.00
whitaker3	0	9	0	2	1	0	0	47.18	46.99	0.00
crack	0	10	1	2	1	0	0	34.91	34.65	0.00
PGP	24	62	325	5	1	0	2	44.54	25.98	0.00
cs4	0	6	1	6	0	0	5	411.37	409.26	0.00

Table 5 Results for Group-1 instances with $s = 2$ and $\alpha = 0.5$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Leaf	Regular	Total time (s)	s-club time (s)	Gap (%)
karate	8	3	41	17	11	0	3	0.16	0.01	0.00
dolphins	3	10	200	16	13	0	2	0.52	0.25	0.00
lesmis	10	8	90	18	16	0	0	0.52	0.04	0.00
polbooks	16	10	268	23	17	0	2	0.41	0.13	0.00
adjnoun	12	10	331	16	12	0	1	1.07	0.67	0.00
football	1	15	973,384	96	2	0	90	92.14	26.56	0.00
jazz	44	26	1,497,964	5,307	56	0	5,249	TL	1271.72	12.62
celegansn	23	21	2,214	13	7	0	2	6.18	5.61	0.00
celegansm	32	10	181	7	2	0	1	0.48	0.09	0.00
email	12	38	1,020	6	1	0	0	39.51	37.44	0.00
polblogs	125	42	452,127	138	1	0	116	TL	1976.77	2.19
netscience	3	21	206	3	1	0	0	1.20	0.04	0.00
add20	52	34	7,088	9	2	0	0	12.59	0.95	0.00
data	1	17	20	3	0	0	1	13.25	12.86	0.00
uk	0	5	1	8	0	0	7	29.07	28.87	0.00
power	2	15	1	3	0	0	1	2.01	1.59	0.00
add32	4	29	58	4	1	0	0	3.19	1.09	0.00
hep-th	18	29	412	4	2	0	0	10.95	6.34	0.00
whitaker3	0	9	1	2	1	0	0	47.00	46.72	0.00
crack	1	9	1	3	1	0	0	72.81	72.51	0.00
PGP	45	47	5,858	3	1	0	0	41.63	8.20	0.00
cs4	0	6	1	7	0	0	6	503.88	501.18	0.00

The number of calls to the separation routine, the number of cuts added, and the number of BC nodes have increased on average when compared to what is observed for $s = 2$. One possible explanation for this behavior is that the initial strength of the master relaxation based on s-clubs in $\mathcal{S}^0$ is not as strong when $s = 3$ compared to when $s = 2$. In other words, the edge star based constraints (22) when $s = 3$ are possibly not as strong as star based constraints (21) when $s = 2$. The relative weakness of the master relaxation based on edge star constraints may be due to large 3-clubs in the graph that do not resemble edge stars, while it is more common for large 2-clubs to resemble stars.

As solving the separation problem for both values of s requires a significant proportion of the overall solution time, we have evaluated the effect of using heuristics to solve the separation problem. Our results show that this approach might improve the performance of Method 3 depending on the test bed; see Appendix B.3 for more details.

We close this section by noting that similar to the $s = 2$ case, the optimal value of θ shows that our model decreases the size of the maximum 3-club significantly except for those cases where $\bar{\omega}_3(G)$ is small. As before, the interdiction problem becomes more difficult to solve when the value of α is decreased.

Table 6 Results for Group-1 instances with $s = 3$ and $\alpha = 2$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)	Gap (%)
karate	5	6	1	8	4	0.32	0.00	0.02	0.00
dolphins	3	19	11,011	95	93	4.49	1.86	0.55	0.00
lesmis	8	11	72	28	20	0.79	0.02	0.18	0.00
polbooks	10	25	50,926	192	187	20.54	7.18	1.94	0.00
adjnoun	10	25	497,453	774	767	429.93	208.39	13.00	0.00
football	2	50	11,670	120	116	232.65	220.00	5.16	0.00
jazz	5	145	45,760	3,633	3,629	TL	549.44	2646.06	40.17
celegansn	28	68	24,388	1,822	1,814	TL	2375.78	1164.87	37.85
celegansm	22	29	5,982	30	28	52.81	2.10	8.98	0.00
email	140	94	1	26	24	TL	3752.68	79.68	81.84
polblogs	340	228	1	20	19	TL	733.57	3180.50	81.93
netscience	6	27	155	8	6	2.78	0.12	0.04	0.00
add20	61	125	3,816	338	330	TL	77.85	787.69	53.32
data	0	32	1	8	6	45.97	43.34	0.04	0.00
uk	0	8	1	5	3	21.87	21.56	0.01	0.00
power	1	27	1	7	5	5.57	4.50	0.02	0.00
add32	5	75	729,074	65	63	TL	15.86	1.03	1.00
hep-th	0	120	1	83	82	TL	3573.15	4.91	44.66
whitaker3	0	15	1	6	4	177.24	175.41	0.01	0.00
crack	0	17	1	6	5	204.25	201.43	0.02	0.00
PGP	4	266	1	104	103	TL	2955.97	40.22	56.79
cs4	0	12	1	6	4	653.00	648.15	0.02	0.00

6.3. Results for Group-2 Instances

We evaluate the performance of Method 3 on Group-2 instances in this section. Tables 9, 10, and 11 show the results for $s = 2$. As mentioned before, graph `Douban` is not included in these experiments because the maximum 2-club for this instance is not found within the time limit. The results on the remaining 17 instances show that all of them are solved to optimality within the one hour time

Table 7 **Results for Group-1 instances with $s = 3$ and $\alpha = 1$ using Method 3.**

Graph G	$x(V)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)	Gap (%)
karate	6	5	17	7	3	0.19	0.00	0.02	0.00
dolphins	9	11	11,187	188	184	8.01	3.93	0.82	0.00
lesmis	8	11	84	9	6	0.41	0.00	0.07	0.00
polbooks	20	12	16,007	74	69	12.21	1.80	0.58	0.00
adjnoun	17	15	237,311	269	263	168.81	59.20	2.26	0.00
football	5	45	82,949	2,496	2,491	TL	3241.66	72.25	46.32
jazz	72	28	63,683	6,165	6,159	TL	1003.51	1751.19	33.32
celegansn	40	43	21,710	2,052	2,044	TL	3009.87	513.28	32.90
celegansm	29	19	21,846	34	30	78.16	1.31	8.82	0.00
email	167	70	1	46	45	TL	3639.63	93.73	74.15
polblogs	350	188	1	24	22	TL	850.14	3217.57	76.88
netscience	6	27	1,887	8	6	6.67	0.12	0.04	0.00
add20	102	47	194,989	4,242	4,235	TL	612.74	1549.38	35.94
data	1	31	32,189	28	25	390.48	182.72	0.19	0.00
uk	0	8	1	7	6	33.21	32.74	0.02	0.00
power	3	25	2,774	18	16	26.67	13.04	0.07	0.00
add32	16	55	935,727	92	89	TL	19.22	1.13	7.99
hep-th	6	114	1	93	91	TL	3457.43	4.33	53.44
whitaker3	0	15	1	7	6	212.12	209.15	0.02	0.00
crack	0	17	1	7	6	269.91	265.80	0.02	0.00
PGP	7	252	1	86	84	TL	2425.85	29.35	63.97
cs4	1	10	1	7	6	749.78	742.57	0.02	0.00

Table 8 **Results for Group-1 instances with $s = 3$ and $\alpha = 0.5$ using Method 3.**

Graph G	$x(V)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)	Gap (%)
karate	7	4	78	2	0	0.24	0.00	0.00	0.00
dolphins	18	5	10,654	52	39	3.79	0.45	0.16	0.00
lesmis	13	7	85	4	0	0.59	0.00	0.00	0.00
polbooks	27	8	42,088	44	34	24.74	0.83	0.20	0.00
adjnoun	22	10	148,295	86	75	117.51	9.64	0.45	0.00
football	40	19	249,850	6,843	6,839	TL	2548.35	81.29	36.66
jazz	90	14	546,993	1,606	1,596	TL	230.80	16.60	19.85
celegansn	58	27	174,674	2,364	2,356	TL	2374.93	70.85	27.40
celegansm	35	14	279,199	60	55	717.01	4.37	0.81	0.00
email	251	50	1	56	54	TL	3671.76	143.89	71.90
polblogs	454	62	1	45	43	TL	1357.63	2118.54	68.46
netscience	12	21	7,867	8	6	30.69	0.12	0.04	0.00
add20	116	33	625,508	615	610	TL	45.23	108.92	24.90
data	1	31	471,772	37	33	TL	258.05	0.24	3.55
uk	0	8	1	9	8	51.10	50.15	0.03	0.00
power	7	22	20,025	23	20	114.11	16.14	0.08	0.00
add32	38	39	442,630	112	108	TL	18.63	0.98	14.14
hep-th	11	110	1	99	98	TL	3107.94	1.95	59.10
whitaker3	0	15	1	9	8	294.46	284.14	0.02	0.00
crack	0	17	1	10	9	405.37	394.57	0.03	0.00
PGP	10,680	0	1	89	88	TL	1783.37	41.73	98.68
cs4	1	10	1	9	8	1005.55	994.03	0.03	0.00

limit except instance `Anybeat` with $\alpha = 0.5$, which has a 1% relative optimality gap at termination. We also find that the value of $\bar{\omega}_2(G)$ remains the same for three instances `G05,G08,G09` with $\alpha = 2$, but in all other cases the 2-club number significantly decreases after interdiction. For example, the cardinality of the maximum 2-club of AS02 is $2,344$, while after interdiction it decreases to 114, 80, and 58, respectively, for α equal to $2, 1$, and 0.5.

