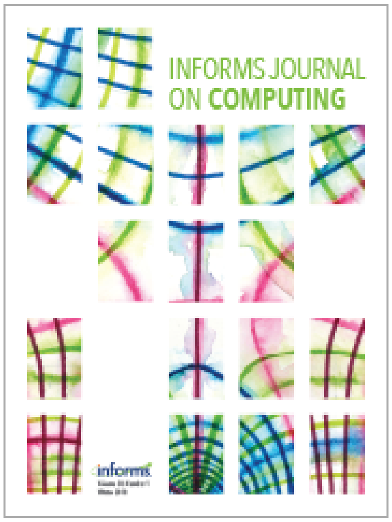


INFORMS Journal on Computing

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

A Branch-and-Cut Algorithm for Submodular Interdiction Games

Kübra Tanınmış, Markus Sinnl

INFORMS JOURNAL ON COMPUTING
Vol. 34, No. 5, September–October 2022, pp. 2634–2657
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

A Branch-and-Cut Algorithm for Submodular Interdiction Games

Kübra Tanınmış,[a] Markus Sinnl [a,*]

[a] Institute of Production and Logistics Management/JKU Business School, Johannes Kepler University, Linz, Austria
*Corresponding author

Contact: kuebra.taninmis_ersues@jku.at, https://orcid.org/0000-0003-1081-4182 (KT); markus.sinnl@jku.at, https://orcid.org/0000-0003-1439-8702 (MS)

Received: March 29, 2021
Revised: October 8, 2021; March 15, 2022; March 27, 2022
Accepted: March 30, 2022
Published Online in Articles in Advance: May 12, 2022

https://doi.org/10.1287/ijoc.2022.1196

Abstract. Many relevant applications from diverse areas such as marketing, wildlife conservation, and defending critical infrastructure can be modeled as interdiction games. In this work, we introduce interdiction games whose objective is a monotone and submodular set function. Given a ground set of items, the leader interdicts the usage of some of the items of the follower in order to minimize the objective value achievable by the follower, who seeks to maximize a submodular set function over the uninterdicted items subject to knapsack constraints. We propose an exact branch-and-cut algorithm for this kind of interdiction game. The algorithm is based on interdiction cuts, which allow the leader to capture the follower's objective function value for a given interdiction decision of the leader and exploit the submodularity of the objective function. We also present extensions and liftings of these cuts and discuss additional preprocessing procedures. We test our solution framework on the weighted maximal covering interdiction game and the bipartite inference interdiction game. For both applications, the improved variants of our interdiction cut perform significantly better than the basic version. For the weighted maximal covering interdiction game for which a mixed-integer bilevel linear programming (MIBLP) formulation is available, we compare the results with those of a state-of-the-art MIBLP solver. Whereas the MIBLP solver yields a minimum of 54% optimality gap within one hour, our best branch-and-cut setting solves all but four of 108 instances to optimality with a maximum of 3% gap among unsolved ones.

History: Accepted by Andrea Lodi, Area Editor for Design & Analysis of Algorithms—Discrete.
Funding: This research was funded in whole, or in part, by the Austrian Science Fund [Grant P 35160-N]. This work was also supported by the Johannes Kepler University Linz, Linz Institute of Technology (LIT) [Grant LIT-2019-7-YOU-211]. LIT is funded by the state of Upper Austria.
Supplemental Material: The online supplement is available at https://doi.org/10.1287/ijoc.2022.1196.

Keywords: interdiction games • submodular optimization • bilevel optimization • branch-and-cut

1. Introduction and Problem Definition

A bilevel optimization problem involves two decision makers with conflicting objectives. First, the *leader* makes a decision and then the *follower* reacts by solving an optimization problem subject to the leader's decision. The players have perfect information of each other's problems and the leader anticipates the follower's optimal response. In other words, they play a sequential game known as a Stackelberg game (Von Stackelberg 1952). Although many real-world problems involving competition and noncooperation can be addressed as bilevel optimization models, even the simplest version of bilevel problems is known to be NP-hard (Jeroslow 1985, Ben-Ayed and Blair 1990). A recent survey on bilevel optimization is presented by Dempe and Zemkoho (2020).

In this paper, we address a special class of the bilevel optimization problems called *interdiction games* (IGs). These problems are two-player zero-sum Stackelberg games and have received considerable attention in recent years. In an IG, the aim of the leader is to attain the maximum deterioration in the follower's optimal objective value by interdicting the follower's decisions. IGs have applications in diverse areas such as marketing (DeNegre 2011), wildlife conservation (Mc Carthy et al. 2016, Sefair et al. 2017), and defending critical infrastructure (Brown et al. 2006). Most IGs that have been studied so far are related to network interdiction, where certain components of a network, such as its edges or nodes, are interdicted by the leader so that the follower cannot use them to achieve its objective. Smith and Song (2020) present a comprehensive survey on network interdiction models. Other popular IGs are the knapsack interdiction problem (DeNegre 2011), and the facility interdiction problem and its variants (Church et al. 2004, Aksen et al. 2014). A more detailed review of IGs and state-of-the-art solution approaches is provided in Section 1.3.

1.1. Problem Definition

In this study, we consider the class of IGs with a submodular and monotone (nondecreasing) objective function. Given a finite ground set N (of items), a function $z : 2^N \rightarrow \mathbb{R}$ is called submodular if $z(S\{i\}) - z(S) \geq z(T \cup \{i\}) - z(T)$, for all $S \subseteq T \subseteq N$ and $i \in N \setminus T$ (alternative definitions by Nemhauser et al. (1978) are provided in Section 2.1). The function z is also monotone if $z(S) \leq z(T)$ for all $S \subseteq T \subseteq N$.

Maximizing a monotone submodular function subject to cardinality/budget constraints is NP-hard (Cornuejols et al. 1977) and it has many applications, including the maximal covering problem (Church and ReVelle 1974, Vohra and Hall 1993), uncapacitated facility location problem (Nemhauser et al. 1978), influence maximization problem under linear threshold and independent cascade models (Kempe et al. 2003), bipartite inference problem (Sakaue and Ishihata 2018, Salvagnin 2019), assortment optimization problem (Kunnumkal and Martínez-de Albéniz 2019), maximum capture location problem (Ljubić and Moreno 2018), feature selection in machine learning models (Krause et al. 2008), sensor placement problem (Guestrin et al. 2005), and generalized assignment problem (Calinescu et al. 2007). Krause and Golovin (2014) present a survey on submodular maximization and its applications.

In particular, we address IGs whose follower seeks to maximize a submodular and monotone set function subject to knapsack constraints. The leader of the game interdicts the usage of a set of items in N in order to minimize the follower's optimal objective value. An interdicted item is not available to the follower at any cost, that is, we assume complete interdiction. The problem addressed is formulated as

$$\min_{x \in X} \max\{z(S) : S \subseteq N \setminus N_x, C(S) \leq \boldsymbol{q}\}, \qquad (1)$$

where $N_x = \{i \in N : x_i = 1\}$ is the set of items that are not available to the follower under the interdiction strategy $\boldsymbol{x}$. The set $X = \{\boldsymbol{x} \in \{0,1\}^n : G\boldsymbol{x} \leq \boldsymbol{b}\}$ is the feasible region of the leader, where G and b are a real valued matrix and a vector of appropriate dimensions and n denotes the number of leader variables. The follower is constrained by L knapsack constraints $C(S) \leq \boldsymbol{q}$, where $C(S) = (c_1(S), \ldots, c_L(S))$ and $c_\ell(S) = \sum_{i \in S} c_i^\ell$ with $c_i^\ell \geq 0\ \forall i \in N, \ell = 1, \ldots, L$, and $\boldsymbol{q}$ is a vector of size L. Without loss of generality, we assume that z is normalized, that is, $z(\emptyset) = 0$, which can be done easily by setting $z(S) \leftarrow z(S) - z(\emptyset)$ (Du et al. 2012).

We note that all the problems mentioned in this section fall under the structure of the inner part of Problem (1) and interdiction versions of these problems can be solved with our solution approach. From the follower's perspective, the optimal solution of this problem yields the smallest achievable optimal objective value in the presence of an optimal adversary that can prevent the usage of some items within a budget, that is, a worst-case scenario (Smith 2010).

1.2. Contribution and Outline

The main contribution of this study is an exact method for solving IGs with a submodular and monotone objective function as given in (1). Using properties of submodular functions we introduce *submodular interdiction cuts* (SICs). They are based on the value of the contribution to the objective value due to adding an item to a given subset of N, which is called *marginal gain*. Problem (1) is reformulated as a single-level problem using our SICs and solved within a branch-and-cut scheme. We also propose various ways to lift our SICs and test the effectiveness of the resulting solution algorithms on the weighted maximal covering interdiction game and the bipartite inference interdiction game.

The outline of the paper is as follows. In the remainder of this section, we give a discussion of previous and related work. In Section 2, we recall basic properties of submodular functions and introduce the problems used in our computational study, namely the weighted maximal covering interdiction game and the bipartite inference interdiction game. In Section 3, we first show how to obtain a single-level reformulation of Problem (1) using interdiction cuts. Next, we introduce our basic SICs and propose improved, lifted, and alternative versions of our SICs. We also give illustrational examples of their occurrence in various interdiction games. Section 4 contains implementation details of our branch-and-cut solution framework, including separation procedures for our SICs. In Section 5, we present the computational results of our approach on the problem families selected as test bed. For the weighted maximal covering interdiction game, for which a mixed-integer bilevel linear programming (MIBLP) formulation is possible, we compare our approach against a state-of-the-art MIBLP solver. We conclude the paper with possible future research directions in Section 6.

1.3. Previous and Related Work

In some cases, IGs can be formulated as MIBLPs, in which case they are solvable via general-purpose MIBLP solvers such as the ones proposed by Xu and Wang (2014), Lozano and Smith (2017b), Fischetti et al. (2017), Yue et al. (2019), and Tahernejad et al. (2020). On the other hand, there exist also specialized methods either for a specific problem type or for more general IGs. In various studies, the IG addressed has a linear follower problem and is formulated as a single-level optimization problem via linear programming duality. This is the case in the works of Wollmer (1964), Wood (1993), and Morton et al. (2007), where the maximum flow interdiction problem is addressed with the aim of analyzing the sensitivity of a transportation network, reducing the flow of drugs on a network, and stopping nuclear smuggling, respectively. Similarly, the shortest path interdiction problem (Golden 1978, Israeli and Wood 2002, Bayrak and Bailey 2008) and the node deletion problem (Shen et al. 2012), which aims to damage the connectivity of a network, can be solved via duality-based approaches. Whereas this approach has been frequently used in IG modeling, many real-world problems give rise to mixed-integer lower-level problems. Among them, there are problems that can still be formulated as an MIP due to their special structure such as the r-median interdiction problem (Church et al. 2004). Using the closest assignment constraints, the follower decision can be integrated to the leader's problem. Some variants like the one with partial interdiction addressed in Aksen et al. (2014) still require MIBLP formulations.

The r-interdiction covering problem introduced in Church et al. (2004) involves finding the facilities to interdict to maximize the coverage reduction. It has applications in determining critical existing emergency facilities such as fire stations or emergency communication systems. Because it involves a single decision maker, the attacker, the problem is not exactly an IG and can be formulated as an MIP. The IG version of this problem with a defender locating facilities after interdiction fulfills the requirements of our framework and is one of the applications we consider in our computational study (see Section 2.2). Facility location interdiction problems have also been considered within a fortification setting called *defender-attacker-defender model*, where the defender seeks to minimize the damage due to interdiction (see, e.g., Brown et al. 2006; Scaparra and Church 2008a,b; Aksen et al. 2010 for the r-interdiction median with fortification; and Dong et al. 2010, Roboredo et al. 2019 for r-interdiction covering with fortification). Cappanera and Scaparra (2011) study shortest path interdiction with fortification. Lozano and Smith (2017a) propose a sampling-based exact method for a more general class of trilevel fortification problems.

The knapsack interdiction problem is one of the widely studied IGs. In one version of this problem, the leader's decision affects the follower's budget. Brotcorne et al. (2013) propose a dynamic programming-based method and a single-level formulation for this version. In a more commonly studied version introduced by DeNegre (2011), the leader interdicts the usage of some items by the follower, which could have an application in corporate marketing strategies. DeNegre (2011) develops a branch-and-cut scheme and Caprara et al. (2016) propose an iterative algorithm for this variant of the knapsack problem. Della Croce and Scatamacchia (2020) compute effective lower bounds on the optimal objective and use them to design an exact algorithm. Another interdiction problem, which recently got more attention in the literature, is the clique interdiction problem. The problem involves minimizing the size of the maximum clique in a network, by interdicting, that is, removing, a subset of its edges (Tang et al. 2016, Furini et al. 2021) or vertices (Furini et al. 2019).

Interdiction games have also been used in stochastic and robust optimization. For example, in Cormican et al. (1998), a stochastic network interdiction problem is considered. In Borrero and Lozano (2021), an attacker affects the objective function of the defender in an uncertain way. Two exact methods are proposed to solve the robust optimization problem of the defender who wants to be prepared for the worst-case scenario.

Regarding generic methods to solve IGs, Tang et al. (2016) propose iterative algorithms for IGs with a mixed-integer follower problem. These algorithms are finitely convergent when the leader variables are restricted to take binary values. Tanınmış et al. (2021) improve the algorithm of Tang et al. (2016) for the binary bilevel problem case, using a covering based reformulation of the problem instead of a duality-based one. Fischetti et al. (2019) address IGs that satisfy an assumption called *downward monotonicity*. They introduce a branch-and-cut approach based on efficient use of interdiction cuts, which previously have been used within problem-specific solution frameworks in several studies, including Israeli and Wood (2002), Cochran et al. (2011), and Caprara et al. (2016). Contardo and Sefair (2021) propose another iterative algorithm to solve binary linear IGs. Their approach is more general in the sense that it can handle both complete interdiction and partial interdiction, where interdicted items are available to the follower at higher cost.

The problems we address and design solution approaches for in this work form a more general class of IGs addressed in Fischetti et al. (2019). The reason is that we allow the objective function of the IG to be linear or nonlinear as long as it is submodular and monotone, which makes solving many additional problem classes possible. In addition to the bipartite inference interdiction game, which is described in Section 2.2 and addressed in our computational study, the assortment optimization problem (McFadden and Train 2000, Bonnet and Simioni 2001), the p-choice facility location problem (Tawarmalani et al. 2002), and more generally speaking multiple-ratio fractional 0-1 programs under mild assumptions (Han et al. 2022) are examples for problems with a nonlinear, submodular, and monotone objective function. On the other hand, a linear objective function of discrete decision variables can equivalently be expressed as a submodular monotone set function given that the objective coefficients are nonnegative.

Several works focus on the robust version of a constrained submodular maximization problem, in which the possibility of removal of some elements from the chosen set is taken into account, such as Orlin et al. (2016), Kazemi et al. (2018), and Bogunovic et al. (2017), who propose approximation algorithms. To the best of our knowledge, this is the first study considering a submodular maximization problem in the lower level of a bilevel setting.

2. Preliminaries

2.1. Submodular Functions

Given a submodular function z, let $\rho_i(S) = z(S \cup \{i\}) - z(S)$ be the marginal gain due to adding $i \in N$ to set $S \subseteq N$. The marginal gain $\rho(\cdot)$ is nonincreasing by definition of a submodular function. The following proposition gives alternative definitions for submodular functions.

Proposition 1 (Nemhauser et al. 1978). *If z is a submodular function, then*

$$z(T) \leq z(S) + \sum_{i \in T \setminus S} \rho_i(S) - \sum_{i \in S \setminus T} \rho_i(S \cup T \setminus \{i\}) \quad S, T \subseteq N, \tag{2}$$

$$z(T) \leq z(S) + \sum_{i \in T \setminus S} \rho_i(S \cap T) - \sum_{i \in S \setminus T} \rho_i(S \setminus \{i\}) \quad S, T \subseteq N. \tag{3}$$

Proposition 2 (Nemhauser et al. 1978). *If z is a submodular and monotone function, then*

$$\rho_i(S) \geq \rho_i(T) \geq 0 \qquad S \subseteq T \subseteq N, i \in N, \tag{4}$$

and the last term in (2) can be removed to obtain the simpler inequality:

$$z(T) \leq z(S) + \sum_{i \in T \setminus S} \rho_i(S) \qquad S, T \subseteq N. \tag{5}$$

2.2. Examples of Submodular Interdiction Games

The following two problems are interdiction variants of submodular optimization problems, which will be used in our computational study. The weighted maximal covering problem (MCP) is a classical problem in location science (see, e.g., Church and ReVelle 1974, Laporte et al. 2015), where the goal is to open q facilities in order to maximize the number of customers covered by these open facilities. In the proposed interdiction variant, which we denote as *weighted maximal coverage interdiction game* (WMCIG) the leader can interdict the opening of some facilities. Similar to interdiction variants of other facility location problems (see, e.g., Section 1.3) applications of the WMCIG are in critical infrastructure protection and facility location under competition. A formal definition of the problem is given next.

Definition 1 (Weighted Maximal Coverage Interdiction Game (WMCIG)). We are given a set of m customers J with profits p_j, $j \in J$, a set of potential facility locations N and for each facility $i \in N$ the set $J(i) \subseteq J$ of customers that facility i covers. Moreover, we are given two integers b and q. The problem of the follower is finding a set of q facilities to open to maximize the profit of covered customers, where the profit of a set $S \subseteq N$ of open facilities is defined as $z(S) = \sum_{j \in J(S)} p_j$, where $J(S) = {}_{i \in S} J(i)$, that is, the profit obtained from customers that are covered by at least one of the facilities in S. The goal of the leader is to interdict b facility locations, that is, to prevent the follower from choosing them, such that the profit of the follower is minimized.

We note that for the MCP a compact mixed-integer programming formulation is known, and thus for the WMCIG a MIBLP formulation can be obtained and the problem can be solved using a MIBLP solver. This formulation is discussed in Section 5, where we also provide a computational comparison between our branch-and-cut and solving WMCIG as a MIBLP with a state-of-the-art MIBLP solver.

The second problem we consider is the interdiction variant of the bipartite inference problem (BIP). In the BIP, we are given the sets of items and targets, and the probabilities of each item of activating each target. The objective is to maximize the total activation probability of the targets via selecting a subset of the items. The BIP is studied in Alon et al. (2012), Sakaue and Ishihata (2018), and Salvagnin (2019), with an application to the allocation of marketing budget among media channels in the former. Its interdiction version could represent a competitive setting where an existing firm tries to undermine the marketing activities of a newcomer by making contracts with some of the available channels so that the newcomer cannot make use of them. From the newcomer's point of view, solving the interdiction version of BIP gives the minimum achievable activation probability of targets in case the existing firm has exclusive agreements with some of the channels within the assumed budget. Contrary to the MCP, for the BIP only a submodular formulation is known.

