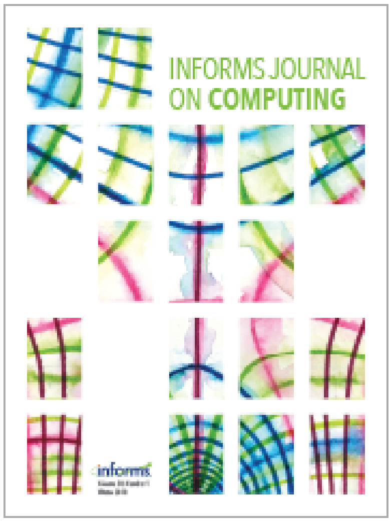



INFORMS Journal on Computing

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

# Interdiction Games and Monotonicity, with Application to Knapsack Problems

Matteo Fischetti, Ivana Ljubić, Michele Monaci, Markus Sinnl

To cite this article:
Matteo Fischetti, Ivana Ljubić, Michele Monaci, Markus Sinnl (2019) Interdiction Games and Monotonicity, with Application to Knapsack Problems. INFORMS Journal on Computing 31(2):390-410. https://doi.org/10.1287/ijoc.2018.0831

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

INFORMS JOURNAL ON COMPUTING
Vol. 31, No. 2, Spring 2019, pp. 390–410
ISSN 1091-9856 (print), ISSN 1526-5528 (online)



# Interdiction Games and Monotonicity, with Application to Knapsack Problems

**Matteo Fischetti,[a] Ivana Ljubić,[b] Michele Monaci,[c] Markus Sinnl[d]**

[a] Department of Information Engineering, University of Padua, 351200 Padua PD, Italy; [b] ESSEC Business School Paris, 95021 Cergy Pontoise, France; [c] Department of Information Engineering, University of Bologna, 40136 Bologna, Italy; [d] Department of Statistics and Operations Research, University of Vienna, 1090 Vienna, Austria

**Contact:** matteo.fischetti@unipd.it, https://orcid.org/0000-0001-6601-0568 (MF); ivana.ljubic@essec.edu, https://orcid.org/0000-0002-4834-6284 (IL); michele.monaci@unibo.it, https://orcid.org/0000-0001-9978-7613 (MM); markus.sinnl@univie.ac.at, https://orcid.org/0000-0003-1439-8702 (MS)

**Received:** December 5, 2016
**Revised:** July 14, 2017; November 14, 2017
**Accepted:** May 2, 2018
**Published Online in Articles in Advance:** April 26, 2019

https://doi.org/10.1287/ijoc.2018.0831



**Abstract.** Two-person interdiction games represent an important modeling concept for applications in marketing, defending critical infrastructure, stopping nuclear weapons projects, or preventing drug smuggling. We present an exact branch-and-cut algorithm for interdiction games under the assumption that feasible solutions of the follower problem satisfy a certain monotonicity property. Prominent examples from the literature that fall into this category are knapsack interdiction, matching interdiction, and packing interdiction problems. We also show how practically relevant interdiction variants of facility location and prize-collecting problems can be modeled in our setting. Our branch-and-cut algorithm uses a solution scheme akin to Benders decomposition based on a family of so-called interdiction cuts. We present modified and lifted versions of these cuts along with exact and heuristic procedures for the separation of interdiction cuts and heuristic separation procedures for the other versions. In addition, we derive further valid inequalities and present a new heuristic procedure. We computationally evaluate the proposed algorithm on a benchmark of 360 knapsack interdiction instances from literature, including 27 instances for which the optimal solution was not known. Our approach is able to solve each of them to optimality within about one minute of computing time on a standard PC (in most cases, within just seconds), and it is up to some orders of magnitude faster than any previous approach from the literature. To further assess the effectiveness of our branch-and-cut algorithm, an additional computational study is performed on 144 randomly generated instances based on 0/1 multidimensional knapsack problems.