Table 9 Results for Group-2 instances with $s = 2$ and $\alpha = 2$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Leaf	Regular	Total time (s)	s-club time (s)	Gap (%)
G04	2	67	1	3	1	0	0	49.19	47.22	0.00
G05	0	89	1	2	1	0	0	15.79	13.77	0.00
G06	1	74	1	3	1	0	0	16.75	15.68	0.00
G08	0	98	25	2	1	0	0	21.81	19.38	0.00
G09	0	103	26	2	1	0	0	23.65	19.99	0.00
B-Alpha	23	99	134	3	1	0	0	21.87	17.16	0.00
B-OTC	35	103	89	3	1	0	0	33.70	27.3	0.00
AS01	42	73	159	3	1	0	0	68.76	45.63	0.00
AS02	40	114	209	3	1	0	0	69.64	47.07	0.00
Ning	30	130	551	3	1	0	0	123.45	113.60	0.00
Hamsterster	19	89	65	3	0	0	1	23.07	20.65	0.00
Escorts	21	120	167	3	1	0	0	104.85	100.16	0.00
Anybeat	54	136	2,602	4	2	0	0	399.65	283.61	0.00
Advogato	40	131	300	3	1	0	0	1219.16	1204.89	0.00
Gplus	100	40	279	4	1	0	0	24.31	16.39	0.00
Facebook1	116	1	1	3	1	0	0	55.15	51.04	0.00
Facebook2	10	1	1	4	2	0	0	0.30	0.23	0.00

Table 10 Results for Group-2 instances with $s = 2$ and $\alpha = 1$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Leaf	Regular	Total time (s)	s-club time (s)	Gap (%)
G04	2	67	134	3	1	0	0	50.07	46.66	0.00
G05	8	79	250	4	2	0	0	40.00	35.24	0.00
G06	5	69	60	3	1	0	0	25.07	19.60	0.00
G08	2	94	640	6	4	0	0	104.19	90.72	0.00
G09	20	78	1,431	3	1	0	0	49.13	30.18	0.00
B-Alpha	50	60	294	3	1	0	0	17.18	9.84	0.00
B-OTC	54	76	3,633	4	1	0	0	67.38	38.39	0.00
AS01	54	60	922	3	2	0	0	73.27	44.99	0.00
AS02	62	80	1,642	3	2	0	0	104.54	47.17	0.00
Ning	45	106	3,140	5	1	0	0	360.93	335.74	0.00
Hamsterster	25	82	614	4	0	0	1	43.33	40.20	0.00
Escorts	37	91	486	3	1	0	0	106.75	100.68	0.00
Anybeat	83	99	19,374	4	2	0	0	507.91	224.23	0.00
Advogato	62	106	5,868	6	2	0	0	2121.36	2027.90	0.00
Gplus	124	5	1	5	1	0	0	23.25	17.95	0.00
Facebook1	116	1	1	3	1	0	0	48.98	45.28	0.00
Facebook2	10	1	1	4	2	0	0	0.28	0.22	0.00

Although during initialization of the master relaxation we add star based constraints only for the top 20% of vertices by degree, as described in Section 5.1, the number of violated constraints that are added on-the-fly is never more than 4 (G08 when $\alpha = 0.5$). As it can be seen under the columns Star and Leaf in the tables, in the vast majority of instances the largest 2-club found in

Table 11 Results for Group-2 instances with $s=2$ and $\alpha=0.5$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Leaf	Regular	Total time (s)	s-club time (s)	Gap (%)
G04	26	47	479	4	1	0	0	158.12	150.56	0.00
G05	63	31	1,909	4	1	0	0	127.67	115.26	0.00
G06	69	31	2,577	3	1	0	0	80.47	53.41	0.00
G08	75	26	476	3	1	0	0	41.43	37.97	0.00
G09	72	30	3,052	3	1	0	0	54.92	46.99	0.00
B-Alpha	81	41	36,709	6	2	0	0	160.82	20.72	0.00
B-OTC	82	54	37,748	4	1	0	0	239.13	21.59	0.00
AS01	73	44	4,687	3	1	0	0	94.28	47.02	0.00
AS02	96	58	31,063	8	2	0	2	674.79	171.61	0.00
Ning	93	75	99,359	4	1	0	0	908.55	171.84	0.00
Hamsterster	60	51	53,082	8	1	0	3	213.56	75.75	0.00
Escorts	79	65	18,715	3	1	0	0	212.38	80.37	0.00
Anybeat	118	71	321,054	6	3	0	0	TL	224.25	1.00
Advogato	102	80	154,273	9	3	0	0	2622.67	1971.76	0.00
Gplus	129	2	1	4	2	0	0	19.54	16.87	0.00
Facebook1	116	1	1	5	2	0	0	86.24	80.73	0.00
Facebook2	10	1	1	4	2	0	0	0.29	0.22	0.00

the interdicted graph is frequently a star and our heuristic never added a constraint using just the leaves detected in H.

Similar to Group-1 instances, the interdiction problem becomes more difficult to solve for smaller values of α, and the number of BC nodes explored and the running time increase noticeably. As an example, the number of explored nodes for instance `Anybeat` increases from 136 when $\alpha=2$ to $321,054$ when $\alpha=0.5$. Moreover, the average running time for the 16 instances that are solved to optimality, increases from 116 seconds to 356 seconds as α decreases from 2 to 0.5.

For $s=3$, as mentioned before, solving the maximum s-club problem to optimality is too time-consuming for instances in Group-2 (See Table 2). Therefore, we use an inexact approach to solve the separation problem to find a sufficiently violated constraint (i.e., corresponding s-club) instead of finding a maximum s-club. Given an integral feasible solution $(\hat{\theta},\hat{x})$ to the master relaxation, instead of finding a maximum s-club in the graph interdicted according to $\hat{x}$, we look for an s-club with cardinality at least $\hat{\theta}+\epsilon$ where ϵ is the minimum violation we seek in the constraint.

In this inexact separation approach, first we rely on the greedy heuristic built into the ICUT solver to detect a sufficiently large s-club. If this heuristic s-club size is at least $\hat{\theta}+\epsilon$, the separation call is terminated early and the corresponding violated constraint is added to the master problem. If the heuristic s-club is not sufficiently large, the exact Gurobi BC algorithm in the ICUT solver is run with a termination condition based on a target objective value. In this setting, the solver stops once it finds an s-club of size at least $\hat{\theta}+\epsilon$. If neither of the above two conditions results in early termination of ICUT, we let it continue to solve the separation problem to optimality. In this case, it will terminate either returning a maximum s-club with violation, i.e., of size greater than $\hat{\theta}$ and smaller than $\hat{\theta}+\epsilon$; or certifying that no violated constraint exists. Note that by design, on our test bed ICUT subproblems do not reach their termination by time limit. After experimentation with $\epsilon = 1.5, 2.5$, and 5 in this inexact separation approach (see Appendix B.3), we chose to employ a minimum constraint violation target of 1.5 for early termination of a separation call.

Moreover, instead of solving the minimum latency-s CDS problem to optimality, we use the following method that is analogous to Algorithm 2 to heuristically find a hereditary subset of the violated 3-club: if a 3-club S contains an edge $\{u,v\}$ such that $\deg_{G[S]}(u) + \deg_{G[S]}(v) - |c_{uv}| = |S|$ where $|c_{uv}|$ is the number of common neighbors of vertices u and v, then $\{u,v\} \subseteq S$ is a minimum latency-3 CDS of $G[S]$ and S is a H-hereditary 3-club for $H = S \setminus \{u,v\}$. Otherwise, we set $H = \{u \in S \mid \deg_{G[S]}(u) = 1\}$. Table 12 shows the results of these experiments for $\alpha = 2$. As it can be seen, only 3 instances `Gplus`, `Facebook1`, and `Facebook2` are solved to optimality within the time limit. We should remind the reader here that all separation calls terminated conclusively even though the *cumulative* separation time exceeds one hour in these instances. Since our previous experiments show that the interdiction problem becomes more difficult to solve on this test bed as the value of α decreases, we have not conducted experiments for $\alpha = 1$ and $\alpha = 0.5$.

Table 12 Results for Group-2 instances with $s = 3$ using inexact separation.

Graph G	$x(V)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)	Gap (%)
G04	10,876	0	1	5	4	TL	4,967.81	0.00	99.62
G05	8,846	0	1	9	8	TL	4,671.76	0.00	99.48
G06	8,717	0	1	87	86	TL	6,325.01	0.03	99.44
G08	5,672	9	1	135	133	TL	3,493.11	0.17	98.97
G09	8,114	0	1	57	56	TL	4,449.13	0.06	99.33
B-Alpha	3,783	0	1	117	116	TL	78.47	0.31	98.02
B-OTC	5,295	8	1	90	88	TL	148.10	0.24	98.32
AS01	62	82	1,050	125	122	TL	195.56	0.06	15.74
AS02	10,900	0	1	74	73	TL	178.72	0.52	99.11
Ning	10,298	0	1	47	46	TL	442.33	0.98	99.00
Hamsterster	1,674	7	18,099	3,386	3,384	TL	2,777.36	9.15	95.26
Escorts	10,106	0	1	5	3	TL	4,002.30	0.00	99.20
Anybeat	12,645	0	1	25	24	TL	190.87	0.73	99.10
Advogato	6,551	0	1	46	45	TL	559.49	1.16	98.33
Gplus	100	41	380	8	5	2405.92	53.99	0.07	0.00
Facebook1	116	1	1	5	3	318.07	170.60	0.07	0.00
Facebook2	10	1	1	6	4	2.50	0.63	0.00	0.00
Douban	154,908	0	1	4	3	TL	4,261.36	0.03	99.92

The results for the Group-2 instances reinforce the conclusions from our experiments with Group-1, that for $s = 3$ the interdiction problem becomes much more challenging to solve.

7. Conclusions

In this paper we proposed an interdiction model to minimize the maximum cardinality of an s-club in the given graph. Motivated by account suspension practices in online social networks, we assume a penalty in the objective function for each vertex interdicted rather than assuming a hard budget constraint. By introducing the concept of H-hereditary s-clubs we derived a better MILP formulation with fewer constraints (when compared to an MILP formulation derived from standard interdiction techniques) and derive results about the polyhedral structure of its LP relaxation and of the convex hull of its feasible solutions. We show that the hereditary subset H inside an H-hereditary s-club can be found equivalently as a latency-s CDS of the s-club. Using these results we design a delayed constraint generation branch-and-cut algorithm for the interdiction problem that identifies violated constraints by solving a maximum s-club problem and a minimum latency-s CDS problem during separation. Our computational studies show that our algorithm can solve the s-club interdiction problem over well-known benchmark instances with more than 10,000 vertices in a few minutes and that it significantly outperforms a similar algorithm that is based only on the naive MILP formulation of the interdiction problem, especially when $s = 2$. The 3-club interdiction problem is still quite challenging to solve on the second group of instances in our test-bed on which solving the NP-hard maximum 3-club problem remains difficult.

Given the importance of conclusive termination during separation calls to the correctness of such a relaxation based decomposition branch-and-cut scheme, further breakthroughs are needed to solve the maximum s-club and minimum latency-s CDS problems on this test bed for $s \geq 3$. The master relaxation also needs further investigation and strengthening, especially for $s \geq 3$, to shift the computational burden away from the separation procedures to the extent possible. These developments and improved inexact separation procedures for $s \geq 3$ can further extend our ability to solve the s-club interdiction on even larger scale social networks.

Acknowledgments

The authors would like to thank Austin Buchanan, Hosseinali Salemi, and Hamidreza Validi for providing us the codes used in (Salemi and Buchanan 2020, Validi and Buchanan 2020) and freely offering their help to integrate it with our code. The availability of these effective implementations were integral to our work. The research of the first and second authors was supported in part by the Office of Naval Research Grant ONR–N00014-19-1-2329. The authors are grateful to the two anonymous reviewers who provided constructive feedback that helped us improve the content and presentation of this paper.

Author Biographies

Niloufar Daemi is a third-year Ph.D. student in the School of Industrial Engineering and Management (IEM) at Oklahoma State University (OSU). She received her B.Sc. in Industrial Engineering from Noshirvani University of Technology in 2012 and M.Sc. in Financial Engineering from Amirkabir University of Technology in 2016. Her research focuses on interdiction problems in networks.

Juan S. Borrero is an Assistant Professor in IEM at OSU. He received his B.Sc. in Math and his M.Sc. in Industrial Engineering from University of Los Andes, Colombia, and his Ph.D. in Industrial Engineering from the University of Pittsburgh in 2017. His research interests include interdiction and bilevel programming problems, robust optimization, and sequential decision-making problems under uncertainty and limited information.