Definition 2 (Bipartite Inference Interdiction Game (BIIG)). Given a set of items N, a set of targets J, and a bipartite graph $G = (N \cup J, A)$ where each arc $(i, j) \in A$ $(i \in N,\ j \in J)$ is labeled with an activating probability p_i, the objective of the follower in the BIIG is to select a set of items $S \subseteq N$ that maximizes the total activation probabilities of all targets

$$z(S) = \sum_{j \in J} p_S(j),$$

where

$$p_S(j) = 1 - \prod_{i \in S:(i,j)\in A} (1 - p_i)$$

denotes the activation probability of target j under the item set choice S. The given activating probability p_i of item $i \in N$ is independent of the target (Sakaue and Ishihata 2018, Salvagnin 2019). The follower can choose at most q items. The objective of the leader is to minimize the total activation probability by interdicting a set of items in N subject to a cardinality constraint in which at most b items can be interdicted. Interdicted items cannot be selected by the follower.

3. Solution Approach

In this section, we first reformulate an IG as a single-level problem using its value function. Then we introduce the basic SIC, which is followed by the improved, lifted, and alternative SICs. Our solution method is based on solving the reformulation with a branch-and-cut scheme, where our various SICs are separated for integer and fractional leader solutions. Implementation details are discussed in Section 4.

3.1. Single-Level Reformulation of IGs

Let $\Phi(\boldsymbol{x})$ be the value function of the follower problem of (1), that is, $\Phi(\boldsymbol{x}) = \max\{z(S) : S \subseteq N \setminus N_{\boldsymbol{x}}, C(S) \le q\}$. Our problem can be reformulated as

$$\min w, \tag{6}$$
$$w \ge \Phi(\boldsymbol{x}), \tag{7}$$
$$G\boldsymbol{x} \le \boldsymbol{b}, \tag{8}$$
$$\boldsymbol{x} \in \{0,1\}^n. \tag{9}$$

Rewriting the value function for given $\boldsymbol{x}$ as $\Phi(\boldsymbol{x}) = \max\{z(S) - \sum_{i\in S} M_i x_i : S \in \mathbb{S}\}$, where $\mathbb{S} = \{S \subseteq N : C(S) \le \boldsymbol{q}\}$ is the set of all feasible follower solutions, allows expressing the feasible region of the follower independent from the leader's decision by penalizing infeasible solutions where $\exists i \in S, x_i = 1$ with big-M_i. Then (1) can be restated as

$$\min w, \tag{10}$$
$$w \ge z(\hat{S}) - \sum_{i\in\hat{S}} M_i x_i \qquad \hat{S} \in \mathbb{S}, \tag{11}$$
$$G\boldsymbol{x} \le \boldsymbol{b}, \tag{12}$$
$$\boldsymbol{x} \in \{0,1\}^n. \tag{13}$$

We note that this reformulation follows the same ideas as proposed for IGs with a linear follower objective function (see, e.g., Israeli and Wood 2002, Caprara et al. 2016, Fischetti et al. 2019). In the linear case the interdiction cuts (11) can be written as $w \ge \boldsymbol{d}^T \hat{\boldsymbol{y}} - \sum_{i\in N} M_i x_i \hat{y}_i$, where $\hat{\boldsymbol{y}}$ is the vector of binary follower variables, and $\boldsymbol{d}$ is the vector of follower objective coefficients. Fischetti et al. (2019) prove the validity of the interdiction cut when $M_i = d_i$ under some assumptions. In the following, we present valid cuts for our problem in the form of (11) using the submodularity of $z(S)$. As is the case with a linear objective function, the values of the big-M coefficients determine the strength of the formulation.

3.2. Basic Submodular Interdiction Cut

The following theorem presents our basic SIC, which uses the marginal gains of the items in the ground set with respect to the empty set to replace the big-M values in (11).

Theorem 1. *Given a follower solution $\hat{S} \in \mathbb{S}$, the following basic SIC is valid for (6)–(9):*

$$w \ge z(\hat{S}) - \sum_{i\in\hat{S}} \rho_i(\emptyset) x_i. \tag{14}$$

Proof. For any feasible leader solution $\boldsymbol{x} \in X$, define the follower solution $S' = \hat{S} \setminus N_{\boldsymbol{x}}$. Because $S' \subseteq N \setminus N_{\boldsymbol{x}}$, and $C(S') \le C(\hat{S})$ due to nonnegativity of c_i^ℓ, S' is a feasible solution for $\boldsymbol{x}$. Due to $z(S)$ being submodular and monotone, and using (5), we have

$$z(\hat{S}) \le z(S') + \sum_{i\in\hat{S}\setminus S'} \rho_i(S') = z(S') + \sum_{i\in\hat{S}} \rho_i(S') x_i.$$

Thus we have

$$w \ge \Phi(\boldsymbol{x}) \ge z(S') \ge z(\hat{S}) - \sum_{i\in\hat{S}} \rho_i(S') x_i \ge z(\hat{S}) - \sum_{i\in\hat{S}} \rho_i(\emptyset) x_i,$$

which shows that the basic SIC for $\hat{S}$ is satisfied for any leader solution $\boldsymbol{x}$. The last inequality follows from the fact that $\rho_i(S') \le \rho_i(\emptyset)$, $i \in N$. □

Remark 1. In the special case of a modular monotone objective function $z(S) = \sum_{i\in S} d_i$, where $\forall i \in N, d_i \ge 0$, each marginal gain $\rho_i(S)$ can be replaced by d_i if $i \notin S$, and by zero otherwise. Then, the SICs we propose can be reduced to the cuts introduced in Fischetti et al. (2019). The downward monotonicity property follows from the nonnegativity of c_i^ℓ.

3.3. Improved Cut

In this section, we show how to obtain an improved version of (14) by exploiting the diminishing gains property of submodular functions, which implies that as the set expands the marginal gains decrease.

Theorem 2. *Given an arbitrary ordering $(i_1, i_2, \ldots, i_T)$ of the items in follower solution $\hat{S} \in \mathbb{S}$, let $\hat{S}_{(1)} = \emptyset$ and $\hat{S}_{(t)} = \{i_1, \ldots, i_{t-1}\}$ for $2 \le t \le T$. The following improved SIC is valid for (6)–(9) and it dominates (14):*

$$w \ge z(\hat{S}) - \sum_{t=1}^{T} \rho_{i_t}(\hat{S}_{(t)}) x_{i_t}. \tag{15}$$

Proof. For any $\boldsymbol{x} \in X$, define $S' = \hat{S} \setminus N_{\boldsymbol{x}}$, which is a feasible set for interdiction decision $\boldsymbol{x}$ (see the proof of Theorem 1). Let $S'_{(t)} = \hat{S}_{(t)} \setminus N_{\boldsymbol{x}}$ denote the items in $\hat{S}_{(t)}$ that are not interdicted in $\boldsymbol{x}$, that is, its maximal feasible subset. Note that $S'_{(t)} \subseteq S'_{(t+1)}$. Using the same ordering of the items in $\hat{S}$, the objective value of S' can be computed incrementally by using the definition of a marginal gain. Starting with an empty set, increasing the objective value at each step by the marginal gain of the next item with respect to the current set yields the objective value of the final set, as done in the following:

$$\begin{aligned} z(S') &= \rho_{i_1}(S'_{(1)})(1-x_{i_1}) + \rho_{i_2}(S'_{(2)})(1-x_{i_2}) + \dots \\ &\quad + \rho_{i_T}(S'_{(T)})(1-x_{i_T}) \\ &= \sum_{t=1}^{T} \rho_{i_t}(S'_{(t)})(1-x_{i_t}) \geq \sum_{t=1}^{T} \rho_{i_t}(\hat{S}_{(t)})(1-x_{i_t}). \end{aligned} \tag{16}$$

Here, if an item i_t is interdicted, its contribution is not included in the sum due to the $(1-x_{i_t})$ multiplier. Also note that $S'_{(t+1)} = S'_{(t)}$ when $x_{i_t} = 1$, since $S'_{(t+1)} = \hat{S}_{(t+1)} \setminus N_x = S'_{(t)} \cup \{i_t\} \setminus N_x$ by definition and $i_t \in N_x$. The last inequality is due to $\rho(\cdot)$ being nonincreasing and $S'_{(t)} \subseteq \hat{S}_{(t)}$. Note that $\sum_{t=1}^{T} \rho_{i_t}(\hat{S}_{(t)}) = z(\hat{S})$, again by definition of marginal gains. Since S' is a feasible follower solution for $\boldsymbol{x} \in X$, we have

$$\begin{aligned} w \geq \Phi(\boldsymbol{x}) \geq z(S') &\geq \sum_{t=1}^{T} \rho_{i_t}(\hat{S}_{(t)})(1-x_{i_t}) \\ &= z(\hat{S}) - \sum_{t=1}^{T} \rho_{i_t}(\hat{S}_{(t)}) x_{i_t}, \end{aligned}$$

which shows that (15) is valid for (6)–(9). It clearly dominates (14) because $\rho_{i_t}(\hat{S}_{(t)}) \leq \rho_{i_t}(\emptyset)$ for each $t \in \{1, \dots, T\}$. $\square$

Although any ordering of the items in a given follower solution gives a valid improved cut, we present heuristic separation procedures in Section 4 yielding the ordering to be used in cut generation.

3.4. Lifted Cut

Next, we propose a method to lift the basic and improved SIC based on the pairwise relationships between some items of the ground set N, which leads to the submodular version of the lifted cut proposed in Fischetti et al. (2019). In a sense, we are informing the model about possible other follower solutions with a better objective value, which can be obtained through the exchange of some items in the current set $\hat{S}$ with superior items outside of $\hat{S}$, if the leader does not interdict their usage. To be eligible for this type of exchange, an item pair (i, j) should satisfy the condition that replacing i with j does not increase the marginal gains of the items in $\hat{S}$ with respect to the rest of the set, as well as any other set obtained from $\hat{S}$ by means of exchanging such pairs, which is usually a result of superiority of j to i. We provide examples of such a relationship after proposing the lifted SIC.

We describe how to lift the improved cut (15) in the following theorem; the proof for the basic cut (14) works similarly and is omitted for brevity. We also give examples on how the condition mentioned earlier can occur in the problems considered in the computational study. Whereas enumerating all the sets described in Theorem 3 requires an exponential number of evaluations, we describe a heuristic procedure in Section 4.4 to efficiently generate a lifted cut, knowing that a lifted cut is not required for the correctness of the algorithm.

Theorem 3. *Given a follower set $\hat{S} \in \mathbb{S}$ and an ordering $(i_1, i_2, \dots, i_T)$ of its elements, let $A = \{a_1, \dots, a_K\} \subseteq \hat{S}$ and $B = \{b_1, \dots, b_K\} \subseteq N \setminus \hat{S}$ such that*

i. *$c^{\ell}_{a_k} \geq c^{\ell}_{b_k}$ for $\ell = 1, \dots, L$, and*
ii. *$\rho_i(S \cup \{b_k\} \setminus \{a_k\}) \leq \rho_i(S)$ for all S such that $(\hat{S} \setminus A) \cup \{a_k\} \subseteq S \subseteq (\hat{S} \cup B) \setminus \{b_k\}$, and $i \in \hat{S} \setminus S$,*

for each $k \in \{1, \dots, K\}$.

Define the subsets $\hat{S}_{(t)} = \{i_1, \dots, i_{t-1}\}$ for $T \geq t \geq 2$ and $\hat{S}_{(1)} = \emptyset$. Also define $A_{(k)} = \{a_1, \dots, a_k\}$ and $B_{(k)} = \{b_1, \dots, b_k\}$ for $k \in \{1, \dots, K\}$, and $A_{(0)} = B_{(0)} = \emptyset$. The following lifted cut is valid for (6)–(9):

$$\begin{aligned} w \geq z(\hat{S}) - \sum_{t=1}^{T} \rho_{i_t}(\hat{S}_{(t)}) x_{i_t} + \sum_{k=1}^{K} (\rho_{b_k}(\hat{S} \cup B_{(k-1)}) \\ - \rho_{a_k}(\hat{S} \cup \{b_k\} \setminus \{a_k\})(1 - x_{b_k}). \end{aligned} \tag{17}$$

Proof. Consider any feasible leader solution $\boldsymbol{x}$. If $x_{b_k} = 1$ for each $k \in \{1, \dots, K\}$, the lifted cut is valid because $\boldsymbol{x}$ satisfies the improved cut (15). Otherwise, let $\bar{K} = \{k \in \{1, \dots, K\} : x_{b_k} = 0\}$, $A_{\bar{K}} = \{a_k : k \in \bar{K}\}$ and $B_{\bar{K}} = \{b_k : k \in \bar{K}\}$. Define the set $S' = ((\hat{S} \setminus A_{\bar{K}}) \cup B_{\bar{K}}) \setminus N_x$, which is feasible for $\boldsymbol{x}$ due to condition (*i*). Since its feasibility implies that $w \geq \Phi(\boldsymbol{x}) \geq z(S')$, showing that $z(S')$ is greater than or equal to the right-hand side (RHS) of (17) would prove the validity of the cut. To this end, we define an intermediate set $S'' = (\hat{S} \setminus A_{\bar{K}}) \cup B_{\bar{K}})$ and compute the following bound on $z(S'')$ as if at each step we add one b_k, $k \in \bar{K}$, to $\hat{S}$ and then remove a_k from the set:

$$\begin{aligned} z(S'') &= z(\hat{S}) + \sum_{k \in \bar{K}} \rho_{b_k}(\hat{S} \cup (B_{(k-1)} \cap B_{\bar{K}}) \setminus (A_{(k-1)} \cap A_{\bar{K}})) \\ &\quad - \sum_{k \in \bar{K}} \rho_{a_k}(\hat{S} \cup (B_{(k)} \cap B_{\bar{K}}) \setminus (A_{(k)} \cap A_{\bar{K}})) \\ &\geq z(\hat{S}) + \sum_{k \in \bar{K}} \rho_{b_k}(\hat{S} \cup B_{(k-1)}) \\ &\quad - \sum_{k \in \bar{K}} \rho_{a_k}(\hat{S} \cup (B_{(k)} \cap B_{\bar{K}}) \setminus (A_{(k)} \cap A_{\bar{K}})). \end{aligned} \tag{18}$$

It is possible to further simplify the RHS of the previous inequality as follows by using condition (ii). The first inequality in the following is obtained by choosing $S = \hat{S} \cup (\{b_2, \ldots, b_k\} \cap B_{\bar{K}}) \setminus (\{a_2, \ldots, a_k\} \cap A_{\bar{K}})$ to have the inequality in (ii), that is, marginal gain becomes larger when b_1 is replaced with a_1. Applying this at each step, we reach the set $\hat{S} \cup \{b_k\} \setminus \{a_k\}$:

$$\begin{aligned}
&\rho_{a_k}(\hat{S} \cup (B_{(k)} \cap B_{\bar{K}}) \setminus (A_{(k)} \cap A_{\bar{K}}))\\
&= \rho_{a_k}(\hat{S} \cup (\{b_1, \ldots, b_k\} \cap B_{\bar{K}}) \setminus (\{a_1, \ldots, a_k\} \cap A_{\bar{K}}))\\
&\le \rho_{a_k}(\hat{S} \cup (\{b_2, \ldots, b_k\} \cap B_{\bar{K}}) \setminus (\{a_2, \ldots, a_k\} \cap A_{\bar{K}}))\\
&\vdots\\
&\le \rho_{a_k}(\hat{S} \cup (\{b_k\} \cap B_{\bar{K}}) \setminus (\{a_k\} \cap A_{\bar{K}}))\\
&\le \rho_{a_k}(\hat{S} \cup \{b_k\} \setminus \{a_k\}).
\end{aligned}$$

The last inequality is due to $k \in \bar{K}$, that is, $a_k \in A_{\bar{K}}$ and $b_k \in B_{\bar{K}}$. We rewrite the inequality in (18) as

$$\begin{aligned}
z(S'') &\ge z(\hat{S}) + \sum_{k \in \bar{K}} \rho_{b_k}(\hat{S} \cup B_{(k-1)}) - \sum_{k \in \bar{K}} \rho_{a_k}(\hat{S} \cup \{b_k\} \setminus \{a_k\})\\
&= z(\hat{S}) + \sum_{k=1}^{K} (\rho_{b_k}(\hat{S} \cup B_{(k-1)}) - \rho_{a_k}(\hat{S} \cup \{b_k\} \setminus \{a_k\}))(1 - x_k).
\end{aligned} \tag{19}$$

The equality follows from that $x_k = 1$ for $k \notin \bar{K}$. In the next step, we evaluate the objective value of $S' = S'' \setminus N_x$ using the identity obtained in (16). Let $(j_1, \ldots, j_T)$ be an ordering of the items in S'', that is identical to $(i_1, \ldots, i_T)$ except that each a_k, $k \in \bar{K}$, is replaced with b_k, that is, $j_t = i_t$ for $j_t \in S'' \setminus B_{\bar{K}}$ and $j_t = b_k \Longleftrightarrow i_t = a_k$ for $k \in \bar{K}$. Define the associated subsets $S''_{(t)} = \{j_1, \ldots, j_{t-1}\}$. Then, due to (16) we have

$$z(S') \ge z(S'') - \sum_{t=1}^{T} \rho_{j_t}(S''_{(t)}) x_{j_t} = z(S'') - \sum_{t=1}^{T} \rho_{i_t}(S''_{(t)}) x_{i_t}. \tag{20}$$

The reason for the equality is that $x_{j_t} = 0$ for $j_t \in B_{\bar{K}}$ and $j_t = i_t$ for $j_t \in S'' \setminus B_{\bar{K}}$. Since $S''_{(t)}$ is obtained through the exchange of some a_k with b_k, condition (ii) implies that $\rho_{i_t}(S''_{(t)}) \le \rho_{i_t}(\hat{S}_{(t)})$. Along with (19) and (20), this inequality leads to

$$\begin{aligned}
z(S') &\ge z(S'') - \sum_{t=1}^{T} \rho_{i_t}(\hat{S}_{(t)}) x_{i_t}\\
&\ge z(\hat{S}) + \sum_{k=1}^{K} (\rho_{b_k}(\hat{S} \cup B_{(k-1)})\\
&\quad - \rho_{a_k}(\hat{S} \cup \{b_k\} \setminus \{a_k\}))(1 - x_k) - \sum_{t=1}^{T} \rho_{i_t}(\hat{S}_{(t)}) x_{i_t},
\end{aligned}$$

which completes the proof. □

Remark 2. For a pair (a_k, b_k) satisfying condition (ii), it is possible that the coefficient $\rho_{b_k}(\hat{S} \cup B_{(k-1)}) - \rho_{a_k}(\hat{S} \cup \{b_k\} \setminus \{a_k\})$ of the last term in (17) is negative. Even if the items are selected in such a way that b_k can replace a_k without any sacrifice in solution quality, the coefficient can still be negative since its components constitute bounds on the true change in the objective due to adding b_k and removing a_k, respectively. Obviously, (17) dominates the improved cut (15) only if the coefficients of the $(1 - x_{b_k})$ terms are nonnegative. Because one can choose the sets A and B accordingly, we assume that those coefficients are positive and call (17) a *lifted cut*.