**History:** Accepted by Karen Aardal, Area Editor for Design and Analysis of Algorithms.
**Funding:** This research was funded by the Vienna Science and Technology Fund [Project ICT15-014]. The work of M. Fischetti and M. Monaci was also supported by Ministero dell'Istruzione, dell'Università e della Ricerca, Italy [Project PRIN-2015]. The work of I. Ljubić and M. Sinnl was also supported by the Austrian Research Fund [Project P 26755-N19].

**Keywords:** interdiction games • bilevel optimization • mixed-integer optimization • branch-and-cut • multidimensional knapsack interdiction • prize-collecting interdiction

## 1. Introduction and Problem Definition

In many real-world optimization scenarios, a decision maker is not deciding alone but has to make her decisions taking decisions of other parties into account. In its simplest form, such a decision process can be modeled as a *two-player Stackelberg game* (von Stackelberg 1952). In such a game, there are two noncooperating players, denoted as leader and follower, taking their decisions in a sequential way (i.e., in the first round, the leader takes an action, and in the second round, the follower reacts to it). Thereby, follower decisions are influenced by the leader who possesses a complete knowledge of the follower optimization setting.

Problems of this nature can be tackled via bilevel optimization, a problem class that received increased attention in recent years and was used to model important problems arising in real-world applications. Because of their relevance, increasingly effective general purpose solvers have been designed very recently: for example, the works of Moore and Bard (1990), DeNegre (2011), Xu (2012), Xu and Wang (2014), Zeng and An (2014), Kleniati and Adjiman (2015), Fischetti et al. (2016, 2017a), and Lozano and Smith (2017) for mixed integer linear bilevel problems.

In this article, we consider a family of mixed integer linear bilevel problems known as *interdiction games* (IGs). This family of problems covers important and diverse applications, such as critical infrastructure defense (Brown et al. 2005, 2006), stopping nuclear weapons projects (Morton et al. 2007, Brown et al. 2009), drug smuggling (Washburn and Wood 1995), military applications (Cormican et al. 1998), and marketing (DeNegre 2011); also, surveys on *network interdiction* are given by Lim and Smith (2007), Smith and Lim (2008), Wood (2010), and Tang et al. (2015).

These problems can be seen as two-player zero-sum Stackelberg games, where the leader and follower typically share a set of *items*, and the leader can select some items and *interdict* their usage by the follower. The adversarial nature of the game is expressed through the common objective function that is optimized in the opposite direction by the two players. Typically, connection between the leader and the follower optimization problems is established through binary decision variables ("interdiction variables") that are controlled by the leader. The only constraints in the follower subproblem involving leader decision variables impose that, if an interdiction variable is selected by the leader, then certain actions of the follower are inhibited. Very often, these actions correspond to setting values of certain follower variables to zero, in which case a 1-1 correspondence between an interdiction leader variable and an interdicted follower variable exists.

More precisely, we focus on IGs stated in the following form:

$$\min_{x\in X} \max_{y\in\mathbb{R}^{n_2}} d^T y \tag{1}$$

$$Qy \le Q_0 \tag{2}$$

$$0 \le y_j \le u_j(1-x_j), \qquad \forall j \in N \tag{3}$$

$$y_j \text{ integer}, \qquad \forall j \in J_y, \tag{4}$$

where

$$X = \{x \in \mathbb{R}^{n_1} : Ax \le b, x_j \text{ integer } \forall j \in J_x, x_j \text{ binary } \forall j \in N\}$$

denotes the set of feasible leader solutions and $n_1$ and $n_2$ are the number of leader variables $x$ and follower variables $y$, respectively. We assume that $d, Q, Q_0, u, A$, and $b$ are given rational matrices/vectors of appropriate size. In particular, vector $u$ provides finite upper bounds on the follower variables $y_j$ involved in constraints (3).

The set $N$ appearing in (3) will be called the *item set*, and it corresponds to the $n = |N|$ items subject to possible interdiction. Therefore, the interpretation of constraints (3) is that the leader can completely "forbid" an item $j \in N$ by setting $x_j = 1$, but if she does not do so, then an arbitrary number of these items (up to $u_j$) can be taken by the follower. Set $J_x$ identifies instead the nonempty subset of indices of the integer-constrained variables in $x$, among which those in $N \subseteq J_x$ are assumed to be binary.