Balabhaskar Balasundaram is the Wilson Bentley Professor in IEM at OSU. He received his B.Tech in Mechanical Engineering from the Indian Institute of Technology-Madras in 2002, and his Ph.D. in Industrial Engineering from Texas A&M University in 2007. His research interests include combinatorial optimization and integer programming, especially graph-theoretic models and their applications in social network analysis and computational biology.

The story behind this paper. Niloufar Daemi is a Ph.D. student of Juan Borrero. Baski Balasundaram has worked on a variety of clique relaxation over the years. Baski started getting involved in the discussions, when Niloufar and Juan started working on an interdiction-based approach to formulating the maximum s-club problem. Baski and Juan had worked on a budget constrained version of the problem before and had some experience with its challenges. In the Summer of 2019, things came together and the three of us started working on interdicting s-clubs. However, with the frequent news around that time about mass deactivation of fake/bot accounts by Facebook and Twitter, a hard budget constraint on the interdiction set seemed too restrictive and artificial. With that change, we had the essential ingredients of the study reported in this paper.

A significant theoretical breakthrough came about in Fall 2019 when we were trying to find ways to write stronger constraints than the big-M constraint $\theta \geq |S|(1-x(S))$. Juan formulated constraints based on stars, using both linear and quadratic expressions initially, that took advantage of the fact that deleting leaves preserves s-club property–a kind of pseudo-heredity. From there, after extensive preliminary computational studies, we concluded that linear constraints were more effective in formulating and solving this problem. We then extended the approach to edge stars, and eventually, hit upon the general concept of H-hereditary s-clubs. The early computational studies based on these new constraints were significantly better. It was apparent that we had something very effective. Most of our technical results on the related polyhedra came about in the later stages through our attempts to demonstrate the strength of these inequalities theoretically. Our only disappointment is Remark 3, which originally started out as a conjecture that $P \subset P'$, which turned out not to be true!

References

Alba RD (1973) A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology* 3(1):113–126.

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

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

Balasundaram B, Pajouh FM (2013) Graph theoretic clique relaxations and applications. Pardalos PM, Du DZ, Graham R, eds., *Handbook of Combinatorial Optimization*, 1559–1598 (New York: Springer), 2nd edition, ISBN 978-1-4419-7996-4.

Barabási AL, Albert R (1999) Emergence of scaling in random networks. *Science* 286(5439):509–512.

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

Butenko S, Wilhelm W (2006) Clique-detection models in computational biochemistry and genomics. *European Journal of Operational Research* 173:1–17.

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

Daemi N, Borrero JS, Balasundaram B (2021a) Decomposition branch-and-cut solver for 2-club interdiction. C++ Codes online at: `https://github.com/niloufardaemi/2club_interdiction`, Accessed January 6, 2022.

Daemi N, Borrero JS, Balasundaram B (2021b) Decomposition branch-and-cut solver for 3-club interdiction. C++ Codes online at: `https://github.com/niloufardaemi/3club_interdiction`, Accessed January 6, 2022.

Dimacs (2012) Graph Partitioning and Graph Clustering: Tenth Dimacs Implementation Challenge. `http://www.cc.gatech.edu/dimacs10/index.shtml`, accessed Feb 2015.

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

Fischetti M, Ljubić I, Monaci M, Sinnl M (2019) Interdiction games and monotonicity, with application to knapsack problems. *INFORMS Journal on Computing* 31(2):390–410.

Fischetti M, Monaci M, Sinnl M (2018) A dynamic reformulation heuristic for generalized interdiction problems. *European Journal of Operational Research* 267(1):40–51.

Furini F, Ljubić I, Martin S, San Segundo P (2019) The maximum clique interdiction problem. *European Journal of Operational Research* 277(1):112–127.

Gurobi Optimization, LLC (2021) Gurobi Optimizer Reference Manual. URL `http://www.gurobi.com`.

Israeli E, Wood RK (2002) Shortest-path network interdiction. *Networks: An International Journal* 40(2):97–111.

Kunegis J (2013) Konect: the Koblenz network collection. *Proceedings of the 22nd International Conference on World Wide Web*, 1343–1350.

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

Lesser O, Tenenboim-Chekina L, Rokach L, Elovici Y (2013) Intruder or welcome friend: Inferring group membership in online social networks. *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, 368–376 (Springer).

Lu Y, Moradi E, Balasundaram B (2018) Correction to: Finding a maximum k-club using the k-clique formulation and canonical hypercube cuts. *Optimization Letters* 12(8):1959–1969, URL `http://dx.doi.org/DOI10.1007/s11590-018-1273-7`.

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

Moon JW, Moser L (1965) On cliques in graphs. *Israel Journal of Mathematics* 3:23–28.

Moradi E, Balasundaram B (2018) Finding a maximum k-club using the k-clique formulation and canonical hypercube cuts. *Optimization Letters* 12(8):1947–1957, URL `http://dx.doi.org/DOI10.1007/s11590-015-0971-7`.

Morton D, Pan F, Saeger K (2007) Models for nuclear smuggling interdiction. *IIE Transactions on Operations Engineering* 38:3–14.

Newman M (2003) The structure and function of complex networks. *SIAM Review* 45:167–256.

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

Pan F, Charlton W, Morton D (2003) Interdicting smuggled nuclear material. Woodruff D, ed., *Network Interdiction and Stochastic Integer Programming*, 1–20 (Kluwer Academic Publishers, Boston).

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

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

Raghavan S, Zhang R (2019) A branch-and-cut approach for the weighted target set selection problem on social networks. *INFORMS Journal on Optimization* 1(4):304–322.

Rossi RA, Ahmed NK (2015) The network data repository with interactive graph analytics and visualization. *AAAI*, URL `http://networkrepository.com`.

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

Smith JC, Song Y (2020) A survey of network interdiction models and algorithms. *European Journal of Operational Research* 283(3):797–811.

Spangler T (2018) Twitter stock slides on report that it has been deleting over 1 million fake accounts daily. `https://variety.com/2018/digital/news/twitter-stock-deleted-fake-accounts-1202868405/`.

Validi H, Buchanan A (2020) The optimal design of low-latency virtual backbones. *INFORMS Journal on Computing* 32(4):952–967.

Wasserman S, Faust K (1994) *Social Network Analysis* (New York: Cambridge University Press).

Appendix A: Proofs of Technical Results

A.1. Proof of Lemma 1

Consider an arbitrary $U \in \mathcal{C}(S,H)$ and suppose $U = S \setminus T$ for some $T \subseteq H$. Clearly, $S \setminus H \subseteq U$. Suppose $x(S \setminus H) \geq 1$. Then, we also have $x(U) \geq 1$. By definition of the set $\Lambda(S,H)$ we know that $\theta \geq 0$, and hence (θ, x) satisfies (7).

Now suppose $x(S \setminus H) = 0$. Then by definition (5), the point (θ, x) satisfies:

$$\theta \geq |S| - x(H) = |U| + |T| - x(H \cap T) - x(H \cap U),$$

because U and T partition S which contains H. As $|T| - x(H \cap T) \geq 0$, it follows that (θ, x) satisfies $\theta \geq |U| - x(H \cap U)$. Again, as $S \setminus H$ and H partition S which contains U, we know that

$$x(U) = x((S \setminus H) \cap U) + x(H \cap U) = x(H \cap U),$$

because $x(S \setminus H) = 0$. Hence, the point (θ, x) satisfies $\theta \geq |U| - x(U) \geq |U| - |U|x(U)$ as claimed. □

A.2. Proof of Lemma 2

Let $(\theta, x) \in \Lambda(S,H)$. If $x(S \setminus J) \geq 1$, we have $|S| - x(J) - |S|x(S \setminus J) \leq 0$ and $\theta \geq 0$. Hence, $(\theta, x) \in \Lambda(S,J)$. Now suppose $x(S \setminus J) = 0$. Then, it follows that $x(S \setminus H) = 0$ and $x(H \setminus J) = 0$ as $J \subset H \subseteq S$. Hence, (θ, x) satisfies $\theta \geq |S| - x(H) \geq 0$. Because $x(H \setminus J) = 0$, it also implies that $\theta \geq |S| - x(J) \geq 0$ and $(\theta, x) \in \Lambda(S,J)$, as desired. □

A.3. Proof of Proposition 1

We prove that any feasible solution of (14) is feasible to (3) and vice versa. First notice that Lemmas 1, 2, and 3 imply that any feasible solution of (14) is feasible to (3). Now, suppose (θ, x) is feasible to (3), which implies that $\theta \geq \bar{\omega}_s(G \setminus T^x) \geq |S'| - |S'|x(S')$ for all $S' \in \mathcal{S}$. Consider $S \in \mathcal{C}^*$ and $H \in \mathcal{H}^*(S)$, chosen arbitrarily, and define $r(S,H,x) = |S| - x(H) - |S|x(S \setminus H)$. Observe that the claim is proven if we can show that $\theta \geq r(S,H,x)$. We consider the following three cases:

(i) $S \subseteq V \setminus T^x$: No vertex of S is interdicted in this case and hence, $x(H) = x(S \setminus H) = 0$ and $r(S,H,x) = |S|$. Because $S \in \mathcal{S}$, we have that $\theta \geq \bar{\omega}_s(G \setminus T^x) \geq |S|$ and the claim holds.

(ii) $(S \setminus H) \cap T^x \neq \emptyset$: At least one of the vertices interdicted by x belongs to $S \setminus H$. In this case, $x(S \setminus H) \geq 1$, which implies that $r(S,H,x) \leq 0$, and the claim holds.

(iii) $(S \setminus H) \cap T^x = \emptyset$ and $H \cap T^x \neq \emptyset$: Because any vertex in S interdicted by x belongs to H, we know that $S \setminus T^x$ is an s-club in $G \setminus T^x$, and it follows that $\theta \geq \bar{\omega}_2(G \setminus T^x) \geq |S \setminus T^x|$. In this case, $r(S,H,x) = |S| - |H \cap T^x| = |S \setminus (H \cap T^x)|$. As $(S \setminus H) \cap T^x = \emptyset$, we have $S \cap T^x = H \cap T^x$ and $S \setminus (H \cap T^x) = S \setminus T^x$. Therefore, $r(S,H,x) = |S \setminus T^x|$ and the claim holds.

Thus, we can conclude that any feasible solution of (3) is feasible to (14). □

A.4. Proof of Proposition 2

Suppose that an s-club U is not critical. Then, there exists another s-club S and a non-empty $H \in \mathcal{H}^*(S)$, such that $U \in \mathcal{C}(S,H)$. In particular, $U = S \setminus T$ for some non-empty $T \subseteq H$ and U is also an s-club because S is an H-hereditary s-club. Now, for a vertex $v \in T$ and consider $U' = U \cup \{v\}$, distinct from U by construction. Note that $U' = (S \setminus T) \cup \{v\} = S \setminus (T \setminus \{v\})$ is also an s-club because $T \setminus \{v\} \subseteq H$ and S is H-hereditary. Then, it follows that U is not one-step maximal.