Remark 3. Condition (ii) in Theorem 3 can be interpreted as a result of superiority between item pairs. Although this type of relationship might seem difficult to detect, it becomes more intuitive on an application basis. For BIIG, (ii) results when the set of neighbors (connected targets) of a_k is a subset of the neighbors of b_k, and $p_{b_k} \ge p_{a_k}$. Thus, if a_k is replaced by b_k, the objective is at least as large as before and the marginal gains of $i \in \hat{S}$ with respect to rest of the set are not larger than before. In some problems, such as WMCIG, superiority has stronger implications. There, for a facility pair (a_k, b_k), if $J(a_k) \subseteq J(b_k)$, then $\rho_{a_k}(\{b_k\}) = 0$. In other words, including b_k in the set renders a_k completely useless and removing a_k does not damage the objective anymore. Note that condition (ii) holds for any pair with $\rho_{a_k}(\{b_k\}) = 0$, although the reverse is not necessarily true. If $\rho_{a_k}(\{b_k\}) = 0$ for each $k \in \{1, \ldots, K\}$, then (17) is reduced to

$$w \ge z(\hat{S}) - \sum_{t=1}^{T} \rho_{i_t}(\hat{S}_{(t)}) x_{i_t} + \sum_{k=1}^{K} \rho_{b_k}(\hat{S} \cup B_{(k-1)})(1 - x_{b_k}). \tag{21}$$

This cut dominates (15) independent of the choices of A and B since the last term is always nonnegative.

Example 1. Consider an instance of the BIIG with three items $N = \{1, 2, 3\}$, three targets $J = \{a, b, c\}$, and the arc list $A = \{(1, a), (2, a), (2, b), (3, a), (3, c)\}$. Let the activation probabilities of the items be given as $p_1 = 0.3$, $p_2 = 0.5$, and $p_3 = 0.4$. Recall that the objective function of the follower is $z(S) = \sum_{j \in J}(1 - \prod_{i \in S:(i,j) \in A}(1 - p_i))$, therefore $z(\emptyset) = 0$ and the gains with respect to the empty set are $\rho_1(\emptyset) = 0.3 + 0 + 0 = 0.3$, $\rho_2(\emptyset) = 0.5 + 0.5 + 0 = 1$, and $\rho_3(\emptyset) = 0.4 + 0 + 0.4 = 0.8$ Consider the set $\hat{S} = \{1, 2\}$ with objective value $z(\hat{S}) = 1 - (1 - 0.3)(1 - 0.5) + 0.5 + 0 = 1.15$. The associated basic SIC is $w \ge 1.15 - 0.3x_1 - x_2$. If we use the ordering $i_1 = 1, i_2 = 2$, the improved cut becomes $w \ge 1.15 - 0.3x_1 - 0.85x_2$.

Now consider the items $1 \in \hat{S}$ and $3 \notin \hat{S}$. Note that they do not satisfy the special condition in Remark 3, that is, $\rho_1(\{3\}) = z(\{1, 3\}) - z(\{3\}) \ne 0$. For condition (ii) in Theorem 3, it is sufficient to check if $\rho_2(\{3\}) \le \rho_2(\{1\})$ because only $i = 2$ and $S = \{1\}$ fit the definition given. We have that $\rho_2(\{3\}) = z(\{2, 3\}) - z(\{3\}) = 1 -$

$(1-0.5)(1-0.4)+0.5+0.4-0.8=1.6-0.8=0.8$ and $\rho_2(\{1\})=z(\{1,2\})-z(\{1\})=1.15-0.3=0.85$, thus the condition is satisfied and a better solution can be found by replacing 1 with 3. For $A=\{1\}$ and $B=\{3\}$, the coefficient of the lifting term is $\rho_3(\{1,2\})-\rho_1(\{2,3\}) = z(\{2,3\})-z(\{1,2\})=1.6-1.15=0.45$. The improved cut is lifted to $w\ge 1.15-0.3x_1-0.85x_2+0.45(1-x_3)$ according to Theorem 3.

Example 2. Consider an instance of WMCIG with four customers $J=\{a,b,c,d\}$ and three potential facility locations $N=\{1,2,3\}$, and the maximum number of facilities to open $q=2$. Let the customers covered by each location be given as $J(1)=\{a,c\}$, $J(2)=\{a,b\}$, $J(3)=\{a,c,d\}$, and the profits of the customers are $p_a=5$, $p_b=9$, $p_c=6$, and $p_d=4$. Now consider the set $\hat{S}=\{1,2\}$ with a total profit $z(\hat{S})=5+9+6=20$. The basic SIC for $\hat{S}$ is unique and $w\ge 20-11x_1-14x_2$. Because the size of $\hat{S}$ is two, there are two ways to generate the improved cut: $w\ge 20-11x_1-9x_2$ and $w\ge 20-6x_1-14x_2$. The facility locations $1\in\hat{S}$ and $3\notin\hat{S}$ satisfy the condition in Remark 3 as $\rho_1(\{3\})=z(\{1,3\})-z(\{3\})=0$. If we define the sets $A=\{1\}$ and $B=\{3\}$, using $\rho_3(\{1,2\})=4$ the first improved cut can be lifted to $w\ge 20-11x_1-9x_2+4(1-x_3)$. Similarly, the second one is lifted to $w\ge 20-6x_1-14x_2+4(1-x_3)$. If $x_3=1$, the lifted cuts are identical to their nonlifted versions, otherwise they yield a better cut.

Example 3. Another well-known application of submodular maximization is the uncapacitated facility location problem, where given the set of facility locations N, customers M, and the value $c_{i,j}$ of facility i to customer j, the objective is to find $S\subseteq N, |S|\le k$ maximizing $z(S)=\sum_{j\in M}\max_{i\in S}c_{i,j}$ (Nemhauser et al. 1978). This type of objective functions could cover several other applications, including sensor placement (Krause and Golovin 2014). Here, if locations $a_k\in S$ and $b_k\notin S$ satisfy $c_{a_k,j}\le c_{b_k,j},\ \forall j\in M$, then they also satisfy condition (ii) in Theorem 3. In other words, since b_k has a higher value for each customer, if a_k is replaced with b_k, the gain due to opening an additional facility i would not be greater. Any customer switching from b_k to i (and thus causing gain due to adding i) would also switch from a_k to i.

Remark 4. If the term $\rho_{i_t}(\hat{S}_{(t)})$ in the lifted cut is replaced by $\rho_{i_t}(\emptyset)$, the resulting cut is the lifted version of the basic SIC (14) and it is obviously valid and dominated by the current one since $\rho_{i_t}(\hat{S}_{(t)})\le\rho_{i_t}(\emptyset)$.

3.5. Alternative Cut

In this section, we propose a method to obtain a new cut that could feed the model with an alternative follower solution set, in case some of the items in the current set are interdicted. This cut generalizes the modified cut in Fischetti et al. (2019) to the submodular case, and unlike the lifted cut, it is not based on superiority implying relationships between item pairs, but on the relative availability of the items.

Theorem 4. *Given a follower set $\hat{S}\in\mathbb{S}$ and an ordering $(i_1,i_2,\dots,i_T)$ of its elements, let $A=\{a_1,\dots,a_K\}\subseteq\hat{S}$ and $B=\{b_1,\dots,b_K\}\subseteq N\setminus\hat{S}$ such that $c^{\ell}_{a_k}\ge c^{\ell}_{b_k}$ for $\ell=1,\dots,L$ and for each $k\in\{1,\dots,K\}$. Define the subsets $\hat{S}_{(t)}=\{i_1,\dots,i_{t-1}\}$ for $T\ge t\ge 2$ and $\hat{S}_{(1)}=\emptyset$. Also define $B_{(k)}=\{b_1,\dots,b_k\}$ for $k\in\{1,\dots,K\}$ and $B_{(0)}=\emptyset$. The following alternative cut is valid for (6)–(9):*

$$w\ge z(\hat{S})-\sum_{t=1}^{T}\rho_{i_t}(\hat{S}_{(t)})x_{i_t} + \sum_{k=1}^{K}\rho_{b_k}(\hat{S}\cup B_{(k-1)}\setminus\{a_k\})(x_{a_k}-x_{b_k}). \tag{22}$$

Proof. Let $\boldsymbol{x}$ be a feasible leader solution. If $x_{a_k}-x_{b_k}\le 0$ for all $k\in\{1,\dots,K\}$, then $\boldsymbol{x}$ satisfies (22) since it already satisfies (15) and $\rho(\cdot)$ is nonnegative by (4). Otherwise, let $\bar{K}=\{k\in\{1,\dots,K\}: x_{a_k}-x_{b_k}>0\}$ denote the index set of the a_k, b_k pairs such that $x_{a_k}=1$ and $x_{b_k}=0$. Define $A_{\bar{K}}=\{a_k: k\in\bar{K}\}$ and $B_{\bar{K}}=\{b_k:k\in\bar{K}\}$. Consider the set $S'=((\hat{S}\setminus A_{\bar{K}})\cup B_{\bar{K}})\setminus N_{\boldsymbol{x}}$, which is feasible for $\boldsymbol{x}$ under the assumption that $c^{\ell}_{a_k}\ge c^{\ell}_{b_k}$ for $\ell=1,\dots,L$ and for each $k\in\bar{K}$. Due to the definition of $A_{\bar{K}}$ and $B_{\bar{K}}$, we have that $A_{\bar{K}}\subseteq N_x$ and $B_{\bar{K}}\cap N_x=\emptyset$. Thus, $S'=(\hat{S}\setminus N_{\boldsymbol{x}})\cup B_{\bar{K}}$ and a lower bound on its objective value can be obtained by estimating the incremental change in the objective value due to adding each $b_k\in B_{\bar{K}}$ to $\hat{S}\setminus N_{\boldsymbol{x}}$ as follows:

$$\begin{aligned} z(S') &= z(\hat{S}\setminus N_{\boldsymbol{x}})+\sum_{k\in\bar{K}}\rho_{b_k}(\hat{S}\setminus N_x\cup(B_{(k-1)}\cap B_{\bar{K}}))\\ &\ge z(\hat{S}\setminus N_{\boldsymbol{x}})+\sum_{k\in\bar{K}}\rho_{b_k}(\hat{S}\setminus a_k\cup B_{(k-1)})\\ &\ge z(\hat{S}\setminus N_{\boldsymbol{x}})+\sum_{k=1}^{K}\rho_{b_k}(\hat{S}\setminus a_k\cup B_{(k-1)})(x_{a_k}-x_{b_k}). \end{aligned}\tag{23}$$

The reason for the first inequality is that $a_k\in N_{\boldsymbol{x}}$ for $k\in\bar{K}$ and $\rho(\cdot)$ is nonincreasing. The second inequality follows from that $x_{a_k}-x_{b_k}=1$ for $k\in\bar{K}$ and $x_{a_k}-x_{b_k}\le 0$ for $k\notin\bar{K}$. Due to (16) in the proof of Theorem 2, we have that $z(\hat{S}\setminus N_{\boldsymbol{x}})\ge z(\hat{S})-\sum_{t=1}^{T}\rho_{i_t}(\hat{S}_{(t)})x_{i_t}$. Thus, $z(\hat{S}\setminus N_{\boldsymbol{x}})$ in (23) can be replaced by its lower bound, yielding

$$w\ge\Phi(\boldsymbol{x})\ge z(S')\ge z(\hat{S})-\sum_{t=1}^{T}\rho_{i_t}(\hat{S}_{(t)})x_{i_t} + \sum_{k=1}^{K}\rho_{b_k}(\hat{S}\setminus a_k\cup B_{(k-1)})(x_{a_k}-x_{b_k}),$$

and hereby completing the proof. □

Theorem 4 can be interpreted as follows. If an item in $\hat{S}$ is interdicted and therefore removed from the solution, one could obtain a better solution by including a noninterdicted item whose costs are at most as large as of the former item. However, the new cut can be worse than the original one as $x_{a_k} - x_{b_k}$ can take a negative value. Thus, (22) is not necessarily a lifted cut, but an alternative to the original basic/improved SIC. Although there can be an exponential number of sets fitting into the condition in Theorem 4, it is enough to find one such pair to generate an alternative cut, for example, via a heuristic method such as the one we propose in Section 4.4. As is the case with the lifted cuts, an alternative cut can be obtained from a basic cut (14) instead of an improved one by simply replacing $\rho_{i_t}(\hat{S}_{(t)})$ in (22) by $\rho_{i_t}(\emptyset)$. However, it would be a weaker cut than (22).

Example 4. Consider the WMCIG instance in Example 2. Suppose that we are given the same set $\hat{S} = \{1,2\}$ and asked to obtain the alternative cut for the ordering $i_1 = 1, i_2 = 2$ and $A = \{1\}$, $B = \{3\}$. The coefficient of the additional term would be $\rho_3(\hat{S} \setminus \{1\}) = \rho_3(\{2\}) = 24 - 14 = 10$. Using the improved cut from Example 2, we obtain the alternative cut

$$w \geq 20 - 11x_1 - 9x_2 + 10(x_1 - x_3).$$

Now consider two interdiction strategies $\boldsymbol{x}^{(1)} = (0,0,1)$ and $\boldsymbol{x}^{(2)} = (1,0,0)$. Whereas $\boldsymbol{x}^{(1)}$ yields an alternative cut worse than the improved one ($w \geq 10$ instead of $w \geq 20$), the alternative cut is better for $\boldsymbol{x}^{(2)}$ ($w \geq 19$ instead of $w \geq 9$).

Remark 5. There is no dominance relationship between the lifted cut (17) and the alternative cut (22) even if the cuts are obtained for the same follower solution $\hat{S}$, the same ordering of the items, and the same exchange sets A and B. Consider Examples 2 and 4. For $\hat{S} = \{1,2\}$, $i_1 = 1, i_2 = 2$, $A = \{1\}$, and $B = \{3\}$, the lifted cut and the alternative cut are $w \geq 20 - 11x_1 - 9x_2 + 4(1 - x_3)$ and $w \geq 20 - 11x_1 - 9x_2 + 10(x_1 - x_3)$, respectively. At $\boldsymbol{x}^{(1)} = (0,0,1)$, these inequalities become $w \geq 20$ and $w \geq 10$, whereas at $\boldsymbol{x}^{(2)} = (1,0,0)$ they become $w \geq 13$ and $w \geq 19$, respectively.

4. Implementation Details

In this section, we propose a branch-and-cut (B&C) scheme to solve Problem (1). We explain the details of the separation of SICs after we provide some observations that can be exploited for a more efficient implementation.

4.1. Dominance Inequalities

The following dominance inequalities can be added to remove some feasible solutions, but it is guaranteed that not all optimal solutions will be cut off. Detection of the item-pairs fitting Theorem 5 depends on the problem structure. Thus, after the main theorem, we give propositions on how to detect them in our considered applications.

Theorem 5. *Let G_i denote the ith column of the matrix G in the leader's problem. If a pair of items $i,j \in N$ satisfies $c_i^\ell \leq c_j^\ell$ for $\ell = 1,\dots,L$, $\rho_i(S) \geq \rho_j(S)$ for all $S \subseteq N \setminus \{i,j\}$ and $G_i \leq G_j$, then the inequality $x_i \geq x_j$ does not cut off all optimal solutions to (6)–(9) if $x_j \geq x_i$ is not already present in the model.*

Proof. Suppose that all optimal solutions are eliminated by dominance inequalities and $\boldsymbol{x}^*$ is one of them. For the sake of simplicity, assume that the model includes exactly one such inequality, which is $x_i \geq x_j$. Because $\boldsymbol{x}^*$ is cut by the dominance inequality, we should have $x_i^* < x_j^*$, that is, $x_i^* = 0$ and $x_j^* = 1$. Now define $\boldsymbol{x}'$ identical to $\boldsymbol{x}^*$ except that $x_i' = 1$, $x_j' = 0$. $\boldsymbol{x}'$ satisfies the dominance inequality, and it is feasible because $G_i \leq G_j$. Now, let S' be an optimal follower response to $\boldsymbol{x}'$. If $j \notin S'$, then S' is also feasible for $\boldsymbol{x}^*$, and $\Phi(\boldsymbol{x}^*) \geq z(S') = \Phi(\boldsymbol{x}')$. Otherwise, $(S' \setminus \{j\}) \cup \{i\}$ is feasible for $\boldsymbol{x}^*$ due to the assumption $c_i^\ell \leq c_j^\ell$ for $\ell = 1,\dots,L$. Moreover, because $\rho_i(S) \geq \rho_j(S)$ for all $S \subseteq N \setminus \{i,j\}$, we have $\rho_i(S' \setminus \{j\}) \geq \rho_j(S' \setminus \{j\})$, which implies by definition of $\rho(\cdot)$ that

$$z((S' \setminus \{j\}) \cup \{i\}) \geq z(S').$$

As a result, $\Phi(\boldsymbol{x}^*) \geq z((S' \setminus \{j\}) \cup \{i\}) \geq z(S') = \Phi(\boldsymbol{x}')$, and it shows that $\boldsymbol{x}'$ is also optimal. In case the model includes $n > 1$ dominance inequalities $x_{i_k} \geq x_{j_k}$, $k \in \{1,\dots,n\}$, the same procedure applies: for each violated inequality k set $x_{i_k}' = 1$ and $x_{j_k}' = 0$, and the same result follows. □

In our implementation, for WMCIG instances we use the condition in Proposition 3, given next, to detect pairs that fit into the description in Theorem 5, due to problem characteristics discussed in Remark 3. For BIIG, on the other hand, we check the condition in Proposition 4 for each pair. We add the resulting dominance inequalities to the initial model. If the conditions are fulfilled in both directions, then the items can substitute each other, and only one of the resulting inequalities is used.