As to the follower, her variable set $\{1, \ldots, n_2\}$ is partitioned into $(N, R)$, where $R$ denotes the indices of the $y$ variables that are not directly linked to $x$ variables via constraints (3). Observe that the inner maximization problem over $y$ (namely, the follower problem for a fixed $x$) can be either a Linear Program (LP) or a Mixed Integer Linear Program (MILP) depending on whether the set $J_y$ of follower integer-constrained variables is empty or not. Also note that we do not require $N \subseteq J_y$ (i.e., interdicted follower variables $y_j$ with $j \in N$ are not necessarily required to be integer, whereas the corresponding $x_j$ must be binary). In addition, in our model, we allow the follower to include "additional variables" $y_j$ with $j \notin N$. Thus, the family of problems that we address includes but is not limited to the IGs typically studied in the literature, where all follower variables can be interdicted by the leader.

In the following, we will denote by $y_N = (y_j)_{j\in N}$ the vector containing only the variables that can be interdicted and by $y_R = (y_j)_{j\in R}$ the vector of remaining decision variables at the follower level. When useful, we will also use notation $Q = (Q_N, Q_R)$ and $d^T = (d_N^T, d_R^T)$.

Whenever $x \in X$, we will say that $x$ is a *feasible interdiction policy*. Given a feasible interdiction policy $\hat{x}$, we will say that $y \in \mathbb{R}^{n_2}$ is a *feasible follower solution* for $\hat{x}$ if $y$ satisfies (2), (3), and (4). In addition, we will denote by $y^*(\hat{x})$ an optimal follower solution obtained by solving (1)–(4) for $x = \hat{x}$. We assume that variable bounds on $x$ and $y$ other than those in (3), if any, are explicitly included in constraints $Ax \le b$ and $Qy \le Q_0$, respectively. Notation $A_j$ or $Q_j$ will be used for the $j$th column of matrix $A$ or $Q$, respectively.

As customary, in what follows, we will assume that the follower problem is feasible and bounded for any feasible interdiction policy $x$.

As observed above, IGs are a special case of more general bilevel optimization problems, in which the leader and the follower take their decisions in a hierarchical fashion, but their own objective functions and the interplay between their decisions can be of a more general form. In IGs (as opposed to the more general bilevel optimization), there is no need to distinguish between the optimistic and pessimistic setting, because both players optimize the same objective function—but in the opposite direction.

The following *(downward) monotonicity* is an important assumption made throughout this article that will be exploited for deriving a valid branch-and-cut (B&C) approach based on interdiction constraints.

**Assumption 1** (Downward Monotonicity). *If $\hat{y} = (\hat{y}_N, \hat{y}_R)$ is a feasible follower solution for a given $x$ and $y' = (y'_N, \hat{y}_R)$ satisfies constraints (4) and $0 \le y'_N \le \hat{y}_N$, then $y'$ is also a feasible follower solution for $x$.*

If all follower variables are binary and $R = \emptyset$, Assumption 1 implies that the family of sets $\mathcal{S} = \{S \subseteq N : Q\chi_S \le Q_0\} \subseteq 2^N$ defines an *independent system*, where $\chi_S$ denotes the 0/1 incidence vector of $S$. However, as shown in Section 2, there are many other classes of IGs that satisfy it.

Observe that the monotonicity assumption does not reduce the computational complexity of the problem in that it is satisfied (among others) by the knapsack interdiction problem (KIP), which has been shown to be $\Sigma_2^P$-hard by Caprara et al. (2014). Moreover, it has been shown by Zenklusen (2010) and Dinitz and Gupta (2013) that monotone IGs remain NP hard, even when $|N| = n_1 = n_2$ and the follower problem is a pure LP (i.e., $J_y = \emptyset$).

Because of monotonicity, we will assume without loss of generality that $d_N > 0$; otherwise, all variables $y_j$ with $j \in N$ and $d_j \leq 0$ could be fixed to zero and removed from the model. In addition, we will assume $Q_N \geq 0$ because of the following result.

**Theorem 1.** *Assumption 1 holds if and only if (iff) there exists a formulation (2)–(4) of the follower problem with* $Q_N \geq 0$.