Conversely, if U is an s-club that is not one-step maximal, then there exists some vertex $v \in V \setminus U$ such that $U \cup \{v\}$ is an s-club. Then, $U \cup \{v\}$ is a $\{v\}$-hereditary s-club. Hence, $U \in \mathcal{C}(U \cup \{v\}, \{v\})$ and is therefore not critical. □

A.5. Results Needed to Prove Proposition 3

The LP relaxation P of Formulation (14) is full dimensional because $(\theta = |V|, x_v = 1/|V| \colon v \in V) \in$ interior(P). Consider an $S \in \mathcal{C}^*$ and $H \in \mathcal{H}^*(S)$ that define the face $F(S,H)$ of the polyhedron P given by the corresponding (S,H)-constraint (14b), that is,

$$F(S,H) := \{(\theta, x) \in P \mid \theta = r(S,H,x)\}. \tag{24}$$

where we recall that $r(S,H,x) = |S| - x(H) - |S|x(S \setminus H)$.

Lemma 4. *The face $F(S,H)$ in equation (24) is not contained within any of the following faces of P: $\{(\theta,x) \in P \mid \theta = 0\}$ and $\{(\theta,x) \in P \mid x_v = i\}$ for each $v \in V$ and $i \in \{0,1\}$.*

Proof. Define $\theta' = |S| - |H|$, $x'_v = 0$ if $v \in S \setminus H$ and $x'_v = 1$ if $v \in (V \setminus S) \cup H$. Note that (θ', x') must belong to P. For any $S' \in \mathcal{C}^*$ and $H' \in \mathcal{H}^*(S')$, if $x'(S' \setminus H') \geq 1$, then $r(S', H', x') \leq 0 \leq \theta'$. On the other hand if $x'(S' \setminus H') = 0$, then $S' \setminus H' \subseteq S \setminus H$, and $\theta' = |S| - |H| \geq |S'| - |H'| = r(S', H', x')$. As $r(S,H,x') = \theta'$, we know that $(\theta', x') \in F(S,H)$. Moreover, point (θ', x') is not in the $(x_v = 1)$-face of $v \in S \setminus H$ and not in the $(x_v = 0)$-face of $v \in (V \setminus S) \cup H$.

Now consider another point defined as $\tilde{\theta} = |S|$, $\tilde{x}_v = 0$ if $v \in S$ and $\tilde{x}_v = 1$ if $v \in V \setminus S$. Note that $(\tilde{\theta}, \tilde{x}) \in F(S,H)$ based on similar arguments. The point $(\tilde{\theta}, \tilde{x})$ is not contained in the $(x_v = 1)$-face of $v \in S$, not contained in the $(x_v = 0)$-face of $v \in V \setminus S$, and not contained in the $(\theta = 0)$-face as $|S| \geq 1$. Next we show that $F(S,H)$ can neither belong to the $(x_v = 0)$-face for $v \in S \setminus H$ nor to the $(x_v = 1)$-face for $v \in V \setminus S$ to complete the proof.

For any $U \in \mathcal{C}^*$, $J \in \mathcal{H}^*(U)$, and $x \in [0,1]^{|V|}$, we know that $r(U, J, \tilde{x}) \leq |S| = \tilde{\theta}$ for any $U \in \mathcal{C}^*$, $J \in \mathcal{H}^*(U)$ as $(\tilde{\theta}, \tilde{x}) \in P$. If in addition $U \neq S$, we claim that $r(U, J, \tilde{x}) \leq |S| - 1$. Indeed, if $U \subset S$ or $U \cap S = \emptyset$ then the claim follows from the definition of $\tilde{x}$. Thus, suppose that $U \cap S \neq \emptyset$ and $U \setminus S \neq \emptyset$.

If $U \cap S \subset S$, i.e., $S \setminus U$ is non-empty, then

$$r(U, J, \tilde{x}) = |U| - |J \setminus S| - |U| \times |(U \setminus J) \setminus S| \leq |U| - |J \setminus S| - |(U \setminus J) \setminus S| = |U \cap S| \leq |S| - 1.$$

Now suppose $S \subset U$. We also know that $U \setminus J \neq \emptyset$, as otherwise U is a clique that contains S, which contradicts $S \in \mathcal{C}^*$. If in addition, $(U \setminus J) \cap (V \setminus S) = \emptyset$, it follows that $U \setminus J \subseteq S$. Consider the following relationships: $U \setminus J \subseteq S \subset U$, which implies that $U \setminus S \subseteq J$. Hence, S can be obtained from U by deleting

$U \setminus S \in \mathcal{H}(U)$, a contradiction to $S \in \mathcal{C}^*$. Therefore, if $S \subset U$ it must be the case that $(U \setminus J) \cap (V \setminus S) \neq \emptyset$. If so, we obtain $r(U,J,\tilde{x}) \leq 0 \leq |S|-1$, as desired. So the claim holds.

We are now ready to demonstrate a point $(\hat{\theta},\hat{x}) \in F(S,H)$ that is not contained in the $(x_v = 0)$-face for an arbitrarily chosen $v \in S \setminus H$. Define $\hat{x}_u = \tilde{x}_u \ \forall u \neq v$ and with $\hat{x}_v = 1/|S|$; let $\hat{\theta} = |S|-1$. Then, $r(S,H,\hat{x}) = |S|-1$ and therefore $(\hat{\theta},\hat{x})$ satisfies the equality constraint in $F(S,H)$. On the other hand, as $\hat{x} > \tilde{x}$ we obtain $r(U,J,\hat{x}) \leq r(U,J,\tilde{x})$ for any $U \in \mathcal{C}^*$ and $J \in \mathcal{H}^*(U)$. Therefore, as $r(U,J,\tilde{x}) \leq |S|-1 = \hat{\theta}$, we conclude that $\hat{\theta} \geq r(U,J,\hat{x})$ for any $U \in \mathcal{C}^*$ and $J \in \mathcal{H}^*(U)$. In other words, $(\hat{\theta},\hat{x})$ belongs to $F(S,H)$ but it does not belong into the face of P induced by $x_v = 0$.

Now we demonstrate a point $(\bar{\theta},\bar{x}) \in F(S,H)$ that is not contained in the $(x_v = 1)$-face for an arbitrarily chosen $v \in V \setminus S$. Consider the same $\tilde{x}$ as in the previous case and define $\bar{x}_u = \tilde{x}_u$ for each $u \neq v$ and set $\bar{x}_v = 1-\epsilon$, where the positive constant $\epsilon < 1/|V|$, and let $\bar{\theta} = \tilde{\theta} = |S|$. Observe that $r(S,H,\bar{x}) = r(S,H,\tilde{x})$ and therefore $(\bar{\theta},\bar{x})$ satisfies the constraint defining $F(S,H)$ at equality. Similarly, $r(U,J,\bar{x}) = r(U,J,\tilde{x})$ for any s-club U that does not contain vertex v. Hence, if $v \notin U$, $(\bar{\theta},\bar{x})$ satisfies the corresponding constraint $\theta \geq r(U,J,x)$. If $v \in J$, then $r(U,J,\bar{x}) = r(U,J,\tilde{x}) + \epsilon \leq |S|-1+1/|V| < |S| = \bar{\theta}$, thus $(\bar{\theta},\bar{x})$ satisfies the constraint $\theta \geq r(U,J,x)$. Finally, if $v \in U \setminus J$, then $r(U,J,\bar{x}) = r(U,J,\tilde{x}) + \epsilon|U| \leq |S|-1+1 = |S| = \bar{\theta}$. Again $(\bar{\theta},\bar{x})$ satisfies the constraint $\theta \geq r(U,J,x)$ if $v \in U \setminus J$. Hence, $(\bar{\theta},\bar{x}) \in F(S,H)$, but it does not belong to the face induced by $x_v = 1$. Hence, $F(S,H)$ is not contained within any of the trivial faces of P. □

A.6. Proof of Proposition 3

Consider $\hat{S} \in \mathcal{C}^*$ and $\hat{H} \in \mathcal{H}^*(\hat{S})$ also chosen arbitrarily such that $(S,H) \neq (\hat{S},\hat{H})$. We claim that there exists a point $(\tilde{\theta},\tilde{x}) \in F(S,H)$ such that $(\tilde{\theta},\tilde{x}) \notin F(\hat{S},\hat{H})$; this assertion in conjunction with Lemma 4 would yield the desired result. This is because, if $F(S,H) \setminus F(\hat{S},\hat{H}) \neq \emptyset$, we know that the face $F(S,H)$ cannot be completely contained in the face $F(\hat{S},\hat{H})$. Since the latter is arbitrary, it shows that no other inequality (14b) induces a face of P that contains $F(S,H)$. Therefore, $F(S,H)$ must be maximal.

First, we assume that $\hat{S} = S$. It then follows that $\hat{H} \neq H$, which in turn implies that $S \setminus H \neq \emptyset$; recall that if $S = H$, then $S = \hat{S}$ must be a clique, in which case $\mathcal{H}^*(\hat{S}) = \{\hat{S}\}$ as it only contains maximal members. Consider the point constructed as follows: $\tilde{\theta} = |S| - |H|$, $\tilde{x}_v = 0$ if $v \in S \setminus H$ and $\tilde{x}_v = 1$ if $v \in (V \setminus S) \cup H$. Note that $(\tilde{\theta},\tilde{x}) \in F(S,H)$ as the defining inequality is active at $(\tilde{\theta},\tilde{x})$ and the point belongs to P (easy to verify). Now, because $H, \hat{H} \in \mathcal{H}^*(S)$, then H is not contained in $\hat{H}$ and vice versa. This observation implies that $S \setminus \hat{H}$ is not contained in $S \setminus H$, consequently $\tilde{x}(S \setminus \hat{H}) \geq 1$. Therefore, $|\hat{S}| - \tilde{x}(\hat{H}) - |\hat{S}|\tilde{x}(\hat{S} \setminus \hat{H}) \leq 0$ and $\tilde{\theta} > 0$, which implies that $(\tilde{\theta},\tilde{x}) \notin F(\hat{S},\hat{H})$.