Proposition 3. *Given a submodular, monotone and normalized function z, a ground set N, and $i,j \in N$, if $z(\{i,j\}) = \rho_i(\emptyset)$, then $\rho_i(S) \geq \rho_j(S)$, $\forall S \subseteq N$.*

Proof. Given that z is normalized, that is, $z(\emptyset) = 0$, $z(\{i,j\}) = \rho_i(\emptyset) + \rho_j(\{i\})$. If $z(\{i,j\}) = \rho_i(\emptyset)$, then $\rho_j(\{i\}) = \rho_j(\{i\} \cup S) = 0$ for all $S \subseteq N$. We need to show that $\rho_i(S) - \rho_j(S) \geq 0$. By definition of marginal gains, we have $\rho_i(S) - \rho_j(S) = z(S \cup \{i\}) - z(S \cup \{j\})$. Using the submodular inequality (2), we can write

$$z(S \cup \{j\}) \leq z(S \cup \{i\}) + \rho_j(S \cup \{i\}) - \rho_i(S \cup \{j\}).$$

Because $\rho_j(S \cup \{i\}) = 0$, the inequality becomes

$$z(S \cup \{i\}) - z(S \cup \{j\}) \geq \rho_i(S \cup \{j\}) \geq 0,$$

which completes the proof. □

Proposition 4. *Given an instance of the BIIG for the bipartite graph $G = (N, J, A)$, let $J(i) = \{k \in J : (i,k) \in A\}$ denote the target set of each item $i \in N$. Let a pair of items $i, j \in N$ satisfy $J(j) \subseteq J(i)$ and $p_i \geq p_j$. Then, $\rho_i(S) \geq \rho_j(S)$, $\forall S \subseteq N \setminus \{i,j\}$.*

Proof. Using the definition of z under BIIG, we have for each $S \in N \setminus \{i,j\}$ that

$$\begin{aligned}\rho_i(S) = z(S \cup \{i\}) - z(S) &= \sum_{k \in J}\left(1 - \prod_{i' \in S \cup \{i\}:(i',k) \in A}(1 - p_{i'})\right)\\ &\quad - \sum_{k \in J}\left(1 - \prod_{i' \in S:(i',k) \in A}(1 - p_{i'})\right)\\ &= \sum_{k \in J(i)}\left(\prod_{i' \in S:(i',k) \in A}(1 - p_{i'}) - \prod_{i' \in S \cup \{i\}:(i',k) \in A}(1 - p_{i'})\right)\\ &\quad + \sum_{k \in J \setminus J(i)}\left(\prod_{i' \in S:(i',k) \in A}(1 - p_{i'}) - \prod_{i' \in S \cup \{i\}:(i',k) \in A}(1 - p_{i'})\right)\\ &= \sum_{k \in J(i)} \prod_{i' \in S:(i',k) \in A}(1 - p_{i'})p_i\\ &\geq \sum_{k \in J(j)} \prod_{i' \in S:(i',k) \in A}(1 - p_{i'})p_j = \rho_j(S).\end{aligned}$$

The reason for the first equality in the last line is that for each $k \in J \setminus J(i)$ the two products are identical, that is, $\{i' \in S : (i',k) \in A\} = \{i' \in S \cup \{i\} : (i',k) \in A\}$. To put it simply, only the activation probabilities of $k \in J(i)$ are affected due to adding i to S. The inequality follows from the assumptions $J(j) \subseteq J(i)$ and $p_i \geq p_j$, in addition to all terms in the product being nonnegative. □

4.2. Maximal Follower Solutions

A follower solution $\hat{S} \in \mathbb{S}$ is called maximal if there is no $S' \in \mathbb{S}$ such that $\hat{S} \subset S'$. Fischetti et al. (2019) consider only maximal follower solutions while separating their interdiction cuts because for their setting they showed that their proposed interdiction cut for a maximal solution $\hat{\boldsymbol{y}}$ dominates the one for $\boldsymbol{y}' < \hat{\boldsymbol{y}}$.

Theorem 6. *Consider a nonmaximal follower solution $S' \in \mathbb{S}$ and $\hat{S} \supset S'$. The basic interdiction cut (14) for $\hat{S}$ does not dominate the one for S'.*

Proof. Suppose that the basic interdiction cut for $\hat{S}$ dominates the one for S'. Then the RHS of the basic cut (14) for S' should be less than or equal to the RHS of the cut for $\hat{S}$, for all $\boldsymbol{x} \in X$. Subtracting the former from the latter yields

$$\begin{aligned}&z(\hat{S}) - \sum_{i \in \hat{S}} \rho_i(\emptyset)x_i - z(S') + \sum_{i \in S'} \rho_i(\emptyset)x_i\\ &= z(\hat{S}) - z(S') - \sum_{i \in \hat{S} \setminus S'} \rho_i(\emptyset)x_i \leq \sum_{i \in \hat{S} \setminus S'} \rho_i(S') - \sum_{i \in \hat{S} \setminus S'} \rho_i(\emptyset)x_i\\ &= \sum_{i \in \hat{S} \setminus S'} (\rho_i(S') - \rho_i(\emptyset)x_i).\end{aligned}$$

The inequality sign comes from the submodular inequality (5). Consider the case that $x_i = 1$ for $i \in \hat{S} \setminus S'$. The difference will be nonpositive because $\rho_i(S') \leq \rho_i(\emptyset)$, which is a contradiction. □

Theorem 7. *Consider a nonmaximal follower solution $S' \in \mathbb{S}$ with the ordering $(i_1, \ldots, i_T)$ and $\hat{S} \supset S'$ with ordering $(i_1, \ldots, i_{T+k})$, where $T, k > 0$. The improved interdiction cut (15) for $\hat{S}$ dominates the one for S'.*

Proof. We need to show that the RHS of (15) for S' is less than or equal to the RHS of the cut for $\hat{S}$ when the given orderings are used to generate the cuts. Define $S'_{(t)} = \hat{S}_{(t)} = \{i_1, \ldots, i_{t-1}\}$ for $T \geq t \geq 2$, $\hat{S}_{(t)} = \{i_1, \ldots, i_{t-1}\}$ for $T + k \geq t > T$, and $\hat{S}_{(1)} = S'_{(1)} = \emptyset$. Then, the difference that needs to be proven nonnegative is

$$\begin{aligned}&z(\hat{S}) - \sum_{t=1}^{T+k} \rho_{i_t}(\hat{S}_{(t)})x_{i_t} - z(S') + \sum_{t=1}^{T} \rho_{i_t}(S'_{(t)})x_{i_t}\\ &= z(\hat{S}) - z(S') - \sum_{t=T+1}^{T+k} \rho_{i_t}(\hat{S}_{(t)})x_{i_t}.\end{aligned}$$

Using $S' \subset \hat{S}$ and $\hat{S} \setminus S' = \{i_{T+1}, \ldots, i_{T+k}\}$, the difference $z(\hat{S}) - z(S')$ can be computed iteratively and the previous expression is rewritten as follows.

$$\begin{aligned}&\sum_{t=T+1}^{T+k} \rho_{i_t}(\hat{S}_{(t)}) - \sum_{t=T+1}^{T+k} \rho_{i_t}(\hat{S}_{(t)})x_{i_t}\\ &= \sum_{t=T+1}^{T+k} \rho_{i_t}(\hat{S}_{(t)})(1 - x_{i_t}) \geq 0. \quad \square\end{aligned}$$

Theorem 7 indicates that replacing a nonmaximal follower solution with a superset by appending new items to it without disrupting the initial ordering yields a better improved cut. This in turn means that one can keep appending items until reaching a maximal solution to obtain a better cut. On the other hand, if the improved cuts for $\hat{S}$ and $S' \subset \hat{S}$ are generated based on arbitrary orderings of their elements, it is not possible to claim that one cut is better than the other because the value of the differences on the RHS depends on $\boldsymbol{x}$.

4.3. Separation of Basic and Improved Submodular Interdiction Cuts

We have different separation procedures for SICs for integer and fractional solutions $\boldsymbol{x}^*$ encountered in our

B&C. We first discuss the separation of integer solutions and then the separation of fractional solutions.

4.3.1. Separation of Integer Solutions. Submodular inequalities (2) and (3) lead to cuts that can be used for solving the constrained submodular maximization problems, that is, the follower problem in (1), such as the ones proposed by Ahmed and Atamtürk (2011), Ljubić and Moreno (2018), and Coniglio et al. (2020). We formulate our separation problem (*SEP*) as an MIP using these single-level submodular cuts in the following. Given a leader solution $\boldsymbol{x}^* \in X$, let $N_{\text{-}\boldsymbol{x}} = N \setminus N_{\boldsymbol{x}} = \{i \in N : x_i = 0\}$ be the set of items available to the follower and $\boldsymbol{y} \in \{0,1\}^{|N_{\text{-}x}|}$ denote the characteristic vector of any follower solution $S \subseteq N_{\text{-}\boldsymbol{x}}$, that is, $S = \{i \in N_{\text{-}\boldsymbol{x}} : y_i = 1\}$. Then, (*SEP*) becomes

$$(SEP) \quad \Phi(\boldsymbol{x}) = \max \theta \tag{24}$$

$$\theta \le z(\hat{S}) + \sum_{i \in N_{\text{-}x} \setminus \hat{S}} \rho_i(\hat{S}) y_i - \sum_{i \in \hat{S}} \rho_i(N_{\text{-}x} \setminus \{i\})$$
$$(1 - y_i) \quad \hat{S} \subseteq N_{\text{-}\boldsymbol{x}}, \tag{25}$$

$$\theta \le z(\hat{S}) + \sum_{i \in N_{\text{-}x} \setminus \hat{S}} \rho_i(\emptyset) y_i - \sum_{i \in \hat{S}} \rho_i(\hat{S} \setminus \{i\})$$
$$(1 - y_i) \quad \hat{S} \subseteq N_{\text{-}\boldsymbol{x}}, \tag{26}$$

$$\sum_{i \in N} c_i^{\ell} y_i \le q_{\ell} \qquad \ell \in \{1, \dots, L\}, \tag{27}$$

$$y_i \in \{0, 1\} \qquad i \in N_{\text{-}\boldsymbol{x}}. \tag{28}$$

4.3.1.1. Solving the Separation Problem (SEP). The separation problem (*SEP*) can be solved via a branch-and-cut scheme where submodular cuts (25) and (26) are generated as they are needed. To this end, the formulation composed of (24), (27), and (28) is solved using an MILP solver. Let $(\theta^*, \boldsymbol{y}^*)$ be the solution of the linear program at the current (follower) B&C tree node. If $\boldsymbol{y}^*$ is integer feasible, then it defines a unique set $\hat{S} = \{i \in N_{\text{-}\boldsymbol{x}} : y_i^* = 1\}$ and the value of $z(\hat{S})$ is computed according to the definition of z. If $\theta^* > z(\hat{S})$, then (25) and (26) are generated for $\hat{S}$ (by evaluating the necessary marginal gains); otherwise, no cut is added. For fractional $\boldsymbol{y}^*$, $\hat{S}$ is obtained in a heuristic way as follows: $\hat{S}$ is initialized as an empty set, the items $i \in N_{\text{-}\boldsymbol{x}}$ are sorted in nonincreasing order of y_i^* values, and they are added to $\hat{S}$ in this order until a knapsack constraint (27) is violated. Constraints (25) and (26) are obtained for $\hat{S}$ and their amounts of violation are computed for $(\theta^*, \boldsymbol{y}^*)$. The violated cuts are added to the problem, if any.

4.3.1.2. The Separation Procedure. Now, let $(w^*, \boldsymbol{x}^*)$ be the optimal solution at the current B&C node with integer $\boldsymbol{x}^*$. The SICs are separated exactly as follows. First, the separation problem (*SEP*) is solved on $N_{\text{-}\boldsymbol{x}^*}$ as described earlier to obtain $\hat{S}$, which is defined by its optimal solution, and its objective value $z(\hat{S})$. If $w^* < z(\hat{S})$, then $\hat{S}$ yields a violated basic (14) or improved (15) SIC. Whereas the coefficients in a basic cut are independent of $\hat{S}$ and only computed once as a preprocessing step, the coefficients of an improved cut depend on $\hat{S}$ and require an ordering of its elements. For the latter, the items in $\hat{S}$ are sorted in nonincreasing order of $\rho_i(\emptyset)$ values, which performs slightly better than nondecreasing or random ordering in our preliminary experiments.

Although we need to separate integer solutions exactly for the correctness of our algorithm, it is also possible to first try to use a heuristic method to find a violated cut instead of solving the separation problem to optimality to potentially speed up the separation. We propose an enhanced exact separation procedure as an alternative to the method described earlier. We first implement a classical greedy algorithm (Algorithm 1), which outputs a feasible follower solution $\hat{S}$ (and an ordering O of the items in $\hat{S}$ to be used in the fractional separation procedures we describe next). If $\hat{S}$ leads to violated SIC, then we are done. Otherwise, we solve the separation problem (*SEP*) with a B&C until a desired solution is reached. The procedure is summarized in Algorithm 2, which returns $\hat{S}$ yielding a violated SIC, if there exists one. Otherwise, it returns an empty set, which implies that the current solution is the new incumbent. The ordering for the improved cut is obtained as before.

Algorithm 1. Greedy $(N, \hat{S}, O)$

while $\exists i \in N \setminus \hat{S}$ such that $C(\hat{S} \cup \{i\}) \le q$ **do**
  $i^* \leftarrow \arg\max_{i \in N \setminus \hat{S} : C(\hat{S} \cup \{i\}) \le q} \frac{\rho_i(\hat{S})}{\max_{\ell} c_i^{\ell}}$
  $\hat{S} \leftarrow \hat{S} \cup \{i^*\}$, $O.add(i^*)$
end while
Return $(\hat{S}, O)$

Algorithm 2. Enhanced Separation of Integer Solutions
Input: An integer feasible leader solution $(w^*, \boldsymbol{x}^*)$
Output: A follower solution $\hat{S} \in \mathbb{S}$

$\hat{S} \leftarrow \emptyset$, $N_{\text{-}\boldsymbol{x}} = \{i \in N : x_i^* = 0\}$, $O = \emptyset$
$(\hat{S}, O) \leftarrow Greedy(N_{\text{-}\boldsymbol{x}}, \hat{S}, O)$
if the SIC defined by $\hat{S}$ is not violated at $(w^*, \boldsymbol{x}^*)$ **then**
  Solve (*SEP*) until a feasible solution $\boldsymbol{y}^*$ with objective $\theta^* > w^*$ is found
  if a solution is found **then**
    $\hat{S} \leftarrow \{i \in N_{\text{-}\boldsymbol{x}} : y_i^* = 1\}$
  else
    There is no violated SIC, $\hat{S} \leftarrow \emptyset$
  end if
end if
Return $\hat{S}$

4.3.2. Separation of Fractional Solutions. At every node with a fractional $\boldsymbol{x}^*$, $\hat{S}$ is obtained in a greedy way. If the relative violation of the resulting cut exceeds the threshold of 1%, then it is added to the problem. This condition helps avoid large subproblems caused by an overload of cuts with small violation. The following three separation options are considered to obtain $\hat{S}$ and the ordering of its elements, which is required for the improved cut:

- S1 (see Algorithm 3): First, a temporary ground set is determined using only totally noninterdicted items, that is, $i : x^*_i = 0$, and then the *Greedy*$(\cdot)$ function is called. If the cut to be generated is an improved cut and the solution is not maximal, the *Greedy*$(\cdot)$ function is reinvoked to reach a maximal solution. This is done according to Theorem 7, that is, items are appended to the end of the current ordering by $O.add(\cdot)$.
- S2 (see Algorithm 4): The same procedure used for S1 is followed except the definition of the ground set. Here, it is obtained based on a rounding of $\boldsymbol{x}^*$, which allows us to include some items with fractional x^*_i in the ground set.
- S3 (see Algorithm 5): Another greedy approach is used to obtain a maximally violated basic/improved cut. The violation increase due to adding an item $i \in N \setminus \hat{S}$ to $\hat{S}$ is denoted by $v_i(\hat{S})$ and evaluated by $\rho_i(\hat{S}) - \rho_i(\emptyset)x^*_i$ for the basic cut and by $\rho_i(\hat{S})(1 - x^*_i)$ for the improved cut. The item with the maximum $v_i(\hat{S})$ (taking also the coefficients of i in follower constraints into consideration) is added to the set until the budget is reached or the maximum $v_i(\hat{S})$ is negative (only possible for the basic cut).

Algorithm 3. S1
Input: A fractional leader solution $\boldsymbol{x}^*$
Output: A follower solution $\hat{S} \in \mathbb{S}$ and an ordering of its elements
$\hat{S} \leftarrow \emptyset$, $N_{\text{-}\boldsymbol{x}} = \{i \in N : x^*_i = 0\}$, $O = \emptyset$
$(\hat{S}, O) \leftarrow$ *Greedy*$(N_{\text{-}\boldsymbol{x}}, \hat{S}, O)$
if *cutType* = *Improved* and $\exists i \in N \setminus \hat{S}$ such that $C(\hat{S} \cup \{i\}) \leq \boldsymbol{q}$ **then**
  $(\hat{S}, O) \leftarrow$ *Greedy*$(N, \hat{S}, O)$
end if
Return $\hat{S}$ and ordering $O = (i_1, \ldots, i_{|\hat{S}|})$

Algorithm 4. S2
Input: A fractional leader solution $\boldsymbol{x}^*$
Output: A follower solution $\hat{S} \in \mathbb{S}$ and an ordering of its elements
$\hat{S} \leftarrow \emptyset$, $O = \emptyset$, $\boldsymbol{x}' \leftarrow 0$
for each $i : x^*_i = 1$ **do**
  $x'_i \leftarrow 1$
end for
while $\exists i \in N : x'_i = 0, G(\boldsymbol{x}' + e_i) \leq \boldsymbol{b}$ **do**
  $i' \leftarrow \arg\max_{i \in N : x'_i = 0, G(\boldsymbol{x}' + e_i) \leq \boldsymbol{b}} x^*_i$
  $x'_{i'} \leftarrow 1$
end while
$N_{\text{-}\boldsymbol{x}} = \{i \in N : x'_i = 0\}$
$(\hat{S}, O) \leftarrow$ *Greedy*$(N_{\text{-}\boldsymbol{x}}, \hat{S}, O)$
if *cutType* = *Improved* and $\exists i \in N \setminus \hat{S}$ such that $C(\hat{S} \cup \{i\}) \leq q$ **then**
  $(\hat{S}, O) \leftarrow$ *Greedy*$(N, \hat{S}, O)$
end if
Return $\hat{S}$ and ordering $O = (i_1, \ldots, i_{|\hat{S}|})$

Algorithm 5. S3
Input: A fractional leader solution $\boldsymbol{x}^*$
Output: A follower solution $\hat{S} \in \mathbb{S}$ and an ordering of its elements
$\hat{S} \leftarrow \emptyset$, $O = \emptyset$
if *cutType* = *Improved* **then**
  $v_i(\hat{S}) := \rho_i(\hat{S})(1 - x^*_i)$
else
  $v_i(\hat{S}) := \rho_i(\hat{S}) - \rho_i(\emptyset)x^*_i$
end if
while $\exists i \in N \setminus \hat{S}$: $C(\hat{S} \cup \{i\}) \leq \boldsymbol{q}$ and $v_i(\hat{S}) \geq 0$ **do**
  $i^* \leftarrow \arg\max_{i \in N \setminus \hat{S} : C(\hat{S} \cup \{i\}) \leq \boldsymbol{q}} \frac{v_i(\hat{S})}{\max_\ell c^\ell_i}$
  $\hat{S} \leftarrow \hat{S} \cup \{i^*\}$, $O.add(i^*)$
end while
Return $\hat{S}$ and ordering $O = (i_1, \ldots, i_{|\hat{S}|})$

4.4. Separation of Lifted and Alternative Cuts

In our implementation, the lifted and alternative cuts are obtained heuristically, after the basic/improved cut is generated. For lifted cuts, as a preprocessing step the dominating list D_i, which contains the items that can replace i according to condition (ii) in Theorem 3, is computed for each $i \in N$ using the problem-specific implications of superiority described in Remark 3 before starting the B&C. Whenever a lifted cut is to be generated from a basic/improved cut for a follower solution $\hat{S}$, sets A and B are initialized as empty sets and determined incrementally as follows. The items in $\hat{S}$ are sorted in nonincreasing order of $\rho_i(\emptyset)$ values. The algorithm iterates over the items in $\hat{S}$ in this order. At each iteration, the next item $i \in \hat{S}$ is picked and the value of $(\rho_j(\hat{S} \cup B) - \rho_i(\hat{S} \cup \{j\} \setminus \{i\}))(1 - x^*_j)$ is checked for each $j \in D_i \setminus \hat{S}$. Note that condition (ii) is not checked at all, since all (i, j) pairs with $j \in D_i$ should satisfy it. If the maximum of these values is positive, i is added to A and the relevant j is added to B, and they are not considered in further evaluations. Once all $i \in \hat{S}$ are considered, the final cut is reached.