**Proof.** The fact that the condition is sufficient is obvious. To show that it is also necessary, consider any formulation of the follower problem (2)–(4). We will prove the claim by showing that every negative entry in $Q_N$, if any, can be increased to zero to produce an alternative system that is not worse (as $y_N \geq 0$) than the original one. To this end, let $q_i^T y \leq q_{i0}$ denote any inequality in $Qy \leq Q_0$ with $q_{ih} < 0$ for a certain $h \in N$ (if any). We have to show that the improved inequality $\bar{q}_i^T y \leq q_{i0}$ is valid for the follower problem, where $\bar{q}_i$ is obtained from $q_i$ by setting $\bar{q}_{ih} = 0$ and leaving the other entries unchanged. Indeed, take any feasible follower solution $\hat{y}$, and let $y'$ be obtained from $\hat{y}$ by setting $y'_h = 0$. Because of Assumption 1, $y'$ is a feasible follower solution; hence, $q_i^T y' \leq q_{i0}$ holds. By construction,

$$\bar{q}_i^T \hat{y} = \sum_{j \neq h} \bar{q}_{ij} \hat{y}_j = \sum_{j \neq h} q_{ij} y'_j = q_i^T y' \leq q_{i0};$$

hence, $\bar{q}_i^T y \leq q_{i0}$ is a valid inequality because of the arbitrariness of $\hat{y}$. □

### 1.1. Our Contribution

In this paper, we study IGs under Assumption 1. We allow for an extended formulation of the follower problem (i.e., for $R \neq \emptyset$), which is useful to model practically relevant situations. Various examples of applications are discussed, including IGs for important network problems, such as the facility location problem and the prize-collecting traveling salesman problem (PCTSP).

We propose a Benders-like algorithm, in which the problem is reformulated as a single-level problem (with an exponential number of constraints called *interdiction cuts*) and all follower variables are projected out. We introduce a new family of interdiction cuts that generalize those given (without proof) by Ralphs (2015) and Caprara et al. (2016) for the special case $R = \emptyset$, giving a formal proof of their validity for general monotone IGs and showing that they are instead not valid for the nonmonotone case. We then propose a procedure for lifting these cuts along with a family of related cuts with validity that is based on certain integer disjunctions. We also introduce a family of new cuts exploiting dominances among items. For interdiction cuts, exact and heuristic separation procedures are designed, whereas for the other families of cuts, we propose fast heuristic separation algorithms. Moreover, we present a fast primal heuristic procedure for quite general (not necessarily monotone) IGs. This heuristic turns out to be extremely effective on some classes of instances, because its execution within a preprocessing procedure dramatically reduces the computing time needed to prove optimality. In our computational study, we consider benchmark sets for the KIP proposed by DeNegre (2011), Tang et al. (2015), and Caprara et al. (2016) and show that our algorithm significantly outperforms the specialized codes proposed in DeNegre (2011) and Caprara et al. (2016) as well as the state-of-the art approaches for IGs (Tang et al. 2015) and for general bilevel mixed integer programming (Fischetti et al. 2017a). We test 360 knapsack interdiction instances from the literature and prove the optimality for all of them—including 27 problems that were previously unsolved. Our algorithm needs at most 84 seconds for solving any of these instances on a standard PC (for only four of these 360 instances, it requires more than 10 seconds), thus outperforming previous approaches from the literature by orders of magnitudes. In addition to the above knapsack interdiction instances from literature, we also generated 144 random instances based on 0/1 multidimensional knapsack problems, with the aim of analyzing the dependency of our approach on the number of leader and follower constraints. To the best of our knowledge, this is by far the largest computational study on IGs reported in the literature.

### 1.2. Outline

In Section 2, we illustrate a number of important practical problems that can be modeled as IGs satisfying the monotonicity property. The basic idea of a B&C framework using interdiction cuts is provided in Section 3, where we also provide theoretical foundations for deriving modified/lifted interdiction cuts as well as valid inequalities based on dominance criteria. In Section 4, we provide implementation details of our framework, including separation algorithms and a primal heuristic procedure. Finally, Section 5 reports our computational study, whereas Section 6 gives a short conclusion.