Now we assume that $S \neq \hat{S}$ and consider the following point: $\tilde{\theta} = |S|$, $\tilde{x}_v = 0$ if $v \in S$ and $\tilde{x}_v = 1$ if $v \in V \setminus S$. Note that $(\tilde{\theta},\tilde{x}) \in F(S,H)$. Suppose that $\hat{S} \setminus \hat{H}$ is not contained in S. Then, $\tilde{x}(\hat{S} \setminus \hat{H}) \geq 1$ and therefore $|\hat{S}| - \tilde{x}(\hat{H}) - \tilde{x}(\hat{S} \setminus \hat{H}) \leq 0$, which implies that $(\tilde{\theta},\tilde{x}) \notin F(\hat{S},\hat{H})$ as $\tilde{\theta} = |S| \geq 1$. Next consider the case where $\hat{S} \setminus \hat{H} \subseteq S$. Because $S, \hat{S} \in \mathcal{C}^*$, by the definition of $\mathcal{C}^*$ we know that $\hat{S} \setminus \hat{H} \neq S$; hence, the containment must be strict. Now, partition $\hat{H}$ as $\hat{H} = \hat{H}_1 \cup \hat{H}_2$, where $\hat{H}_1 = \hat{H} \setminus S$ and $\hat{H}_2 = \hat{H} \cap S$. From the fact that $\hat{S} \setminus \hat{H} \subset S$, it follows that the right-hand side of the constraint inducing face $F(\hat{S},\hat{H})$ evaluated at $(\tilde{\theta},\tilde{x})$ becomes:

$$|\hat{S}| - \tilde{x}(\hat{H}) - \tilde{x}(\hat{S} \setminus \hat{H}) = |\hat{S}| - \tilde{x}(\hat{H}) = |\hat{S}| - |\hat{H}_1|.$$

Now, we claim that $|\hat{S}| - |\hat{H}_1| < |S|$. Suppose, for the sake of contradiction that this is not the case, i.e, $|\hat{S}| - |\hat{H}_1| = |S|$. Then, as $\hat{S} \setminus \hat{H}_1 \subseteq S$ this would imply that $\hat{S} \setminus \hat{H}_1 = S$. However, this would contradict the definition of $\mathcal{C}^*$ as $S \in \mathcal{C}^*$. Therefore, $|\hat{S}| - |\hat{H}_1| < |S| = \tilde{\theta}$ implying that $(\tilde{\theta}, \tilde{x}) \notin F(\hat{S}, \hat{H})$, which completes the proof. □

A.7. Elementary Properties of Convex Hull $\mathcal{P}$

LEMMA 5. *Consider a non-empty graph $G = (V, E)$ and positive integer s. The convex hull of feasible solutions to Formulation* (14)*, denoted by $\mathcal{P}$, is full dimensional.*

Proof. Let e_v denote the $|V|$-dimensional unit vector with v-th component at one. It is easy to verify that the following $|V| + 2$ points in $\mathcal{P}$, $(\theta = |V|, x = e_v) \in \mathcal{P}$ for each $v \in V$, $(\theta = |V|, x = \mathbf{0})$, and $(\theta = 0, x = \mathbf{1})$, are affinely independent. □

LEMMA 6. *Consider a non-empty graph $G = (V, E)$ and positive integer s. The valid inequality $x_v \geq 0$ induces a facet of $\mathcal{P}$ for each $v \in V$.*

Proof. Let the "zero face" corresponding to vertex v be denoted as $F_v \coloneqq \{(\theta, x) \in \mathcal{P} \mid x_v = 0\}$. As $\dim(\mathcal{P}) = |V| + 1$, we demonstrate the same number of affinely independent points contained in F_v to establish this claim. The following can be easily verified as affinely independent points: $(\theta = |V|, x = e_u) \in \mathcal{P}$ for each $u \in V \setminus \{v\}$, $(\theta = |V|, x = \mathbf{0})$, and $(\theta = 1, x = \mathbf{1} - e_v)$.

□

A.8. Proof of Theorem 1

Consider and s-club $S \in \mathcal{C}^*$ and an $H \in \mathcal{H}(S)$. Note that $S \setminus H$ may be empty if $S = H$ is a clique. We know from Lemma 6 that $\mathcal{P}^{S \setminus H} \coloneqq \mathcal{P} \cap \{(\theta, x) \mid x_v = 0 \ \forall v \in S \setminus H\}$ is a face of $\mathcal{P}$, and hence,

$$\dim(\mathcal{P}^{S \setminus H}) = \dim(\mathcal{P}) - |S \setminus H|.$$

The inequality $\theta \geq |S| - x(H)$ is valid for $\mathcal{P}^{S \setminus H}$ because $x_v = 0$ for every $v \in S \setminus H$ and S is H-hereditary. The following collection of $\dim(\mathcal{P}^{S \setminus H})$ points can be easily verified to be contained in the face,

$$F_Q \coloneqq \{(\theta, x) \in \mathcal{P}^{S \setminus H} \mid \theta = |S| - x(H)\}.$$

1. Construct the first $|H|$ points $(\hat{\theta}, \hat{x})$ for every vertex $u \in H$ where $\hat{\theta} = |S| - 1$ and $\hat{x}$ is defined as
$$\hat{x}_v = \begin{cases} 1, & \text{if } v = u, \\ 1, & \text{if } v \in V \setminus S, \\ 0, & \text{if } v \in S \setminus \{u\}. \end{cases}$$
2. Construct the next $|V \setminus S|$ points $(\hat{\theta}, \hat{x})$ for every vertex $u \in V \setminus S$ where $\hat{\theta} = |S|$ and $\hat{x}$ is defined as
$$\hat{x}_v = \begin{cases} 0, & \text{if } v \in S \cup \{u\} \\ 1, & \text{if } v \in V \setminus (S \cup \{u\}). \end{cases}$$
Note that because S is a critical s-club, $S \cup \{u\}$ cannot be an s-club based on Proposition 2.
3. Finally consider the point, $(\hat{\theta}, \hat{x})$ where $\hat{\theta} = |S|$ and $\hat{x}$ defined as
$$\hat{x}_v = \begin{cases} 0, & \text{if } v \in S \\ 1, & \text{if } v \in V \setminus S. \end{cases}$$

The foregoing $\dim(\mathcal{P}^{S \setminus H}) = |V| + 1 - |S| + |H|$ points can be verified to be affinely independent, establishing our claim. □

A.9. Proof of Theorem 2

Validity of inequality (18) is easy to see as the clique S is hereditary under vertex deletion and it is an s-club for every $s \geq 2$. We prove that the face F' of $\mathcal{P}$ induced by inequality (18) is maximal. That is,

$$F' \coloneqq \{(\theta, x) \in \mathcal{P} \mid \theta + x(S) = |S|\}.$$

Consider an arbitrary proper face of $\mathcal{P}$ given by:

$$F \coloneqq \{(\theta, x) \in \mathcal{P} \mid a_0\theta + \sum_{i \in V} a_i x_i = b\},$$

which we assume contains F' in order to arrive at a contradiction.

Consider the following point: $\theta = |S|; x_u = 1\ \forall u \notin S; x_u = 0\ \forall u \in S$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$|S|a_0 + \sum_{i \notin S} a_i = b. \tag{25}$$

Now consider the following point for some $\ell \in S$: $\theta = |S| - 1; x_u = 1\ \forall u \in (V \setminus S) \cup \{\ell\}; x_u = 0\ \forall u \in S \setminus \{\ell\}$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(|S| - 1)a_0 + \sum_{i \notin S} a_i + a_\ell = b. \tag{26}$$

From equations (25) and (26), we can conclude that $a_\ell = a_0\ \forall \ell \in S$ and rewrite F as follows:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0 x(S) + \sum_{i \notin S} a_i x_i = b\}. \tag{27}$$

Finally, consider the following point for an arbitrary vertex $\ell \notin S$: $\theta = |S \setminus N_G(\ell)|$; $x_u = 1\ \forall u \in N_G(\ell) \cup [V \setminus (S \cup \{\ell\})]$; $x_u = 0\ \forall u \in \{\ell\} \cup S \setminus N_G(\ell)$. Because S is a *maximal* clique, vertex ℓ cannot be adjacent to every vertex in S. Hence, we know that $S \setminus N_G(\ell)$ is a *non-empty* clique. We also know that $\{\ell\} \cup S \setminus N_G(\ell)$ is not an s-club as vertex ℓ is isolated in the graph interdicted according to x. As $\ell \notin S$, we know that $x(S) = |S \cap N_G(\ell)|$, and hence $\theta + x(S) = |S|$, implying that $(\theta, x) \in F'$. Now, using equation (27) we can obtain the following equation as $F' \subseteq F$:

$$a_0|S \setminus N_G(\ell)| + a_0|S \cap N_G(\ell)| + \sum_{i \notin S \cup \{\ell\}} a_i = b. \tag{28}$$

From equations (25) and (28), we can conclude that $a_\ell = 0$ for each $\ell \notin S$ and $b = |S|a_0$. We now know that F has the following form:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0 x(S) = |S|a_0\}.$$

As F is a proper face, we know that $a_0 \neq 0$ and we can conclude that $F' = F$ is a maximal proper face. □

A.10. Proof of Theorem 3

Validity of inequality (19) follows from the observation that for any $x \in \{0,1\}^{|V|}$ and $\theta \in \mathbb{R}$, we know that $(\theta, x) \in \mathcal{P}$ if and only if $\theta \geq \bar{\omega}_s(G \setminus T^x)$. We know that if $x_v = 0$, $\bar{\omega}_s(G \setminus T^x) \geq \deg_G(v) + 1 - x(N_G(v))$ and the inequality is valid. If $x_v = 1$ and $N_G(v)$ is an independent set, the inequality imposes that $\theta \geq 1 - x(N_G(v))$, which is valid for all $x \in \mathcal{P}$.

As before, we show that inequality (19) induces a maximal face of $\mathcal{P}$. We define,

$$F' := \{(\theta, x) \in \mathcal{P} \mid \theta + x(N_G(v)) + d_v x_v = d_v + 1\},$$

where $d_v \equiv \deg_G(v)$. Consider an arbitrary proper face of $\mathcal{P}$ given by:

$$F := \{(\theta, x) \in \mathcal{P} \mid a_0\theta + \sum_{i \in V} a_i x_i = b\},$$

which we assume contains F'.

Consider the following point: $\theta = d_v + 1; x_u = 1\ \forall u \notin N_G[v]; x_u = 0\ \forall u \in N_G[v]$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(d_v + 1)a_0 + \sum_{i \notin N_G[v]} a_i = b. \tag{29}$$

Now consider the following point for some $\ell \in N_G(v)$: $\theta = d_v; x_u = 1\ \forall u \notin N_G[v]; x_\ell = 1; x_u = 0\ \forall u \in N_G[v] \setminus \{\ell\}$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$d_v a_0 + \sum_{i \notin N_G[v]} a_i + a_\ell = b. \tag{30}$$

From equations (29) and (30), we can conclude that $a_\ell = a_0\ \forall \ell \in N_G(v)$ and rewrite F as follows:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0 x(N_G(v)) + a_v x_v + \sum_{i \notin N_G[v]} a_i x_i = b\}.$$

Next we consider an arbitrary $\ell \notin N_G[v]$. As $N_G[v] \in \mathcal{C}^*$ is a one-step maximal (critical) s-club, we know that $N_G[v] \cup \{\ell\}$ cannot be an s-club; otherwise, we will contradict the definition of $\mathcal{C}^*$ (see Proposition 2). Hence, we consider the following point: $\theta = d_v + 1; x_u = 0\ \forall u \in N_G[v] \cup \{\ell\}; x_u = 1\ \forall u \notin N_G[v] \cup \{\ell\}$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(d_v + 1)a_0 + \sum_{i \notin N_G[v] \cup \{\ell\}} a_i = b. \tag{31}$$

Now from equations (29) and (31), we can conclude that $a_\ell = 0$ for each $\ell \notin N_G[v]$ and $b = (d_v + 1)a_0$. We now know that F has the following form:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0 x(N_G(v)) + a_v x_v = (d_v + 1)a_0\}.$$

Finally to identify the coefficient a_v, we consider the following point: $\theta = 1; x_u = 0\ \forall u \in N_G(v); x_u = 1\ \forall u \notin N_G(v)$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$a_0 + a_v = (d_v + 1)a_0. \tag{32}$$

Hence, $a_v = d_v a_0$. Because F is a proper face, we know that $a_0 \neq 0$ and thus we conclude that $F' = F$ is a maximal proper face. □

A.11. Proof of Theorem 4

Validity of inequality (20) follows from the observation that for any feasible solution $(\theta, x) \in \mathcal{P}$ of Formulation (14), $\theta \geq \bar{\omega}_s(G \setminus T^x)$. If $x_u = x_v = 0$, we know that $\bar{\omega}_s(G \setminus T^x) \geq \deg_G(u) + \deg_G(v) - c_{uv} - x(L_{uv})$ and the inequality is satisfied. If $x_u = x_v = 1$, we require $\theta \geq \min\{1, c_{uv}\} - x(L_{uv})$ which holds.