To generate an alternative cut after obtaining the basic/improved cut for a given $\hat{S}$, A and B are initialized as empty sets and obtained incrementally as follows. The algorithm iterates over the items in $\hat{S}$ in nonincreasing order of $\rho_i(\emptyset)$ values. At each iteration, the next item $i \in \hat{S}$ is picked and the value of $\rho_j(\hat{S} \cup B \setminus \{i\})(x^*_i - x^*_j)$ is checked for each $j \in N \setminus \hat{S}$, such that $c^\ell_i \geq c^\ell_j$ for each $\ell = 1, \ldots, L$. If the largest one of these values is positive, i is added to A, j is added to B, and they are not considered for further evaluations. Once all $i \in \hat{S}$ are processed, the resulting sets A and B yield the final alternative cut. Note that this procedure would not yield a new cut for an integer leader solution $\boldsymbol{x}^*$, as the integer separation procedure leads to $x_i^* = 0$ for $i \in \hat{S}$. For this reason, alternative cuts are only generated for fractional $\boldsymbol{x}^*$.

5. Computational Results

The algorithms we propose have been implemented in C++ using IBM ILOG CPLEX 12.10 as the MILP solver with its default settings. Each experiment uses a single thread of an Intel Xeon E5-2670v2 machine with 2.5 GHz processor. The time limit is 3,600 seconds and the memory allocated to each experiment is 12 gigabytes. We consider the two applications introduced in Section 2.2 and generate random data sets of them to test our framework. In the following sections we present the instance-generation procedures and the obtained results. All of the instances used are available at https://msinnl.github.io/pages/submodularinterdiction.html.

In our experiments, the following settings are considered for our B&C:

- B: Only the basic cut (14) is used for separation.
- I: Instead of a basic cut, an improved submodular interdiction cut (15) is separated.
- L: Once (14) or (15) is obtained, it is lifted heuristically to (17).
- D: Dominance inequalities are added to the initial model according to Theorem 5.
- A: In addition to the basic cut (14), improved cut (15), or lifted cut (17), the alternative cut (22) is generated heuristically.
- E: For the separation of integer solutions, the enhanced procedure in Algorithm 2 is used.

We include each of the components incrementally. The basic setting is B-s, where $s \in \{\text{S1, S2, S3}\}$ denotes the method used to obtain S for fractional $\boldsymbol{x}^*$ and it is followed by I-s, IL-s, ILD-s, ILDA-s, and finally ILDAE-s, which includes all the improvements and cut types we propose.

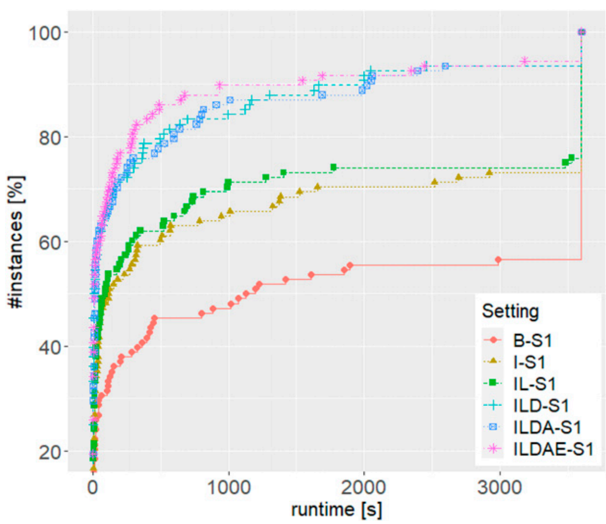
5.1. Weighted Maximal Covering Interdiction Game

WMCIG instances used in our study are generated following a similar procedure proposed by ReVelle et al. (2008). Customer coordinates are generated randomly in $[0,10]$. Potential facility locations are the same as the current customer locations, that is, $n = m$, and $m \in \{50, 60, 70, 80, 90, 100\}$. The profits p_j, $\forall j$ are randomly generated in $[1,100]$. Coverage is determined based on Euclidean distances and radius of coverage $r \in \{1,2,3\}$, that is, a facility at location i covers customer j if $d_{ij} \le r$, where d_{ij} is the Euclidean distance between i and j. The number of facilities to open is $q = 0.1n$ and the interdiction budget b takes value in $\{0.1n, 0.2n\}$. Three instances are generated for each (n, r, b) combination.

Recall that condition (ii) of Theorem 3 is equivalent to $\rho_{a_k}(b_k) = 0$ for WMCIG, as explained in Remark 3. Therefore, the lifted cuts generated for this problem are in the form of (21).

The plots of the results in terms of running times and final optimality gaps for each separation option $s \in \{S1, S2, S3\}$ are provided in Figures 1, 2, and 3, respectively. The optimality gaps are obtained by $100 \times (z^* - \underline{z})/(0.1 + z^*)$, where z^* and $\underline{z}$ denote the objective value of the best integer solution and the best bound, respectively. Time spent for all the subroutines such as obtaining lifted/alternative cuts and adding dominance inequalities are included in the solution time. We see in Figure 1 that using improved

Figure 1. (Color online) WMCIG S1 Results



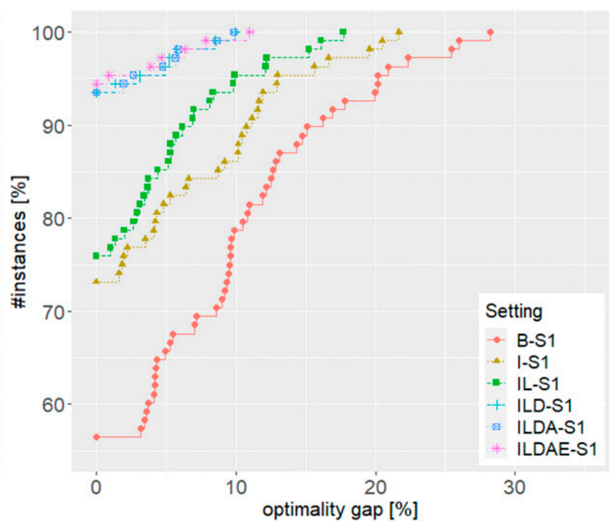
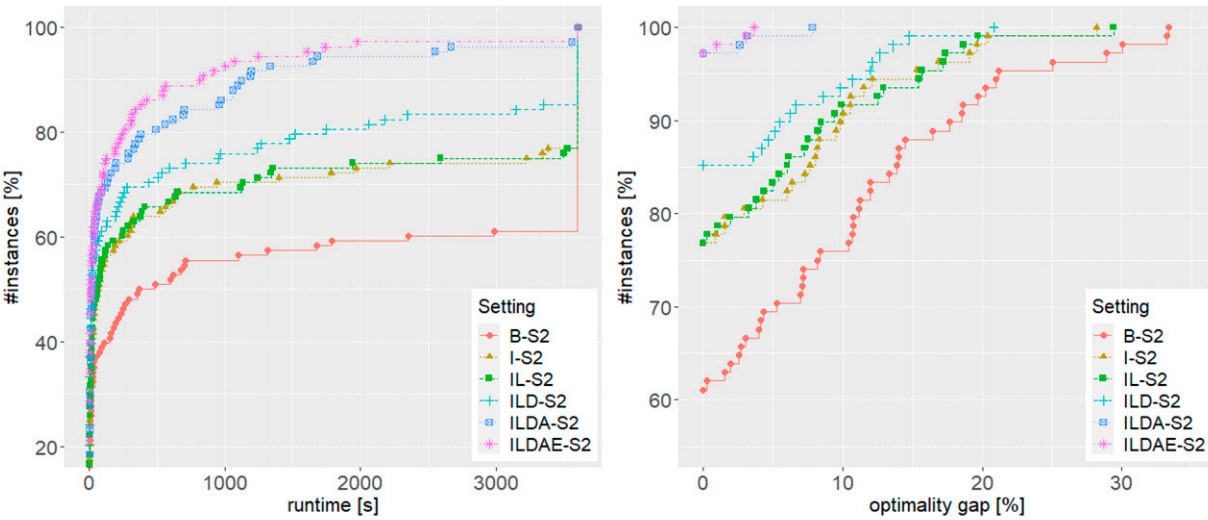


Figure 2. (Color online) WMCIG S2 Results



cuts (I) instead of the basic one (B) causes a significant improvement in terms of running time and final optimality gaps. Whereas the ratio of instances solved to optimality is 57% under B-S1, it is increased to 73% under I-S1. Adding lifted cuts (L) also improves both measures, especially final optimality gaps at the end of the time limit. The next component, dominance inequalities, yields a significant improvement and the ratio of instances solved to optimality becomes 93%. Whereas the addition of alternative cuts to the improved/lifted ones does not make an apparent contribution to the performance, enhanced integer separation decreases the average solution time. The ineffectiveness of alternative cuts can be explained by the ground set definition used for S1, that is, $x_i^* = 0$ for $i \in \hat{S}$, except the cases in which some i with $x_i^* > 0$ are also included to reach a maximal set. This definition usually causes nonpositive $(x_{a_k} - x_{b_k})$ values in the last term of alternative cuts, which results in a smaller violation than the original improved cut.

In Figure 2, we present the results for S2. Here, whereas 61% of the instances are solved to optimality under the basic setting B-S2, the maximum optimality gap is 33%, which is large compared with B-S1. This value remains larger until the alternative cuts are included (setting ILDA-S2), which causes a substantial decrease in running time and final gap unlike option S1. In S2, the ground set for the follower problem is defined based on a rounding scheme. Thus, the heuristic separation procedure for alternative cuts given in Section 4.4 is able to find eligible item pairs with $(x_{a_k} - x_{b_k}) > 0$ more easily, which explains the difference in the effect of component A under S1 and S2. After the addition of the enhanced integer separation component (E), the solution times decrease more and the maximum optimality gap is reduced to 3%. This

Figure 3. (Color online) WMCIG S3 Results

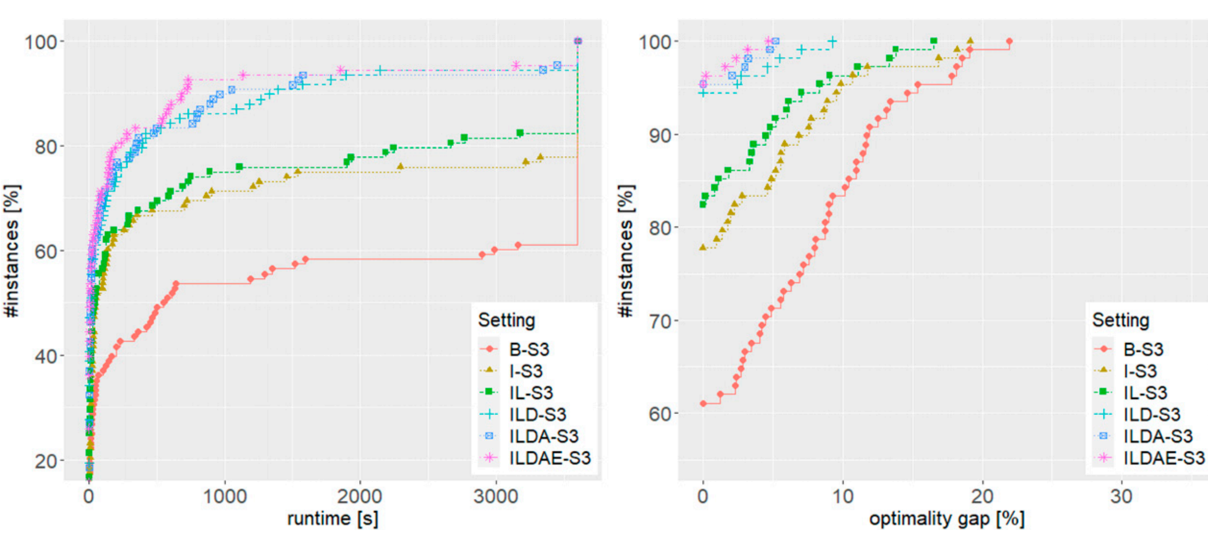


Table 1. Results of WMCIG Experiments with the Complete Settings (ILDAE) of Each Separation Option and with the Benchmark MIBLP Solver MIX++

	Time (seconds)				Gap (%)				rGap (%)				# of nodes			# of SICs		
(n, q, b, r)	S1	S2	S3	MIX_{++}	S1	S2	S3	MIX_{++}	S1	S2	S3	MIX_{++}	S1	S2	S3	S1	S2	S3
(50,5,5,1)	0.7	0.4	0.7	TL	0.0	0.0	0.0	66.7	23.3	14.3	17.3	99.7	84.3	52.7	50.7	169.0	173.7	208.3
(50,5,5,2)	4	1.6	1.1	TL	0.0	0.0	0.0	73.9	18.7	16.1	17.6	100.0	243.0	224.0	204.7	227.7	268.7	329.0
(50,5,5,3)	3.3	1.6	4.1	TL	0.0	0.0	0.0	84.3	14.3	15.0	15.6	99.6	315.7	376.0	329.0	148.7	174.3	194.7
(50,5,10,1)	2.2	1.6	0.8	TL	0.0	0.0	0.0	95.4	36.5	28.9	32.1	100.0	550.3	194.7	238.3	949.0	818.3	1,226.7
(50,5,10,2)	5.2	2.6	3.5	TL	0.0	0.0	0.0	88.9	29.3	27.7	29.6	100.0	1,458.7	684.3	680.3	1,192.0	1,141.3	1,682.0
(50,5,10,3)	2.6	2.0	2.5	TL	0.0	0.0	0.0	80.1	24.1	26.9	24.2	100.0	312.3	434.3	176.0	256.7	412.3	421.0
(60,6,6,1)	1.7	0.9	0.8	TL	0.0	0.0	0.0	95.9	21.7	18.3	20.1	100.0	256.3	108.0	146.0	346.3	348.7	480.3
(60,6,6,2)	40.6	18.6	10.3	TL	0.0	0.0	0.0	89.1	19.9	18.8	21.5	100.0	1,041.7	1,115.7	907.3	640.7	854.0	891.0
(60,6,6,3)	0.7	2.0	0.9	TL	0.0	0.0	0.0	85.1	14.5	20.1	17.7	99.4	614.0	1,407.3	1,065.3	298.0	445.0	429.3
(60,6,12,1)	1.9	1.9	1.1	TL	0.0	0.0	0.0	97.0	32.2	28.0	25.4	100.0	539.0	174.7	180.0	1,211.0	1,025.7	1,098.7
(60,6,12,2)	117.8	44.3	82.2	TL	0.0	0.0	0.0	94.7	27.6	30.9	30.3	100.0	10,259.3	6,057.3	7,097.3	4,216.0	3,781.0	6,813.0
(60,6,12,3)	2.3	8.0	3.9	TL	0.0	0.0	0.0	86.8	22.0	30.1	27.0	100.0	1,160.0	1,986.7	619.3	809.0	1,304.7	1,372.3
(70,7,7,1)	1.7	2.0	1.8	TL	0.0	0.0	0.0	98.7	23.2	20.5	22.4	100.0	746.0	242.0	315.0	895.3	821.3	1,043.0
(70,7,7,2)	187.5	128.0	346.8	TL	0.0	0.0	0.0	95.9	18.3	20.3	19.6	100.0	5,659.7	8,115.3	5,457.0	1,177.0	1,135.0	1,224.3
(70,7,7,3)	2.2	2.7	1.8	TL	0.0	0.0	0.0	85.6	15.3	21.0	17.6	100.0	911.3	2,900.3	1,828.3	425.7	596.0	495.3
(70,7,14,1)	71.3	67.8	120.4	TL	0.0	0.0	0.0	97.0	35.8	32.6	33.0	100.0	5,040.0	1,430.3	2,117.7	7,980.7	6,302.3	9,609.3
(70,7,14,2)	863.3	775.0	1,278.2	TL	0.0	0.0	0.1	99.6	27.4	30.8	30.7	100.0	47,759.0	24,592.0	25,209.7	12,366.0	12,365.7	15,427.7
(70,7,14,3)	43.2	13.9	7.2	TL	0.0	0.0	0.0	88.1	24.7	30.7	27.4	100.0	2,506.3	2,708.3	804.0	1,629.0	1,607.0	1,901.7
(80,8,8,1)	10	4.8	4.7	TL	0.0	0.0	0.0	99.4	21.7	20.3	22.4	100.0	980.0	545.0	617.0	1,367.7	1,207.7	1,724.0
(80,8,8,2)	223.4	200.3	183.2	TL	0.0	0.0	0.0	99.8	17.8	20.2	20.2	100.0	11,962.3	20,627.0	11,207.3	2,436.7	2,775.7	2,721.7
(80,8,8,3)	2.6	3.8	6.1	TL	0.0	0.0	0.0	99.9	16.1	24.9	19.8	100.0	995.7	3,898.3	2,453.0	470.7	692.7	667.7
(80,8,16,1)	201.6	64.9	108.7	TL	0.0	0.0	0.0	99.7	35.0	36.9	35.8	100.0	8,700.7	1,619.0	2,416.7	13,614.0	7,757.3	12,632.7
(80,8,16,2)	1,860.5	2,007.6	1,997.8	TL	1.3	0.3	0.5	99.6	26.9	30.9	29.6	100.0	74,498.0	53,411.3	29,401.0	19,615.0	20,274.7	26,901.3
(80,8,16,3)	4.8	9.4	4.0	TL	0.0	0.0	0.0	100.0	26.2	31.3	29.8	100.0	1,213.0	2,961.0	694.3	1,090.3	1,577.0	1,229.0
(90,9,9,1)	62	25.4	37.5	TL	0.0	0.0	0.0	97.4	20.0	21.1	21.6	100.0	3,445.0	890.7	1,525.7	4,741.0	2,806.7	3,768.7
(90,9,9,2)	697.6	815.8	613.3	TL	0.0	0.0	0.0	95.1	17.8	18.8	18.5	100.0	54,221.0	52,596.0	30,823.0	5,261.3	4,480.0	5,431.3
(90,9,9,3)	5.2	9.0	2.4	TL	0.0	0.0	0.0	83.0	18.2	23.8	20.5	100.0	373.7	5,118.7	914.3	651.7	908.7	901.0
(90,9,18,1)	966.8	285.0	473.9	TL	0.0	0.0	0.0	99.3	33.4	34.8	36.3	100.0	21,404.3	3,748.0	4,431.7	26,092.3	14,426.3	22,072.7
(90,9,18,2)	3,214.1	1,687.9	1,768.8	TL	2.9	1.2	1.1	99.6	26.0	30.6	29.7	100.0	57,842.7	27,101.3	18,752.0	24,595.3	19,133.0	22,814.0
(90,9,18,3)	12.3	48.3	10.7	TL	0.0	0.0	0.0	99.9	26.4	32.0	29.4	100.0	1,668.0	9,765.7	1,300.7	1,485.3	2,788.0	2,380.0
(100,10,10,1)	87.3	26.7	41.4	TL	0.0	0.0	0.0	98.7	23.4	23.3	22.3	100.0	8,410.0	1,277.0	1,799.7	7,730.3	3,089.3	4,975.3
(100,10,10,2)	285.5	281.4	321.6	TL	0.0	0.0	0.0	97.3	17.5	20.9	19.8	100.0	24,383.7	30,392.0	12,767.0	5,350.0	4,709.3	6,533.0
(100,10,10,3)	7.8	17.7	15.4	TL	0.0	0.0	0.0	94.5	18.3	26.4	21.9	100.0	909.0	10,693.7	3,846.3	1,018.3	1,462.3	1,345.3
(100,10,20,1)	TL	1,950.1	3,448.1	TL	7.3	1.0	2.4	99.6	34.9	34.1	35.7	100.0	22,439.7	5,621.7	9,238.0	59,397.0	34,759.3	52,794.3
(100,10,20,2)	1,354.5	737.2	367.7	TL	0.0	0.0	0.0	100.0	25.8	31.9	29.6	100.0	38,421.0	18,487.3	5,056.3	11,156.7	7,613.3	8,312.7
(100,10,20,3)	72.9	314.1	73.1	TL	0.0	0.0	0.0	99.6	29.1	35.4	31.5	99.8	8,242.7	34,327.0	3,764.7	3,279.7	5,546.0	4,851.0
Average	389.5	265.7	315.2	TL	0.3	0.1	0.1	93.5	24.0	25.8	25.1	100.0	11,654.6	9,330.1	5,234.3	6,230.3	4,710.6	6,302.9

Notes. The results are aggregated over the three instances with the same n, q, b, and r values, and given as averages. TL indicates that the time limit of 3,600 seconds is reached for all instances involved in the average.