If $x_u = 1$ and $x_v = 0$, then $\theta \geq \deg_G(v) - x(L_{uv})$ which is valid. Finally, if $x_u = 0$ and $x_v = 1$, we require $\theta \geq \deg_G(u) - c_{uv} - x(L_{uv}) + \min\{1, c_{uv}\}$. Here, we consider two cases. If $c_{uv} = 0$, the inequality becomes $\theta \geq \deg_G(u) - x(L_{uv})$, and if $c_{uv} \geq 1$, the inequality becomes $\theta \geq \deg_G(u) - c_{uv} - x(L_{uv}) + 1$ and the inequality is valid in both cases.

Next, we show that the face of $\mathcal{P}$ induced by inequality (20) is maximal.

Let F' denote the face of $\mathcal{P}$ induced by inequality (20). That is,

$$F' := \{(\theta, x) \in \mathcal{P} \mid \theta + x(L_{uv}) + (d_u - c_{uv})x_u + (d_v - \min\{1, c_{uv}\})x_v = d_u + d_v - c_{uv}\},$$

where $d_u \equiv \deg_G(u)$. Consider an arbitrary proper face of $\mathcal{P}$ given by:

$$F := \{(\theta, x) \in \mathcal{P} \mid a_0\theta + \sum_{i \in V} a_i x_i = b\},$$

which we assume contains F'.

Consider the following point: $\theta = d_u + d_v - c_{uv}; x_i = 1 \ \forall i \notin N(u) \cup N(v); x_i = 0 \ \forall i \in N(u) \cup N(v)$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(d_u + d_v - c_{uv})a_0 + \sum_{i \notin N(u) \cup N(v)} a_i = b. \tag{33}$$

Now for some $\ell \in L_{uv}$, consider the following point: $\theta = d_u + d_v - c_{uv} - 1; x_i = 1 \ \forall i \notin N(u) \cup N(v); x_\ell = 1; x_i = 0 \ \forall i \in N(u) \cup N(v) \setminus \{l\}$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(d_u + d_v - c_{uv} - 1)a_0 + \sum_{i \notin N(u) \cup N(v)} a_i + a_\ell = b. \tag{34}$$

From equations (33) and (34), we can conclude that $a_\ell = a_0 \ \forall \ell \in L_{uv}$ and rewrite F as follows:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0 x(L_{uv}) + a_u x_u + a_v x_v + \sum_{i \notin N(u) \cup N(v)} a_i x_i = b\}.$$

Next we consider an arbitrary $\ell \notin N(u) \cup N(v)$. As $N(u) \cup N(v) \in \mathcal{C}^*$, we know that $N(u) \cup N(v) \cup \{\ell\}$ cannot be an s-club; otherwise, we will contradict the definition of $\mathcal{C}^*$ (see Proposition 2). Hence, we consider the following point: $\theta = d_u + d_v - c_{uv}; x_i = 0 \ \forall i \in N(u) \cup N(v) \cup \{\ell\}; x_i = 1 \ \forall i \notin N(u) \cup N(v) \cup \{\ell\}$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(d_u + d_v - c_{uv})a_0 + \sum_{i \notin \{N(u) \cup N(v)\} \cup \{\ell\}} a_i = b. \tag{35}$$

Now from equations (33) and (35), we can conclude that $a_\ell = 0$ for each $\ell \notin N(u) \cup N(v)$ and $b = (d_u + d_v - c_{uv})a_0$. We now know that F has the following form:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0 x(L_{uv}) + a_u x_u + a_v x_v = (d_u + d_v - c_{uv})a_0\}.$$

To identify the coefficients a_u and a_v, we consider the following two cases.

(i) Suppose $c_{uv} = 0$.

Consider the point $\theta = d_v; x_i = 0\ \forall i \in N(u) \cup N(v) \setminus \{u\}; x_i = 1\ \forall i \notin L_{uv}; x_u = 1$ As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$d_v a_0 + a_u = (d_u + d_v)a_0. \tag{36}$$

Hence, $a_u = d_u a_0$.

Now, consider the following point to determine coefficient a_v: $\theta = d_u; x_i = 0\ \forall i \in N(u) \cup N(v) \setminus \{v\}; x_i = 1\ \forall i \notin L_{uv}; x_v = 1$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$d_u a_0 + a_v = (d_u + d_v)a_0. \tag{37}$$

Hence, $a_v = d_v a_0$.

(ii) Suppose $c_{uv} \geq 1$.

Consider the point: $\theta = d_v; x_i = 0\ \forall i \in N(u) \cup N(v) \setminus \{u\}; x_i = 1\ \forall i \notin L_{uv}; x_u = 1$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$d_v a_0 + a_u = (d_u + d_v - c_{uv})a_0. \tag{38}$$

Hence, $a_u = (d_u - c_{uv})a_0$.

Finally, consider the following point to determine coefficient a_v: $\theta = 1; x_i = 0\ \forall i \in L_{uv}; x_i = 1\ \forall i \notin L_{uv}; x_u = x_v = 1$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$a_0 + (d_u - c_{uv})a_0 + a_v = (d_u + d_v - c_{uv})a_0. \tag{39}$$

Hence, $a_v = (d_v - 1)a_0$.

Combining the two cases together we obtain $a_u = (d_u - c_{uv})a_0$ and $a_v = (d_v - \min\{1, c_{uv}\})a_0$. Because F is a proper face, we know that $a_0 \neq 0$ and thus we conclude that $F' = F$ is a maximal proper face. □

A.12. Proof of Proposition 4

($\Longrightarrow$) The claim is trivial if H is empty; suppose not. Because S and $S \setminus H$ are s-clubs, it follows that $G[S]$ and $G[S \setminus H]$ are both connected. Hence, $S \setminus H$ dominates $G[S]$. It suffices to show that between distinct, non-adjacent vertices u and v in S, there exists a path of length at most s whose internal vertices belong to $S \setminus H$. The claim is trivially true if u and v belong to $S \setminus H$.

Suppose u and v belong to H. Define $T := H \setminus \{u, v\}$. Because S is an H-hereditary s-club, we know that $S \setminus T$ is an s-club that contains u and v. Hence, there exists a path of length at most s between u and v in $G[S \setminus T]$ and the internal vertices on this path clearly do not intersect H.

Now suppose $u \in S \setminus H$ and $v \in H$. Define $T' := H \setminus \{v\}$. As before, $S \setminus T'$ is an s-club that contains both u and v and the internal vertices on some path of length at most s between them in $G[S \setminus T]$ are all contained in $S \setminus H$.

($\Longleftarrow$) For an arbitrary $T \subseteq S \setminus D$, we need to show that $S \setminus T$ is an s-club. Consider distinct, non-adjacent vertices u and v in $S \setminus T$. By definition, there exists a u, v-path of length at most s in $G[S]$ with all its internal vertices contained in D. None of these vertices are deleted when T is deleted and the path is preserved in $G[S \setminus T]$. □

Appendix B: Supplementary Experimental Results

B.1. Comparison of Root Node Performance of Method 1 and Method 3

We have compared the performance of Method 1 and Method 3 in the root node by setting a termination condition on the number of explored nodes. All other solver parameters including primal heuristics and general purpose cutting planes are at their default settings. With this condition, the solver terminates when the total number of branch-and-cut nodes explored exceeds the value specified in the Gurobi `NodeLimit` parameter (which is 1 in our case). Tables 13 and 14 show the results of these experiments. Comparing the quality of the objective values and gaps obtained by each method in the root node shows that except for graph `football` for $s = 2$, Method 3 gives the same or a smaller gap and a smaller objective value than Method 1 at the root node. These results (along with the results presented in Appendix B.2) suggest that the improvements observed in Method 3 are predominantly because the heredity-based formulation is better than the standard formulation.

Table 13 **Root node comparison of Method 1 and Method 3 on Group-1 instances for $s = 2$ and $\alpha = 0.5$.**

Graph G	Method	#CB	#Cuts	Total time (s)	s-club time (s)	Obj Val	Gap (%)
karate	3	17	14	0.14	0.02	7.00	8.52
	1	84	82	0.26	0.12	16.50	92.42
dolphins	3	12	11	0.39	0.20	11.50	20.66
	1	45	44	1.78	1.66	12.00	83.33
lesmis	3	18	16	0.32	0.02	13.00	13.89
	1	47	43	0.15	0.07	21.50	93.02
polbooks	3	19	15	0.41	0.11	18.00	18.80
	1	51	49	1.20	1.11	24.50	91.84
adjnoun	3	12	9	0.90	0.59	16.00	20.11
	1	81	77	2.52	2.27	28.50	93.97
football	3	42	38	11.36	11.14	16.00	20.54
	1	34	29	10.12	10.03	15.50	77.88
celegansn	3	9	5	4.59	4.36	32.50	21.77
	1	40	39	4.82	4.07	135.00	98.12
celegansm	3	7	3	0.44	0.08	26.00	9.79
	1	36	35	0.66	0.51	238.00	98.74
email	3	5	1	24.35	23.65	44.50	9.76
	1	29	27	153.49	152.52	66.50	86.29
netscience	3	3	1	1.16	0.04	22.50	5.63
	1	73	71	2.14	1.27	29.50	49.37
add20	3	7	1	2.60	0.77	62.50	20.01
	1	98	97	93.01	18.97	124.00	86.88
data	3	3	1	12.59	12.19	17.50	4.62
	1	3	1	13.76	13.45	17.50	9.36
uk	3	8	7	27.04	26.85	5.00	0.00
	1	7	6	23.97	23.77	5.00	0.00
power	3	3	1	2.05	1.61	16.00	0.00
	1	36	34	27.39	26.16	16.00	5.52
add32	3	4	1	3.21	1.15	31.00	3.95
	1	54	53	23.96	13.78	32.00	33.55
hep-th	3	4	2	10.55	6.40	38.00	7.63
	1	163	162	465.74	267.26	51.00	48.35
whitaker3	3	2	1	44.94	44.68	9.00	0.00
	1	2	1	48.05	47.72	9.00	0.00
crack	3	3	1	69.49	69.19	9.50	0.00
	1	3	1	88.99	88.58	9.50	0.00
PGP	3	3	1	28.05	8.29	69.50	9.68
	1	138	137	933.31	636.85	206.00	84.89
cs4	3	7	6	716.82	713.64	6.00	0.00
	1	8	7	666.96	663.10	6.00	0.00