Table 2. Average Number of Cuts and Average Time Spent for Cut Generation Under Setting ILDAE for WMCIG Instances

	Number of cuts				Time (seconds)			
	#SIC	#L	#A	#D	t_{Pre}	t_{SEP}	t_L	t_A
S1	6,230.3	5,089.5	0.1	76.2	<0.1	68.1	0.1	0.1
S2	4,710.6	112.5	4,413.7	76.2	<0.1	48.1	<0.1	1.5
S3	6,302.9	1,387.9	4,437.4	76.2	<0.1	49.7	0.1	0.6

result shows that a better search can be done when the time due to solving separation problems to optimality is saved.

The last option S3, whose results are plotted in Figure 3, yields a maximum gap of 20% under the basic setting B-S3, which is notably smaller compared with S1 and S2, although the optimal solution ratio is similar to those of the previous options. On the other hand, I-S3, IL-S3, and ILD-S3 yield better running time and final gaps than their counterparts under S1 and S2. Since alternative cuts yield a slight performance improvement compared with S2, S3 falls barely behind of S2 in the complete setting ILDAE-S3, with a maximum gap of 5%.

In Table 1, the results of the complete (ILDAE) settings of all three fractional separation options are presented in terms of running time in seconds, final gaps, root gaps, number of branch-and-cut tree nodes, and number of SICs generated. The first three measures are also compared with those obtained with the state-of-the-art MIBLP solver, using its default setting MIX++ (Fischetti et al. 2017). The MIBLP formulation of WMCIG is provided in the online supplement, and the MIBLP solver is publicly available at https://msinnl.github.io/pages/bilevel.html. The numbers in the table show averages over three instances with the same parameter setting. We see that, whereas MIX++ is not able to solve any of the instances within the time limit of one hour and yields an average gap of 93.5%, this value is below 1% with our settings. Even the minimum final gap obtained with MIX++, which is not reported in the table, is 54%. The difference between root gaps is also notable. The average root gap is almost 100% with MIX++ as opposed to 25%, which is the average under S1, S2, and S3. When we focus only on our settings, we see that ILDAE-S2 is the best performing setting in terms of solution time, whereas it yields slightly larger root gaps than the others. The average tree size is considerably smaller under S3, and S2 requires the smallest number of cuts.

In the first half of Table 2, we present the average number of SICs generated, the number of lifted (#L) and alternative cuts (#A), and the number of dominance inequalities (#D) added to the initial model, under the complete setting ILDAE for each separation option. We observe that a significant number of cuts of each type are added, except alternative cuts under S1, which we discussed earlier. Recall that alternative cuts are added in addition to the original improved/lifted cut. Thus, the number of improved cuts that are not lifted can be obtained by #SIC–#A–#L. Also note that, #L (#A) excludes the number of lifted (alternative) cuts that are obtained but not added to the model because they are not violated by the current node solution. The average total times spent in the cut-generation procedures of a single instance are shown in the second half of Table 2. The preprocessing time includes the time to obtain dominance inequalities and the dominating lists, which are used for lifted cut generation as described in Section 4.4, which is clearly negligible. SEP denotes the time to obtain all $\hat{S}$ for which SICs are generated, and it is the most time-consuming step of separation, as expected. Both lifted

Figure 4. (Color online) BIIG S1 Results

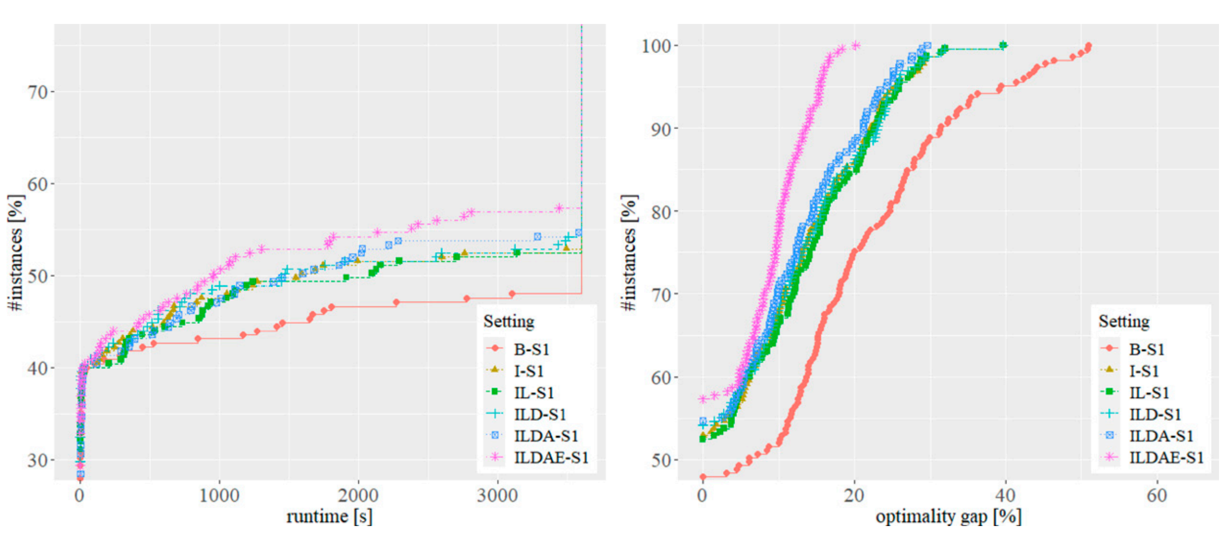
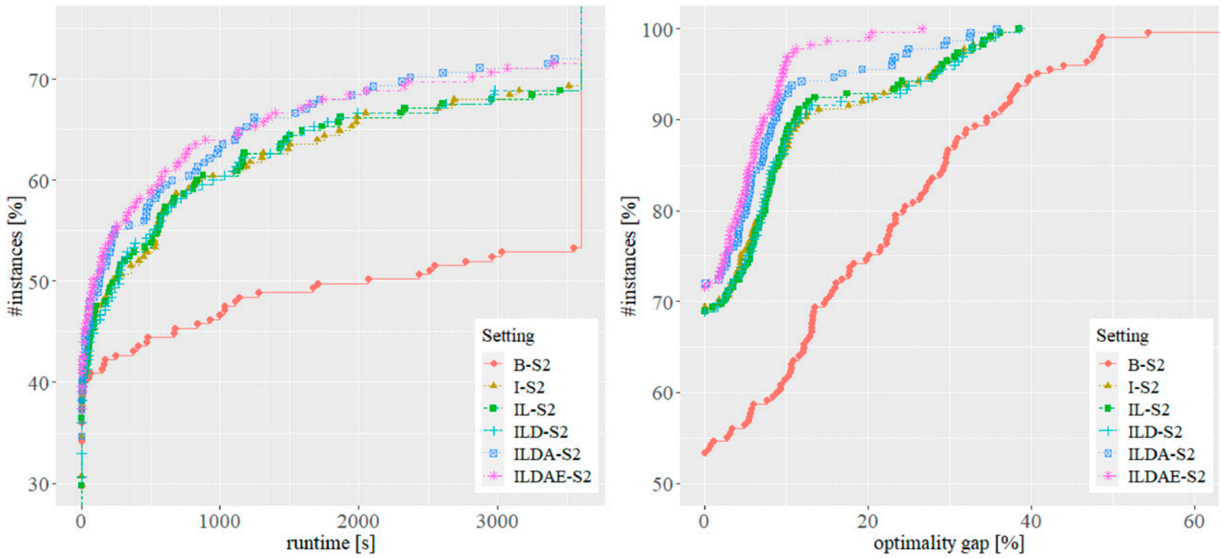


Figure 5. (Color online) BIIG S2 Results



and alternative cut-generation heuristics are time efficient when the number of cuts given on the left is taken into account. The detailed results of all instances are given in the online supplement.

5.2. Bipartite Inference Interdiction Game

While generating the BIIG instances, we adopt the parameter settings used in Salvagnin (2019) for the bipartite inference problem, which constitutes the lower level of BIIG. We do not include the parameter values that lead to failing to solve the problem within one hour according to their results, as we have an additional problem layer. As a result, the instances are generated as follows. The activating probability p_i is sampled uniformly in $[0, 1]$ for each $i \in N$. For the density d of the graphs, that is, the probability of having an arc between each (i, j) pair, in addition to 0.07, which is the only value used in Salvagnin (2019), two more values $\{0.1, 0.15\}$ are determined, and the arcs are generated in a completely random manner. The number of items $n \in \{20, 50, 100\}$, the number of targets $m \in \{2n, 5n, 10n\}$, and number of items q that the follower can choose is in {5, 10} for $n = 20$, {10, 20} for $n = 50$, and equal to 10 for $n = 100$. The leader can interdict $b = 5$ items if $n = 20$ and 10 items if $n > 20$. Five distinct instances are generated for each parameter setting. Note that while solving the separation problem of BIIG, we make use of the greedy fractional separation proposed in Salvagnin (2019).

The results of the experiments in terms of running time and final gaps are plotted in Figures 4, 5, and 6.

Figure 6. (Color online) BIIG S3 Results

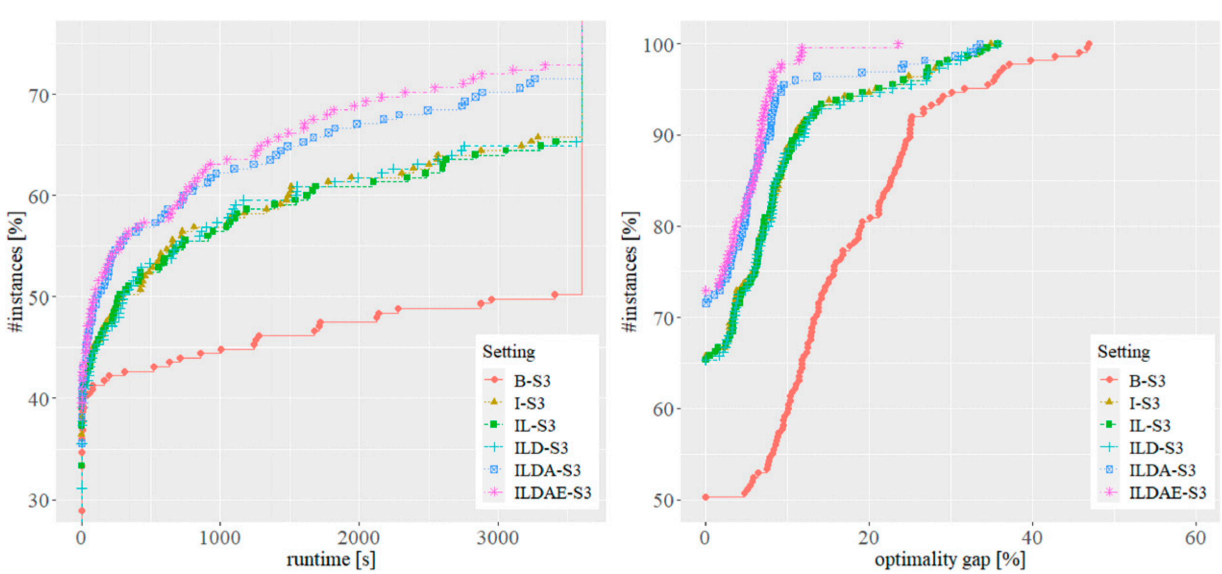


Table 3. Results of BIIG Experiments with the Complete Settings (ILDAE) of Each Separation Option

	Time (seconds)			Gap (%)			rGap (%)			# of nodes			# of SICs		
(n,m,q,b,d)	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3
(20,40,5,5,0.07)	0.1	0.2	0.1	0.0	0.0	0.0	40.8	25.8	25.0	152.2	63.4	71.0	185.8	208.0	253.4
(20,40,5,5,0.10)	0.1	0.6	0.1	0.0	0.0	0.0	38.2	22.9	23.1	92.6	49.8	42.8	142.0	151.6	177.2
(20,40,5,5,0.15)	0.1	0.4	0.1	0.0	0.0	0.0	37.9	27.0	24.1	137.8	53.4	49.4	198.8	197.4	253.2
(20,40,10,5,0.07)	0.1	0.2	0.1	0.0	0.0	0.0	26.4	25.0	10.2	101.2	38.2	12.8	108.0	135.4	77.8
(20,40,10,5,0.10)	0.8	0.1	0.2	0.0	0.0	0.0	27.4	26.2	16.5	174.4	63.6	28.4	181.8	194.0	135.4
(20,40,10,5,0.15)	0.4	0.1	0.2	0.0	0.0	0.0	24.2	25.5	21.4	302.2	74.4	46.8	378.0	252.0	252.8
(20,100,5,5,0.07)	2.1	0.2	0.3	0.0	0.0	0.0	37.5	23.0	23.0	93.4	32.8	43.8	152.6	144.2	185.2
(20,100,5,5,0.10)	1.5	0.2	0.2	0.0	0.0	0.0	39.9	27.7	25.0	144.0	57.6	63.2	192.6	221.2	253.8
(20,100,5,5,0.15)	1.8	0.1	0.2	0.0	0.0	0.0	35.2	24.2	25.5	118.4	39.0	45.4	202.0	204.4	272.8
(20,100,10,5,0.07)	4.2	0.6	0.4	0.0	0.0	0.0	27.3	26.6	16.8	278.0	78.0	25.6	289.8	283.2	145.4
(20,100,10,5,0.10)	5.8	0.7	0.5	0.0	0.0	0.0	28.1	27.3	20.7	620.4	168.0	93.6	667.0	589.4	500.0
(20,100,10,5,0.15)	4.6	0.2	0.2	0.0	0.0	0.0	28.6	29.4	24.3	497.8	140.2	71.0	582.2	471.4	432.4
(20,200,5,5,0.07)	0.6	0.1	0.1	0.0	0.0	0.0	38.6	23.2	23.2	116.4	36.6	45.0	167.4	143.0	172.4
(20,200,5,5,0.10)	0.4	0.1	0.1	0.0	0.0	0.0	39.8	24.1	22.9	122.2	57.4	61.6	200.6	221.8	272.4
(20,200,5,5,0.15)	0.9	0.2	0.1	0.0	0.0	0.0	40.4	27.6	27.9	174.4	55.6	67.6	287.0	279.6	340.4
(20,200,10,5,0.07)	2.5	0.2	0.2	0.0	0.0	0.0	25.8	24.2	15.5	273.4	67.8	32.8	295.2	252.0	179.0
(20,200,10,5,0.10)	5.0	0.5	0.6	0.0	0.0	0.0	29.6	29.0	20.8	642.6	169.4	116.0	852.6	615.2	586.4
(20,200,10,5,0.15)	5.0	1.2	1.4	0.0	0.0	0.0	29.0	28.6	27.1	1,242.4	361.2	206.6	1,509.4	1,062.8	1,127.6
(50,100,10,10,0.07)	497.3	31.6	56.7	0.0	0.0	0.0	40.0	33.2	33.7	12,609.0	1,003.6	974.2	18,981.6	5,642.4	7,243.0
(50,100,10,10,0.10)	461.0	32.4	38.4	0.0	0.0	0.0	40.4	35.6	34.1	10,727.2	1,175.6	1,067.8	19,606.2	6,663.8	8,034.0
(50,100,10,10,0.15)	1,766.8	223.5	390.7	0.5	0.0	0.0	38.3	36.6	36.0	17,883.8	3,337.6	2,482.4	31,304.8	13,407.2	17,467.8
(50,100,20,10,0.07)	3,156.8	161.6	59.9	5.6	0.0	0.0	26.6	26.6	24.8	20,414.4	2,898.8	1,017.6	33,435.8	14,403.2	7,721.8
(50,100,20,10,0.10)	TL	3,494.3	2,634.8	12.6	2.4	1.2	29.1	27.7	27.9	16,025.2	13,164.2	6,371.6	37,761.8	60,292.8	50,837.8
(50,100,20,10,0.15)	TL	3,208.4	2,963.0	11.5	4.5	4.0	28.4	29.0	28.2	15,514.4	13,250.2	5,929.4	41,684.2	54,774.2	50,338.2
(50,250,10,10,0.07)	1,217.3	134.9	183.3	0.0	0.0	0.0	38.9	34.2	34.4	17,704.4	1,702.6	1,743.4	33,834.2	11,320.8	14,456.6
(50,250,10,10,0.10)	1,726.4	415.4	419.8	0.7	0.0	0.0	38.4	36.3	35.1	19,732.6	3,144.2	2,718.6	39,226.0	17,692.8	23,349.8
(50,250,10,10,0.15)	3,177.3	449.7	574.9	1.6	0.0	0.0	40.4	38.4	38.0	24,927.8	4,618.4	3,416.4	49,746.6	24,966.8	30,909.0
(50,250,20,10,0.07)	TL	2,531.9	1,682.5	11.8	2.2	1.4	29.3	28.7	27.9	19,511.0	8,230.6	3,461.4	43,228.4	49,402.0	33,842.6
(50,250,20,10,0.10)	TL	TL	TL	15.0	7.1	5.8	31.1	29.4	29.9	18,926.4	7,611.8	4,752.6	45,617.8	56,344.0	58,479.8
(50,250,20,10,0.15)	TL	TL	TL	13.9	6.3	6.6	29.4	29.5	28.2	14,680.6	11,602.6	5,102.2	45,123.0	61,814.2	57,854.0
(50,500,10,10,0.07)	1,439.4	150.6	238.5	1.0	0.0	0.0	41.8	36.3	34.7	18,658.8	1,909.6	2,095.6	37,891.8	12,595.4	17,501.2
(50,500,10,10,0.10)	2,273.7	1,363.2	1,569.0	3.8	0.7	0.9	40.2	36.8	37.4	18,907.8	4,431.8	3,693.2	51,974.8	31,283.0	37,255.6
(50,500,10,10,0.15)	2,274.1	728.5	1,003.0	1.4	0.0	0.0	38.9	39.3	38.0	23,072.4	4,885.0	4,070.2	50,126.6	27,064.2	37,763.6
(50,500,20,10,0.07)	TL	2,586.8	2,459.2	11.6	2.9	2.5	28.7	28.3	27.6	21,892.0	5,827.0	3,893.4	52,372.8	44,324.6	44,379.0
(50,500,20,10,0.10)	TL	TL	TL	16.3	9.1	7.9	30.5	30.1	29.7	19,493.2	6,110.8	4,409.8	54,661.0	48,570.2	58,183.2
(50,500,20,10,0.15)	TL	TL	TL	14.8	9.5	9.5	30.1	29.8	30.1	13,530.0	8,170.0	4,968.2	47,696.0	52,888.4	59,810.2
(100,200,10,10,0.07)	3,072.4	1,043.6	1,356.8	4.5	0.0	0.0	31.7	30.0	30.9	21,213.0	5,626.4	6,024.4	61,647.6	25,075.4	35,530.0
(100,200,10,10,0.10)	3,402.8	1,730.9	1,880.5	5.2	1.1	1.4	30.1	32.0	31.5	24,544.2	8,748.2	6,887.6	63,576.4	31,085.4	40,010.2
(100,200,10,10,0.15)	TL	TL	3,345.8	10.0	6.4	4.1	28.6	32.6	30.9	9,591.6	8,634.2	4,623.4	33,710.4	23,607.6	37,860.6
(100,500,10,10,0.07)	TL	2,250.5	2,819.6	7.4	0.8	1.4	32.6	31.1	31.7	15,817.0	5,795.8	6,365.8	65,104.0	41,323.4	58,243.6
(100,500,10,10,0.10)	TL	TL	TL	9.8	4.8	4.8	30.4	32.5	32.1	14,287.8	5,808.0	5,947.8	59,755.6	41,601.8	62,433.0
(100,500,10,10,0.15)	TL	TL	TL	14.7	11.3	8.8	28.4	33.5	32.2	4,123.0	6,051.6	4,114.2	21,075.2	24,076.8	43,642.0
(100,1000,10,10,0.07)	TL	3,000.8	3,058.1	7.5	2.9	3.7	31.5	33.1	32.5	16,490.6	4,858.2	5,463.4	70,259.4	47,598.6	64,524.6
(100,1000,10,10,0.10)	TL	3,188.7	TL	8.7	2.4	4.3	31.2	34.7	33.7	12,813.2	6,811.4	5,014.2	64,218.8	53,052.8	61,764.4
(100,1000,10,10,0.15)	TL	TL	TL	14.8	13.0	7.3	29.6	35.9	33.6	3,755.2	4,212.6	3,366.2	23,333.4	22,955.8	43,749.6
Average	1,744.5	1,234.1	1,234.2	4.5	1.9	1.7	33.1	30.0	28.0	10,047.3	3,582.8	2,468.9	26,745.5	20,210.1	23,751.2