Table 14 **Root node comparison of Method 1 and Method 3 on Group-1 instances for $s = 3$ and $\alpha = 0.5$.**

Graph G	Method	#CB	#Cuts	Total time(s)	s-club time(s)	LCDS time(s)	Obj Val	Gap (%)
karate	3	2	0	0.24	0.00	0.00	7.50	5.89
	1	62	59	0.16	0.05		15.00	82.14
dolphins	3	33	30	1.07	0.41	0.24	17.50	34.99
	1	66	66	0.84	0.63		20.50	70.95
lesmis	3	4	0	0.6	0.00	0.00	13.50	7.81
	1	81	80	0.32	0.10		25.50	79.88
polbooks	3	35	31	2.88	0.79	0.32	22.50	22.63
	1	42	41	0.50	0.22		29.00	78.45
adjnoun	3	53	49	8.7	6.80	0.59	27.50	39.84
	1	51	49	4.01	3.31		31.00	76.20
celegansm	3	55	50	16.77	4.18	0.87	31.50	13.63
	1	32	31	8.80	4.23		184.50	94.69
netscience	3	8	6	13.22	0.11	0.05	27.00	8.49
	1	113	111	7.56	1.86		40.00	56.01
uk	3	9	8	46.01	44.95	0.04	8.00	0.00
	1	13	11	51.20	50.12		8.00	0.00
power	3	23	20	27.96	14.72	0.11	25.50	5.22
	1	211	210	193.99	164.46		51.50	58.11
whitaker3	3	9	8	274.45	264.57	0.04	15.00	0.00
	1	14	12	348.51	339.10		15.00	0.00
crack	3	10	9	383.09	372.56	0.04	17.00	0.00
	1	18	16	606.28	595.34		17.00	0.00
cs4	3	9	8	961.8	948.99	0.04	10.50	0.00
	1	14	14	1172.67	1158.99		10.50	0.00

B.2. Impact of Gurobi Heuristics on Method 3

As our DBC algorithm only separates integral solutions, it stands to reason that its performance will depend on the effectiveness of the primal heuristics built into the Gurobi solver that produce integral solutions to the master relaxation. In order to examine the dependency of our algorithm performance on Gurobi primal heuristics, we have performed experiments that disable these heuristics. Table 15 and 16 report the results for $s = 2$ and $s = 3$, respectively. When $s = 2$, we find that 16 out of 20 graphs are solved faster when turning off the primal heuristics (47% decrease on average), while the running times increase for other instances `adjnoun` (24%), `football` (122%), `celegansn` (55%) and `PGP` (16%). It can also be seen that in general, the number of explored nodes increases, and the number of callbacks and cuts decreases when Gurobi heuristics are turned off. When $s = 3$ the decrease in the running times is 39% on average for 9 out of 12 instances and for graphs `adjnoun`, `power` and `crack`, the running times increase 6%, 24%, and 4%, respectively. The number of explored nodes increases for 6 instances while the number of callbacks and cuts increase only for the graph `power`.

Based on these results, it is difficult to conclude that turning Gurobi heuristics on or off leads to a consistent, predictable impact on the overall performance. This may be attributed to the conflicting forces at play. Turning off Gurobi heuristics can result in fewer (or no) integral solutions encountered at the root node that invoke separation calls, with less time spent finding s-clubs and re-solving node relaxations as a result. In some (easier) instances, this can be beneficial as the wallclock time is reduced by simply letting the tree enumerate. However, on other instances, turning off Gurobi heuristics resulting in fewer integral solutions leading to fewer separation calls and fewer constraints generated at the root node, costs us in overall performance. A weaker relaxation at the root node and a larger tree size are a result of primal heuristic solutions not triggering constraint generation as often when turned off. However, the performance that is elicited by the choice seems to be very instance specific, and no doubt a function of the master relaxation strength and integrality for the instance under consideration.

Table 15 **Impact of Gurobi heuristics on Method 3 when solving Group-1 instances for $s = 2$ and $\alpha = 0.5$.**

Graph G	Heuristics	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)
karate	Off	8	3	80	17	15	0.07	0.01
	On	8	3	41	17	14	0.16	0.01
dolphins	Off	3	10	211	14	12	0.23	0.17
	On	3	10	200	16	15	0.52	0.25
lesmis	Off	10	8	335	18	16	0.06	0.01
	On	10	8	90	18	16	0.52	0.04
polbooks	Off	20	8	476	26	22	0.33	0.26
	On	16	10	268	23	19	0.41	0.13
adjnoun	Off	12	10	848	20	16	1.33	1.20
	On	12	10	331	16	13	1.07	0.67
football	Off	1	15	3,686,723	137	133	204.61	47.15
	On	1	15	973,384	96	92	92.14	26.56
celegansn	Off	23	21	3,290	18	14	9.57	9.18
	On	23	21	2,214	13	9	6.18	5.61
celegansm	Off	32	10	335	6	1	0.31	0.05
	On	32	10	181	7	3	0.48	0.09
email	Off	12	38	1,445	5	2	38.04	36.88
	On	12	38	1,020	6	1	39.51	37.44
netscience	Off	3	21	88	1	0	0.31	0.03
	On	3	21	206	3	1	1.20	0.04
add20	Off	52	34	4,701	8	1	8.98	0.83
	On	52	34	7,088	9	2	12.59	0.95
data	Off	1	17	28	1	0	5.10	4.84
	On	1	17	20	3	1	13.25	12.86
uk	Off	0	5	1	3	3	12.78	12.65
	On	0	5	1	8	7	29.07	28.87
power	Off	2	15	1	1	1	1.21	0.76
	On	2	15	1	3	1	2.01	1.59
add32	Off	4	29	56	1	0	0.90	0.26
	On	4	29	58	4	1	3.19	1.09
hep-th	Off	18	29	826	3	1	7.78	5.92
	On	18	29	412	4	2	10.95	6.34
whitaker3	Off	0	9	1	1	0	25.90	25.64
	On	0	9	1	2	1	47.00	46.72
crack	Off	1	9	1	1	0	32.36	32.08
	On	1	9	1	3	1	72.81	72.51
PGP	Off	45	47	4,613	4	0	48.14	17.10
	On	45	47	5,858	3	1	41.63	8.20
cs4	Off	0	6	1	4	3	363.48	361.44
	On	0	6	1	7	6	503.88	501.18

Table 16 Impact of Gurobi heuristics on Method 3 when solving Group-1 instances for $s = 3$ and $\alpha = 0.5$.

Graph G	Heuristics	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)
karate	Off	11	2	134	1	0	0.14	0.00	0.00
	On	7	4	78	2	0	0.24	0.00	0.00
dolphins	Off	20	4	7,344	21	19	1.86	0.27	0.08
	On	18	5	10,654	52	39	3.79	0.45	0.16
lesmis	Off	13	7	148	1	0	0.37	0.00	0.00
	On	13	7	85	4	0	0.59	0.00	0.00
polbooks	Off	27	8	47,587	21	18	16.84	0.10	0.07
	On	27	8	42,088	44	34	24.74	0.83	0.20
adjnoun	Off	26	8	181,072	29	27	124.49	3.15	0.11
	On	22	10	148,295	86	75	117.51	9.64	0.45
celegansm	Off	35	14	154,883	35	34	348.04	0.80	0.17
	On	35	14	279,199	60	55	717.01	4.37	0.81
netscience	Off	14	20	8,841	3	3	7.34	0.07	0.01
	On	12	21	7,867	8	6	30.69	0.12	0.04
uk	Off	2	7	1	6	6	44.45	42.54	0.03
	On	0	8	1	9	8	51.10	50.15	0.03
power	Off	7	22	25,879	31	29	141.99	30.96	0.16
	On	7	22	20,025	23	20	114.11	16.14	0.08
whitaker3	Off	0	15	1	6	6	238.44	221.16	0.03
	On	0	15	1	9	8	294.46	284.14	0.02
crack	Off	0	17	1	8	7	420.60	407.57	0.05
	On	0	17	1	10	9	405.37	394.57	0.03
cs4	Off	1	10	1	6	6	755.52	741.10	0.03
	On	1	10	1	9	8	1005.55	994.03	0.03

B.3. Impact of Exact and Inexact Separation on Method 3

We performed experiments to evaluate the impact of inexact separation on the overall performance of Method 3. Given an integral feasible solution $(\hat{\theta}, \hat{x})$ to the master relaxation, instead of finding a maximum s-club in the graph interdicted according to $\hat{x}$, we look for an s-club with cardinality at least $\hat{\theta} + \epsilon$ using the procedure described in Section 6.3, where ϵ is the minimum violation we seek in the constraint. We experimented with $\epsilon = 1.5$, 2.5, and 5.

Results of these experiments and their comparison with the default setting are shown in Tables 17–20. The last two columns of these tables, #ICUT-H and #ICUT-ϵ respectively indicate the number of lazy cuts detected using the first and second early termination attempts. Tables 17–19 report the results for $\epsilon = 5$, 2.5, and 1.5 when $s = 2$. As it can be seen, $\epsilon = 5$ is too large of a target for early termination and the separation problem is solved to optimality in most of the iterations of our test bed. As the value of ϵ decreases, more ϵ-violated cuts are found early.

Tables 18 and 19 show that for both $\epsilon = 2.5$ and $\epsilon = 1.5$, the decrease in the running times is about 25% on average for 18 out of 20 instances. For the other 2 instances, the running times increase 4% and 6% on average for $\epsilon = 2.5$ and $\epsilon = 1.5$, respectively. Considering only those instances in Table 19 ($\epsilon = 1.5$) that take at least one minute to be solved, which are `football, crack` and `cs4`, we can observe that the running times of `football` and `crack` decrease 68% and 10% respectively, and the running time of `cs4` increases 10% when inexact separation is used. For all the other instances that take less than a minute to solve, the running times decrease at an average of 24%.

Regarding the number of cuts in Table 19 ($\epsilon = 1.5$), for the graph `football`, 48 out of 55 total cuts are found by early termination. The total number of callbacks and cuts decreased from 96 and 92, respectively, to 60 and 55, which suggests that the cuts are sufficiently strong. For the graph `celegansn`, although all the violated constraints are found by heuristics, the total number of callbacks and cuts increased, which means that the cuts added using heuristics are weaker when compared to the cuts added by the exact solution in this instance. In other instances, the number of callbacks and total number of cuts are the same for both exact and inexact separation or the difference is negligible.