Notes. The results are aggregated over the five instances with the same n, m, B, k, and d values, and given as averages. TL indicates that the time limit of 3,600 seconds is reached for all instances involved in the average.

Table 4. Average Number of Cuts and Average Time Spent for Cut Generation Under Setting ILDAE for BIIG Instances

	Number of cuts				Time (seconds)			
	#SIC	#L	#A	#D	t_{Pre}	t_{SEP}	t_L	t_A
S1	26,745.5	9.1	13.3	1.1	<0.1	281.9	0.2	0.1
S2	20,210.1	0.9	1,6005.5	1.1	<0.1	297.6	0.1	5.6
S3	23,751.2	0.1	15,402.2	1.1	<0.1	219.4	0.1	4.2

Common to all three separation options, the improved cuts (I) make the largest contribution for both measures. The I, IL, and ILD settings perform similarly. We observe that the number of lifted cuts and dominance inequalities added are small (see Table 4), which leads to different branch-and-cut trees but not notable improvement in performance. We attribute this situation to the rareness of item pairs suitable for use in these cuts, due to the structure of the instances, that is, it is difficult to have that one item covers all the targets that another item covers and has a larger activation probability as described in Remark 3 and Proposition 4. Under S1 (see Figure 4), the alternative cuts cause slightly smaller solution times and the maximum gap reduces from 40% to 30%. Including component E further decreases this number to 20%.

The plots for S2 are shown in Figure 5. As is the case with WMCIG instances, B-S2 setting yields large final optimality gaps. The increase in the number of instances solved to optimality due to the improved SICs is larger than with S1, as can be seen from both plots. Alternative cuts have a larger contribution than they have under S1. With all the components (ILDAE-S2), the maximum optimality gap is 26% and the 72% of instances are solved to optimality within the time limit. The performance of S3 is similar to that of S2, except the optimality gaps under the basic setting, which are better in the former.

Next, in Table 3 we present the average results for the ILDAE setting under all three options, with a similar structure as used in Table 1, except the MIX++ columns, since BIIG does not fit into the MIX++ setting, due to not having a compact MIBLP formulation. Each number denotes the average over five instances. In terms of running time and final gaps, while S2 and S3 perform similarly, S1 falls behind. S3 outperforms the others in terms of root gaps. As before, the tree size is smallest under S3. Since the number of instances solved to optimality is larger with this setting, it is understood that with S3 the optimal solution is reached in a smaller number of subproblems. Finally, the number of SICs generated is smaller when using S2. The number of cuts of each type and the average time spent for preprocessing, solving the separation problem (greedy or via solving SEP), cut lifting, and alternative cut generation are shown in Table 4. The time spent for getting lifted/alternative cuts is also small under each separation option. The detailed results for two instances from each class shown in Table 3 are provided in the online supplement.

5.3. Budgeted Extensions of WMCIG

Most of the applications of constrained submodular maximization problem involve cardinality constraints on the number of selected items, such as the uncapacitated location problem (Nemhauser et al. 1978), influence maximization problem Kempe et al. (2003), submodular observation selection problem (Krause

Figure 7. (Color online) WMCIG-L S1 Results

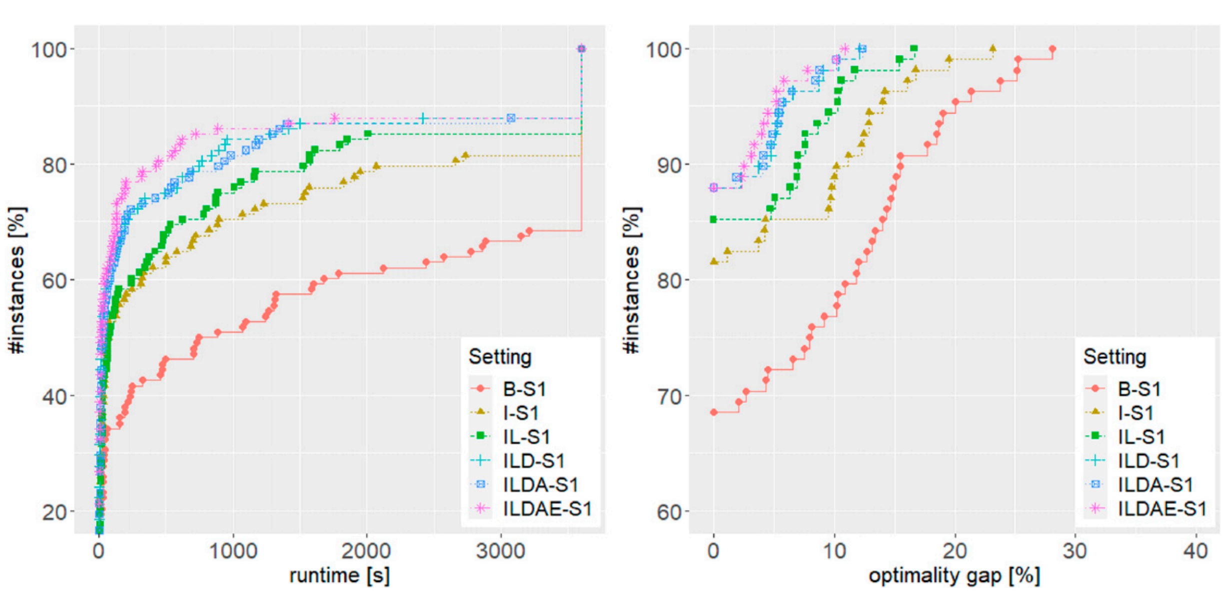
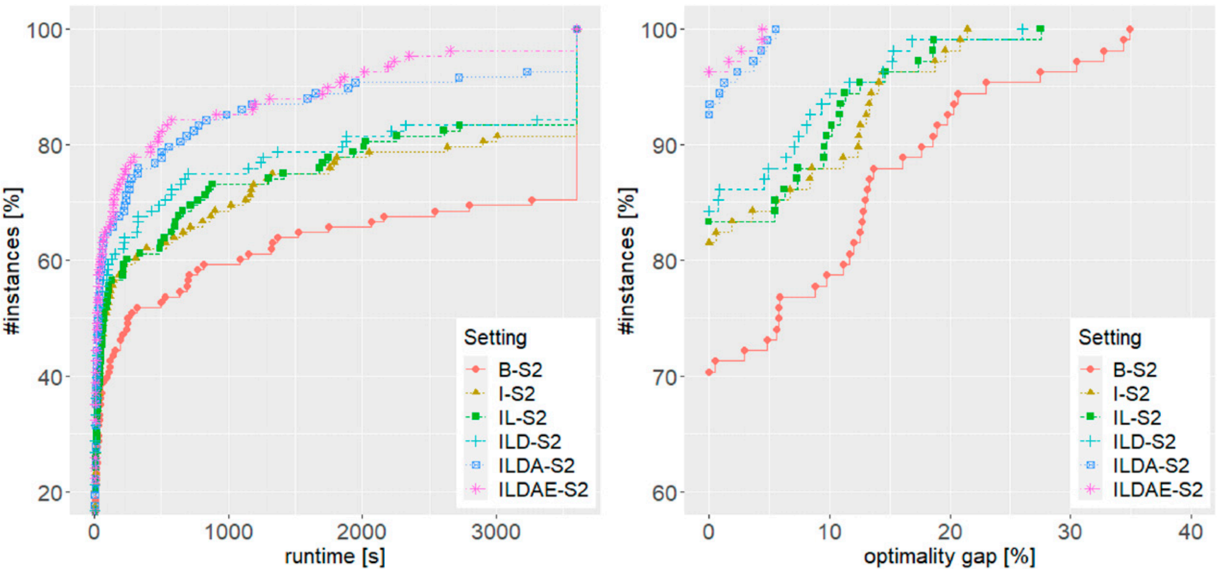


Figure 8. (Color online) WMCIG-L S2 Results



et al. 2008), and the maximum coverage problem, for which we considered the WMCIG in the first part of our computational study. Moreover, many works on interdiction games in the literature focus on cardinality-constrained interdiction in their computational studies, for example, Tang et al. (2016), Lozano and Smith (2017a), Furini et al. (2021). However, our solution approach can also handle budget constraints. Thus, in this section, we consider budgeted extensions of WMCIG where the leader and/or the follower have budget constraints instead of cardinality constraints.

The first extension of WMCIG involves a leader budget constraint $\sum_i g_i \leq b$, where g_i denotes the cost of interdicting facility location i. We refer to this problem as WMCIG-L. We consider the same instance-generation scheme described in Section 5.1 except that we set an interdiction budget $b \in \{\sum_i g_i/5, \sum_i g_i/10\}$ and sample each g_i independently from the integers in [1, 10]. The results are shown in Figures 7, 8, and 9, in terms of runtimes and optimality gaps at the end of one hour. The results are similar to those obtained for WMCIG, except the contribution of dominance inequalities is smaller here, which could be expected because the interdiction costs should also be taken into account as described in Theorem 5. As before, we report the average number of cuts and time spent for several algorithm components in Table 5.

Figure 9. (Color online) WMCIG-L S3 Results

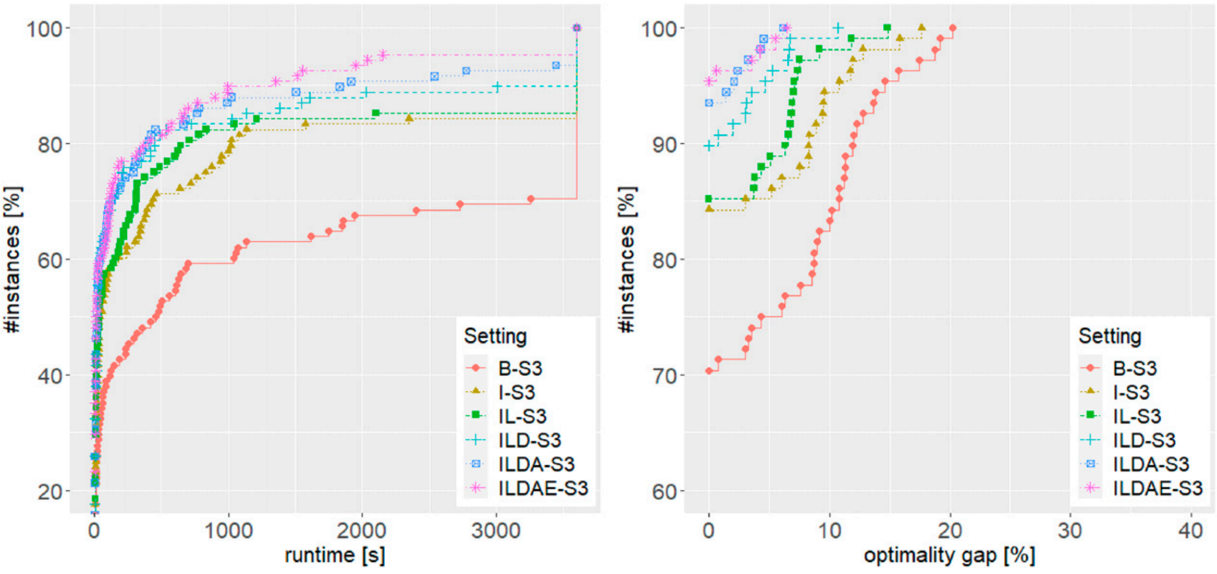


Table 5. Average Number of Cuts and Average Time Spent for Cut Generation Under Setting ILDAE for WMCIG-L Instances

	Number of cuts				Time (seconds)			
	#SIC	#L	#A	#D	t_{Pre}	t_{SEP}	t_L	t_A
S1	8,131.4	5,904.6	0.1	39.1	< 0.1	62.3	0.1	0.1
S2	5,539.7	144.7	5,127.4	39.1	< 0.1	65.5	0.1	3.0
S3	7,550.7	1,695.0	5,184.0	39.1	< 0.1	52.2	0.1	0.8

Table 6. Average Number of Cuts and Average Time Spent for Cut Generation Under Setting ILDAE for WMCIG-LF Instances

	Number of cuts				Time (seconds)			
	#SIC	#L	#A	#D	t_{Pre}	t_{SEP}	t_L	t_A
S1	15,479.3	5,149.7	2,599.4	23.8	< 0.1	11.6	0.2	0.2
S2	18,245.5	877.8	12,227.8	23.8	< 0.1	21.8	0.2	2.0
S3	18,315.3	1,418.4	9,373.6	23.8	< 0.1	25.0	0.2	1.4

In the second extension to WMCIG, we consider budget constraints for both the leader and the follower, and we refer to the problem as WMCIG-LF. The follower budget constraint $c(S) = \sum_{i \in N} c_i \le q$ replaces the cardinality constraint in WMCIG. For the experiments, we use the instances of WMCIG-L and add the follower budget constraint. This is done by sampling c_i values independently in $[1, 10]$ and setting the budget $q = \sum_i c_i / 10$. The results are shown in Figures 10, 11, and 12. It is clear that the most effective algorithmic component is the improved cut. The average number of cuts and time spent for algorithm components are given in Table 6. Compared with the results obtained for the other problem, the relative number of lifted cuts is much less. This could be explained by the fact that due to the follower budget constraint, there are less item pairs that fulfill the conditions for lifting.

6. Conclusion

In this paper, we have presented an exact method to solve interdiction games with a submodular and monotone objective function. Such problems have many real-world applications, as described in Section 1.1. We introduce submodular interdiction cuts (SIC) by exploiting the special properties of submodular set functions. We also develop improved and lifted variants of these SICs. The branch-and-cut framework that we design based on SICs involves several other components, such as dominance inequalities, greedy algorithms for separation of fractional solutions, and an enhanced separation procedure for integer solutions. We also investigate the impact of using maximal sets while building SICs instead of nonmaximal ones, and use the obtained information to design better separation schemes.

To assess the performance of our solution algorithm and its individual components, we conduct a computational study on the weighted maximal covering interdiction game and the bipartite inference interdiction game. The results show that the components of our framework provide significant improvements with respect to the basic version. Moreover, our method vastly outperforms a state-of-the-art general-purpose mixed-integer bilevel linear programming (MIBLP) solver for the weighted maximal covering interdiction game (for which a MIBLP formulation is possible).

Figure 10. (Color online) WMCIG-LF S1 Results

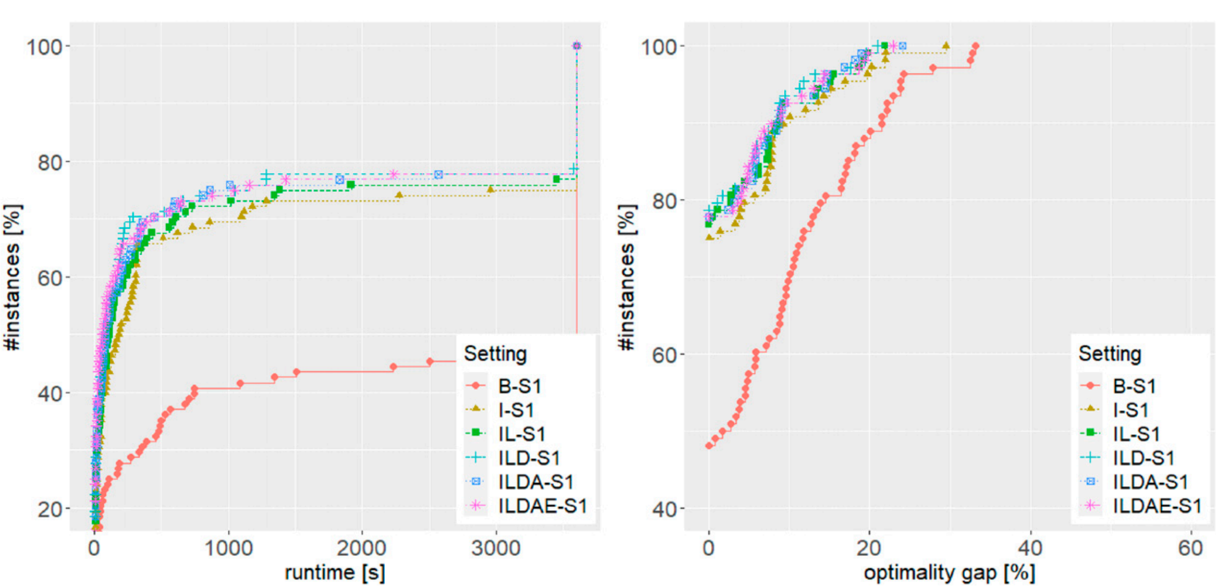
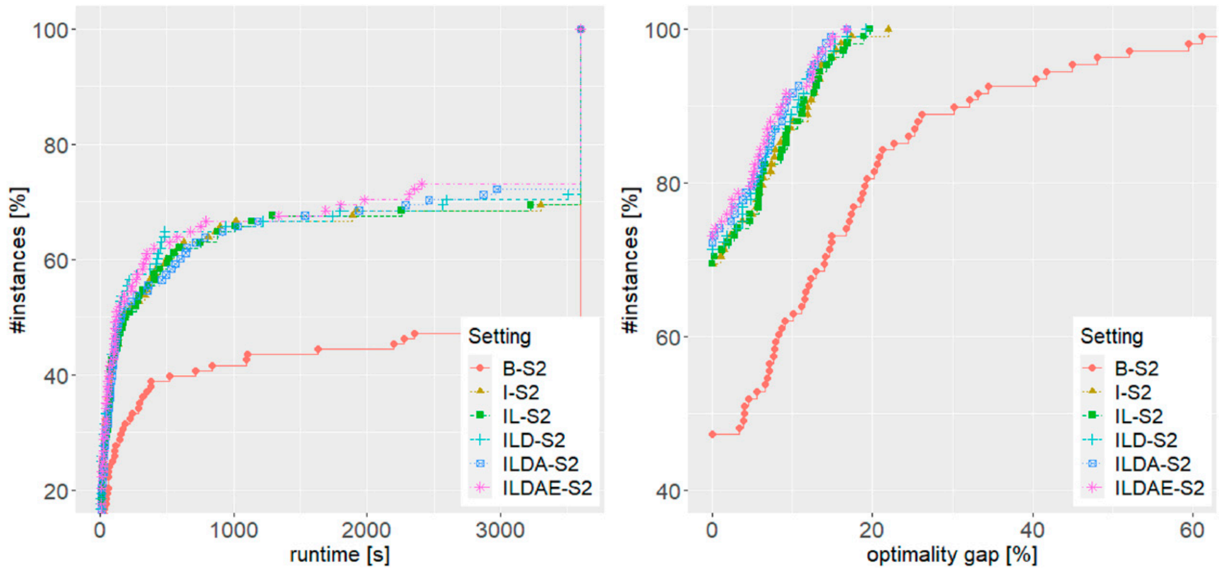


Figure 11. (Color online) WMCIG-LF S2 Results

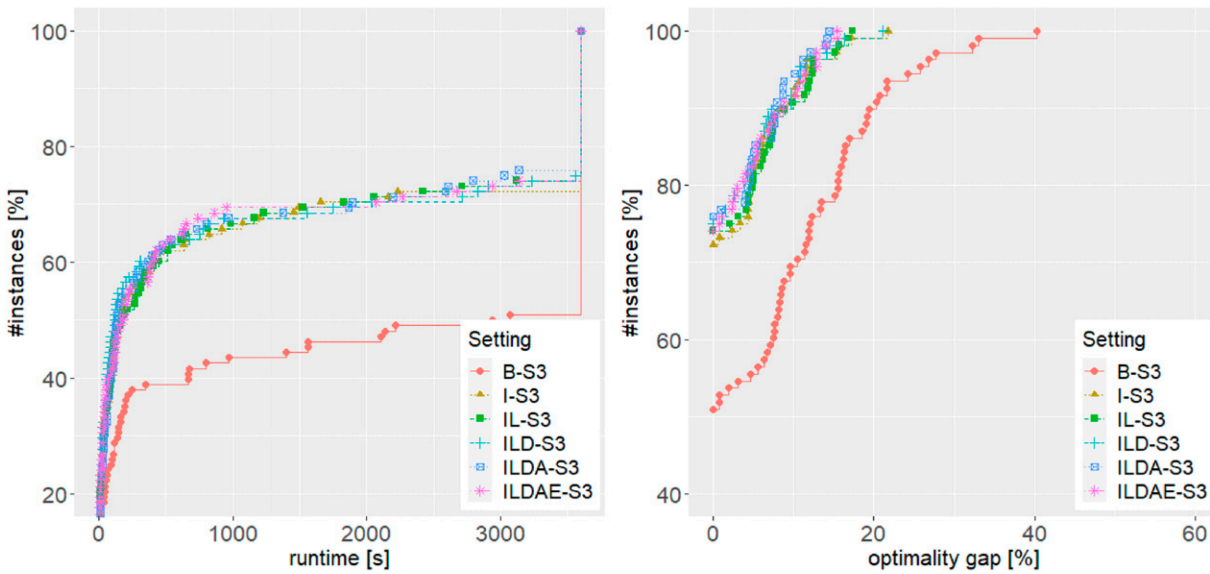


Regarding further work, a natural extension of interdiction games is the fortification problem, where a third problem layer includes the interdiction game as a constraint. Several studies address defender-attacker-defender games, such as Cappanera and Scaparra (2011), Lozano and Smith (2017a), Lozano et al. (2017), and Zheng and Albert (2018). It could be interesting to study such games with submodular objective function. Another possible future research direction is developing methods for the solution of stochastic or robust submodular interdiction games. Finally, one could also focus on concrete submodular interdiction games and try to extend our general-purpose framework with problem-specific components.

Acknowledgments

The authors thank Markus Leitner, Ivana Ljubić, and Michele Monaci for initial discussions on the topic. The authors also thank the anonymous reviewers and the editors for their valuable comments and suggestions. For the

Figure 12. (Color online) WMCIG-LF S3 Results



purpose of open access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

References

Ahmed S, Atamtürk A (2011) Maximizing a class of submodular utility functions. *Math. Programming* 128(1–2):149–169.

Aksen D, Akca SŞ, Aras N (2014) A bilevel partial interdiction problem with capacitated facilities and demand outsourcing. *Comput. Oper. Res.* 41:346–358.

Aksen D, Piyade N, Aras N (2010) The budget constrained r-interdiction median problem with capacity expansion. *Central Eur. J. Oper. Res.* 18(3):269–291.

Alon N, Gamzu I, Tennenholtz M (2012) Optimizing budget allocation among channels and influencers. Mille A, Gandon F, Misselis J, Rabinovich M, Staab S, eds. *Proc. 21st Internat. Conf. World Wide Web* (Association for Computing Machinery, New York), 381–388.

Bayrak H, Bailey MD (2008) Shortest path network interdiction with asymmetric information. *Networks* 52(3):133–140.

Ben-Ayed O, Blair CE (1990) Computational difficulties of bilevel linear programming. *Oper. Res.* 38(3):556–560.

Bogunovic I, Mitrović S, Scarlett J, Cevher V (2017) Robust submodular maximization: A non-uniform partitioning approach. Precup D, Teh YW, eds. *Internat. Conf. Machine Learn.* (PMLR), 508–516.

Bonnet C, Simioni M (2001) Assessing consumer response to protected designation of origin labelling: A mixed multinomial logit approach. *Eur. Rev. Agricultural Econom.* 28(4):433–449.

Borrero JS, Lozano L (2021) Modeling defender-attacker problems as robust linear programs with mixed-integer uncertainty sets. *INFORMS J. Comput.* 33(4):1570–1589.

Brotcorne L, Hanafi S, Mansi R (2013) One-level reformulation of the bilevel knapsack problem using dynamic programming. *Discrete Optim.* 10(1):1–10.

Brown G, Carlyle M, Salmerón J, Wood K (2006) Defending critical infrastructure. *Interfaces* 36(6):530–544.

Calinescu G, Chekuri C, Pál M, Vondrák J (2007) Maximizing a submodular set function subject to a matroid constraint. Fischetti M, Williamson DP, eds. *Internat. Conf. Integer Programming Combin. Optim.* (Springer, Berlin, Heidelberg, Germany), 182–196.

Cappanera P, Scaparra MP (2011) Optimal allocation of protective resources in shortest-path networks. *Transportation Sci.* 45(1): 64–80.

Caprara A, Carvalho M, Lodi A, Woeginger GJ (2016) Bilevel knapsack with interdiction constraints. *INFORMS J. Comput.* 28(2): 319–333.

Church R, ReVelle C (1974) The maximal covering location problem. *Papers Regional Sci. Assoc.* 32:101–118.

Church RL, Scaparra MP, Middleton RS (2004) Identifying critical infrastructure: The median and covering facility interdiction problems. *Ann. Assoc. Amer. Geographers* 94(3):491–502.

Cochran JJ, Cox LA, Keskinocak P, Kharoufeh JP, Smith JC, Wood RK (2011) Bilevel network interdiction models: Formulations and solutions. *Wiley Encyclopedia of Operations Research and Management Science* (Wiley, Hoboken, NJ).

Coniglio S, Furini F, Ljubic I (2020) Maximizing submodular utility functions combined with a set-union operator over a discrete set Working paper, Mathematical Sciences, University of Southampton, England. http://www.optimization-online.org/DB_HTML/2020/04/7715.html.

Contardo C, Sefair JA (2021) A progressive approximation approach for the exact solution of sparse large-scale binary interdiction games. *INFORMS J. Comput.* 34(2):890–908.

Cormican KJ, Morton DP, Wood RK (1998) Stochastic network interdiction. *Oper. Res.* 46(2):184–197.

Cornuejols G, Fisher M, Nemhauser G (1977) Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Sci.* 23:789–810.

Della Croce F, Scatamacchia R (2020) An exact approach for the bilevel knapsack problem with interdiction constraints and extensions. *Math. Programming* 183(1):249–281.

Dempe S, Zemkoho A (2020) *Bilevel optimization* (Springer, Cham, Switzerland).

DeNegre S (2011) Interdiction and discrete bilevel linear programming. PhD thesis, Lehigh University, Bethlehem, PA.

Dong L, Xu-chen L, Xiang-tao Y, Fei W (2010) A model for allocating protection resources in military logistics distribution system based on maximal covering problem. *2010 Internat. Conf. Logist. Systems Intelligent Management* (ICLSIM), vol. 1 (IEEE, New York), 98–101.

Du D, Ko KI, Hu X (2012) *Design and Analysis of Approximation Algorithms*, vol. 62 (Springer, New York).

Fischetti M, Ljubić I, Monaci M, Sinnl M (2017) A new general-purpose algorithm for mixed-integer bilevel linear programs. *Oper. Res.* 65(6):1615–1637.

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

Furini F, Ljubić I, Martin S, San Segundo P (2019) The maximum clique interdiction problem. *Eur. J. Oper. Res.* 277(1):112–127.

Furini F, Ljubić I, San Segundo P, Zhao Y (2021) A branch-and-cut algorithm for the edge interdiction clique problem. *Eur. J. Oper. Res.* 294(1):54–69.

Golden B (1978) A problem in network interdiction. *Naval Res. Logist. Quart.* 25(4):711–713.

Guestrin C, Krause A, Singh AP (2005) Near-optimal sensor placements in Gaussian processes. Dzeroski S, Raedt LD, Wrobel S, eds. *Proc. 22nd Internat. Conf. Machine Learn.* (Association for Computing Machinery, New York), 265–272.

Han, S, Gómez, A, Prokopyev, OA (2022) Fractional 0–1 programming and submodularity. *J. Glob. Optim.* https://doi.org/10.1007/s10898-022-01131-5.

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

Jeroslow RG (1985) The polynomial hierarchy and a simple model for competitive analysis. *Math. Programming* 32(2):146–164.

Kazemi E, Zadimoghaddam M, Karbasi A (2018) Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. Dy J, Krause A, eds. *Internat. Conf. Machine Learn.* (PMLR), 2544–2553.

Kempe D, Kleinberg J, Tardos É (2003) Maximizing the spread of influence through a social network. Getoor L, Senator T, Domingos P, Faloutsos C, eds. *Proc. 9th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York), 137–146.

Krause A, Golovin D (2014) Submodular function maximization. *Tractability* 3:71–104.

Krause A, McMahan HB, Guestrin C, Gupta A (2008) Robust submodular observation selection. *J. Machine Learn. Res.* 9(12):2761–2801.

Kunnumkal S, Martínez-de Albéniz V (2019) Tractable approximations for assortment planning with product costs. *Oper. Res.* 67(2):436–452.

Laporte G, Nickel S, Saldanha da Gama F, eds. (2015) *Location Science*, 1st ed. (Springer, Cham, Switzerland).

Ljubić I, Moreno E (2018) Outer approximation and submodular cuts for maximum capture facility location problems with random utilities. *Eur. J. Oper. Res.* 266(1):46–56.

Lozano L, Smith JC (2017a) A backward sampling framework for interdiction problems with fortification. *INFORMS J. Comput.* 29(1):123–139.

Lozano L, Smith JC (2017b) A value-function-based exact approach for the bilevel mixed-integer programming problem. *Oper. Res.* 65(3):768–786.

Lozano L, Smith JC, Kurz ME (2017) Solving the traveling salesman problem with interdiction and fortification. *Oper. Res. Lett.* 45(3):210–216.

Mc Carthy SM, Tambe M, Kiekintveld C, Gore M, Killion A (2016) Preventing illegal logging: Simultaneous optimization of resource teams and tactics for security. Schuurmans D, Wellman M, eds. *Proc. AAAI Conf. Artificial Intelligence* (AAAI Press, Palo Alto, CA), vol. 30.

McFadden D, Train K (2000) Mixed MNL models for discrete response. *J. Appl. Econometrics* 15(5):447–470.

Morton DP, Pan F, Saeger KJ (2007) Models for nuclear smuggling interdiction. *IIE Trans.* 39(1):3–14.

Nemhauser GL, Wolsey LA, Fisher ML (1978) An analysis of approximations for maximizing submodular set functions—I. *Math. Programming* 14(1):265–294.

Orlin JB, Schulz AS, Udwani R (2016) Robust monotone submodular function maximization. Louveaux Q, Skutella M, eds. *Integer Programming and Combinatorial Optimization* (Springer International Publishing, Cham), 312–324.

ReVelle C, Scholssberg M, Williams J (2008) Solving the maximal covering location problem with heuristic concentration. *Comput. Oper. Res.* 35(2):427–435.

Roboredo MC, Aizemberg L, Pessoa AA (2019) An exact approach for the r-interdiction covering problem with fortification. *Central Eur. J. Oper. Res.* 27(1):111–131.

Sakaue S, Ishihata M (2018) Accelerated best-first search with upper-bound computation for submodular function maximization. McIlraith S, Weinberger K, eds. *Proc. AAAI Conf. Artificial Intelligence* (AAAI Press, Palo Alto, CA), vol. 32.

Salvagnin D (2019) Some experiments with submodular function maximization via integer programming. Rousseau L-M, Stergiou K, eds. *Internat. Conf. Integration Constraint Programming Artificial Intelligence Oper Res*. (Springer, Cham, Switzerland), 488–501.

Scaparra MP, Church RL (2008a) A bilevel mixed-integer program for critical infrastructure protection planning. *Comput. Oper. Res.* 35(6):1905–1923.

Scaparra MP, Church RL (2008b) An exact solution approach for the interdiction median problem with fortification. *Eur. J. Oper. Res.* 189(1):76–92.

Sefair JA, Smith JC, Acevedo MA, Fletcher RJ Jr (2017) A defender-attacker model and algorithm for maximizing weighted expected hitting time with application to conservation planning. *IISE Trans.* 49(12):1112–1128.

Shen S, Smith JC, Goli R (2012) Exact interdiction models and algorithms for disconnecting networks via node deletions. *Discrete Optim.* 9(3):172–188.

Smith JC (2010) Basic interdiction models. Cochran JJ, Cox LA, Keskinocak P, Kharoufeh JP, Smith JC, eds. *Wiley Encyclopedia of Operations Research and Management Science* (Wiley, Hoboken, NJ).

Smith JC, Song Y (2020) A survey of network interdiction models and algorithms. *Eur. J. Oper. Res.* 283(3):797–811.

Tahernejad S, Ralphs TK, DeNegre ST (2020) A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. *Math. Programming Comput.* 12(4): 529–568.

Tang Y, Richard JPP, Smith JC (2016) A class of algorithms for mixed-integer bilevel min–max optimization. *J. Global Optim.* 66(2):225–262.

Tanınmış K, Aras N, Altınel İK (2021) Improved x-space algorithm for min-max bilevel problems with an application to misinformation spread in social networks. *Eur. J. Oper. Res*. 297(1):40–52.

Tawarmalani M, Ahmed S, Sahinidis NV (2002) Global optimization of 0-1 hyperbolic programs. *J. Global Optim.* 24(4):385–416.

Vohra RV, Hall NG (1993) A probabilistic analysis of the maximal covering location problem. *Discrete Appl. Math.* 43(2):175–183.

Von Stackelberg H (1952) *The Theory of the Market Economy* (Oxford University Press, New York).

Wollmer R (1964) Removing arcs from a network. *Oper. Res.* 12(6): 934–940.

Wood RK (1993) Deterministic network interdiction. *Math. Comput. Model.* 17(2):1–18.

Xu P, Wang L (2014) An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions. *Comput. Oper. Res.* 41:309–318.

Yue D, Gao J, Zeng B, You F (2019) A projection-based reformulation and decomposition algorithm for global optimization of a class of mixed integer bilevel linear programs. *J. Global Optim.* 73(1):27–57.

Zheng K, Albert LA (2018) An exact algorithm for solving the bilevel facility interdiction and fortification problem. *Oper. Res. Lett.* 46(6):573–578.