The results of the experiments for the 3-club interdiction problem showed similar behavior, thus we only report the results for $\epsilon = 1.5$, the case where more violated constraints are found by inexact separation. In Table 20, out of 12 instances, the running times decrease 31% on average for six instances and increase about 7% on average for three instances. Comparing the decrease in the running times for all the instances (25% when $s = 2$ and 31% when $s = 3$ for $\epsilon = 1.5$) with the decrease in the more challenging instances that take at least a minute to solve (39% when $s = 2$ and 41% when $s = 3$ for $\epsilon = 1.5$) shows that using inexact separation is more helpful for solving the more challenging instances.

Table 17 Inexact versus exact separation on Group-1 instances with $s=2$, $\alpha=0.5$, and $\epsilon=5$.

Graph G	Method	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	#ICUT-H	#ICUT-ϵ
karate	inexact	8	3	41	17	14	0.15	0.02	0	0
	exact	8	3	41	17	14	0.16	0.01		
dolphins	inexact	3	10	200	16	15	0.38	0.18	3	1
	exact	3	10	200	16	15	0.52	0.25		
lesmis	inexact	10	8	90	18	16	0.31	0.01	3	0
	exact	10	8	90	18	16	0.52	0.04		
polbooks	inexact	16	10	268	23	19	0.46	0.11	6	0
	exact	16	10	268	23	19	0.41	0.13		
adjnoun	inexact	12	10	528	15	13	0.89	0.52	2	2
	exact	12	10	331	16	13	1.07	0.67		
football	inexact	1	15	356,392	56	51	28.40	10.89	19	0
	exact	1	15	973,384	96	92	92.14	26.56		
celegansn	inexact	23	21	1,528	24	20	2.80	2.37	20	0
	exact	23	21	2,214	13	9	6.18	5.61		
celegansm	inexact	30	11	181	7	3	0.45	0.06	1	0
	exact	32	10	181	7	3	0.48	0.09		
email	inexact	12	38	1,020	6	1	32.54	30.56	1	0
	exact	12	38	1,020	6	1	39.51	37.44		
netscience	inexact	3	21	206	3	1	1.34	0.03	1	0
	exact	3	21	206	3	1	1.20	0.04		
add20	inexact	52	34	7,088	9	2	12.51	0.87	0	0
	exact	52	34	7,088	9	2	12.59	0.95		
data	inexact	1	17	20	3	1	10.02	9.64	1	0
	exact	1	17	20	3	1	13.25	12.86		
uk	inexact	0	5	1	8	7	26.94	26.71	0	0
	exact	0	5	1	8	7	29.07	28.87		
power	inexact	2	15	1	3	1	1.86	1.40	0	0
	exact	2	15	1	3	1	2.01	1.59		
add32	inexact	4	29	58	4	1	3.02	0.96	0	0
	exact	4	29	58	4	1	3.19	1.09		
hep-th	inexact	18	29	412	4	2	10.66	5.95	1	0
	exact	18	29	412	4	2	10.95	6.34		
whitaker3	inexact	0	9	1	2	1	27.56	27.26	1	0
	exact	0	9	1	2	1	47.00	46.72		
crack	inexact	1	9	1	3	1	64.39	63.97	1	0
	exact	1	9	1	3	1	72.81	72.51		
PGP	inexact	45	47	5,858	3	1	40.49	7.42	1	0
	exact	45	47	5,858	3	1	41.63	8.20		
cs4	inexact	0	6	1	8	6	521.39	517.40	0	3
	exact	0	6	1	7	6	503.88	501.18		

Table 18 Inexact versus exact separation on Group-1 instances with $s=2$, $\alpha=0.5$, and $\epsilon=2.5$.

Graph G	Method	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	#ICUT-H	#ICUT-ϵ
karate	inexact	8	3	41	17	14	0.10	0.01	8	0
	exact	8	3	41	17	14	0.16	0.01		
dolphins	inexact	3	10	218	16	14	0.33	0.16	9	3
	exact	3	10	200	16	15	0.52	0.25		
lesmis	inexact	10	8	90	18	16	0.27	0.01	10	0
	exact	10	8	90	18	16	0.52	0.04		
polbooks	inexact	16	10	268	23	19	0.39	0.10	16	0
	exact	16	10	268	23	19	0.41	0.13		
adjnoun	inexact	12	10	394	15	13	0.68	0.33	10	1
	exact	12	10	331	16	13	1.07	0.67		
football	inexact	1	15	510,258	65	60	38.98	12.73	22	8
	exact	1	15	973,384	96	92	92.14	26.56		
celegansn	inexact	23	21	1,528	24	20	2.61	2.21	20	0
	exact	23	21	2,214	13	9	6.18	5.61		
celegansm	inexact	30	11	181	7	3	0.42	0.06	1	0
	exact	32	10	181	7	3	0.48	0.09		
email	inexact	12	38	1,020	6	1	31.47	29.42	1	0
	exact	12	38	1,020	6	1	39.51	37.44		
netscience	inexact	3	21	206	3	1	1.21	0.03	1	0
	exact	3	21	206	3	1	1.20	0.04		
add20	inexact	52	34	7,088	9	2	12.42	0.85	1	0
	exact	52	34	7,088	9	2	12.59	0.95		
data	inexact	1	17	20	3	1	9.67	9.28	1	0
	exact	1	17	20	3	1	13.25	12.86		
uk	inexact	0	5	1	8	7	20.94	20.71	3	0
	exact	0	5	1	8	7	29.07	28.87		
power	inexact	2	15	1	3	1	1.83	1.35	1	0
	exact	2	15	1	3	1	2.01	1.59		
add32	inexact	4	29	58	4	1	3.11	0.95	1	0
	exact	4	29	58	4	1	3.19	1.09		
hep-th	inexact	18	29	412	4	2	10.64	5.93	1	0
	exact	18	29	412	4	2	10.95	6.34		
whitaker3	inexact	0	9	1	2	1	27.21	26.90	1	0
	exact	0	9	1	2	1	47.00	46.72		
crack	inexact	1	9	1	3	1	65.84	65.38	1	0
	exact	1	9	1	3	1	72.81	72.51		
PGP	inexact	45	47	5,858	3	1	40.66	7.44	1	0
	exact	45	47	5,858	3	1	41.63	8.20		
cs4	inexact	0	6	1	9	7	540.55	536.14	3	0
	exact	0	6	1	7	6	503.88	501.18		

Table 19 Inexact versus exact separation on Group-1 instances with $s = 2$, $\alpha = 0.5$, and $\epsilon = 1.5$.

Graph G	Method	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	#ICUT-H	#ICUT-ϵ
karate	inexact	8	3	41	17	14	0.09	0.01	9	0
	exact	8	3	41	17	14	0.16	0.01		
dolphins	inexact	3	10	215	16	14	0.39	0.19	12	0
	exact	3	10	200	16	15	0.52	0.25		
lesmis	inexact	10	8	90	18	16	0.28	0.01	11	0
	exact	10	8	90	18	16	0.52	0.04		
polbooks	inexact	16	10	268	23	19	0.32	0.03	18	0
	exact	16	10	268	23	19	0.41	0.13		
adjnoun	inexact	12	10	433	18	16	0.66	0.33	14	0
	exact	12	10	331	16	13	1.07	0.67		
football	inexact	1	15	503,717	60	55	29.72	5.86	23	25
	exact	1	15	973,384	96	92	92.14	26.56		
celegansn	inexact	23	21	1,528	24	20	2.60	2.19	20	0
	exact	23	21	2,214	13	9	6.18	5.61		
celegansm	inexact	30	11	181	7	3	0.41	0.06	1	0
	exact	32	10	181	7	3	0.48	0.09		
email	inexact	12	38	1,020	6	1	31.40	29.49	1	0
	exact	12	38	1,020	6	1	39.51	37.44		
netscience	inexact	3	21	206	3	1	1.21	0.03	1	0
	exact	3	21	206	3	1	1.20	0.04		
add20	inexact	52	34	7,088	9	2	12.40	0.88	1	0
	exact	52	34	7,088	9	2	12.59	0.95		
data	inexact	1	17	20	3	1	9.71	9.34	1	0
	exact	1	17	20	3	1	13.25	12.86		
uk	inexact	0	5	1	8	7	21.04	20.82	3	0
	exact	0	5	1	8	7	29.07	28.87		
power	inexact	2	15	1	3	1	1.80	1.35	1	0
	exact	2	15	1	3	1	2.01	1.59		
add32	inexact	4	29	58	4	1	3.10	0.94	1	0
	exact	4	29	58	4	1	3.19	1.09		
hep-th	inexact	18	29	412	4	2	10.86	6.01	1	0
	exact	18	29	412	4	2	10.95	6.34		
whitaker3	inexact	0	9	1	2	1	27.85	27.56	1	0
	exact	0	9	1	2	1	47.00	46.72		
crack	inexact	1	9	1	3	1	65.88	65.50	1	0
	exact	1	9	1	3	1	72.81	72.51		
PGP	inexact	45	47	5,858	3	1	40.18	7.43	1	0
	exact	45	47	5,858	3	1	41.63	8.20		
cs4	inexact	0	6	1	9	7	555.40	550.93	3	0
	exact	0	6	1	7	6	503.88	501.18		

Table 20 Inexact versus exact separation on Group-1 instances with $s = 3$, $\alpha = 0.5$, and $\epsilon = 1.5$.

Graph G	Method	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)	#ICUT-H	#ICUT-ϵ
karate	inexact	7	4	78	2	0	0.21	0.00	0.00	0	0
	exact	7	4	78	2	0	0.24	0.00	0.00		
dolphins	inexact	18	5	9,137	66	56	3.46	0.24	0.28	30	3
	exact	18	5	10,654	52	39	3.79	0.45	0.16		
lesmis	inexact	13	7	85	4	0	0.54	0.00	0.00	0	0
	exact	13	7	85	4	0	0.59	0.00	0.00		
polbooks	inexact	27	8	70,436	56	49	29.27	0.50	0.37	22	3
	exact	27	8	42,088	44	34	24.74	0.83	0.20		
adjnoun	inexact	24	9	189,962	119	107	119.30	5.07	0.66	35	33
	exact	22	10	148,295	86	75	117.51	9.64	0.45		
celegansm	inexact	35	14	98,595	39	37	353.80	1.62	1.06	9	18
	exact	35	14	279,199	60	55	717.01	4.37	0.81		
netscience	inexact	12	21	7,867	8	6	30.70	0.07	0.06	4	0
	exact	12	21	7,867	8	6	30.69	0.12	0.04		
uk	inexact	0	8	1	9	8	36.30	35.24	0.04	0	0
	exact	0	8	1	9	8	51.10	50.15	0.03		
power	inexact	7	22	17,530	24	22	105.00	16.35	0.12	8	3
	exact	7	22	20,025	23	20	114.11	16.14	0.08		
whitaker3	inexact	0	15	1	9	8	142.25	131.67	0.04	1	5
	exact	0	15	1	9	8	294.46	284.14	0.02		
crack	inexact	0	17	1	10	9	192.31	176.81	0.05	0	6
	exact	0	17	1	10	9	405.37	394.57	0.03		
cs4	inexact	1	10	1	9	8	581.47	568.57	0.04	0	6
	exact	1	10	1	9	8	1005.55	994.03	0.03		