## 2. Applications

We briefly describe some relevant IGs that satisfy the monotonicity assumption and therefore, can be tackled by our proposed methods.

An important observation is that Theorem 1 requires the nonnegativity condition to be satisfied only by the columns of the constraint matrix $Q$ associated with follower variables that can be interdicted—the remaining columns being immaterial for what concerns the downward monotonicity property. This fact greatly extends the applicability of our results, because many practically relevant interdiction variants of classical problems, like the facility location and the PCTSP (and other similar prize collecting problems), can be handled by our approach.

## 2.1. Knapsack Problems

A prominent example of an IG that satisfies the monotonicity property is the KIP studied by DeNegre (2011), Tang et al. (2015), and Caprara et al. (2016). The problem models a Stackelberg game, in which both leader and follower own their private knapsacks with capacities $a_0$ and $q_0$ (say) and fill them by choosing items from a common item set $N$. Each item $j \in N$ has a positive profit $d_j$ and weights $a_j$ and $q_j$ in the leader and the follower problems, respectively. In the first step, the leader chooses some of the items while respecting her own knapsack capacity (called *interdiction budget*). In the second step, the follower solves a 0/1 knapsack problem and selects some of the items that are not taken by the leader to maximize the profit while respecting her capacity constraint. The goal of the leader is to obtain the worst possible outcome for the follower. Using binary variables $x_j$ and $y_j$ to denote the items selected by the leader and the follower, respectively, KIP can be modelled as an IG as follows:

$$\min_x \max_y \sum_{j\in N} d_j y_j \tag{5}$$

$$\sum_{j\in N} a_j x_j \le a_0 \tag{6}$$

$$\sum_{j\in N} q_j y_j \le q_0 \tag{7}$$

$$y_j \le 1 - x_j, \qquad \forall j \in N \tag{8}$$

$$x_j, y_j \in \{0, 1\} \qquad \forall j \in N. \tag{9}$$

As mentioned in DeNegre (2011), a typical application of this problem arises in marketing when a company $A$ dominates the market and company $B$ wishes to design a marketing campaign, while choosing the specific geographic regions to target, subject to the available budget. Whenever companies $A$ and $B$ target the same region, the marketing campaign of company $B$ fails. Consequently, the goal of the hostile company $A$ is to minimize the established benefit of company $B$. In DeNegre (2011), the author solves KIP through a cutting-plane procedure, in which the problem is reformulated as a single-level problem with an exponential number of constraints to be separated on the fly by using disjunctive cut-generating LPs. In Caprara et al. (2016), a problem-tailored approach is introduced; in this iterative MILP-based procedure, the lower and upper bounds are sequentially improved until an optimal solution (or a given time limit) is reached. Finally, because of the simplicity of its definition, the KIP is a commonly used benchmark for testing solvers for bilevel optimization as well. In Tang et al. (2015), the authors propose three ideas for deriving a generic solver for IGs. A new generic solver for bilevel mixed integer programs has been recently proposed in Fischetti et al. (2017a). In both papers, KIP instances constitute an important part of the considered benchmark set.

Of course, the most natural generalizations of the knapsack problem, namely the multidimensional knapsack problem and the multiple knapsack problem, satisfy the monotonicity property as well.

## 2.2. Facility Location Problems

Many IGs considered in the previous literature assume that every variable at the follower level can be interdicted by the leader (i.e., that $R = \emptyset$). In the following, we illustrate an important application from marketing/facility location, which requires existence of additional decision variables at the follower level that cannot be explicitly interdicted by the leader but contribute to the follower objective function. Assume that there are two companies, say $A$ (the leader) and $B$ (the follower), that compete for the same set of customers. Let $I$ be the set of available facilities and $J$ be the set of customers served by them. Assume that company $A$ dominates the market (i.e., it has already established service facilities, and all customers are currently served by $A$) and that company $B$ wants to enter the market. For $B$, facility opening costs $f_i \ge 0$ need to be paid for each $i \in I$, and profit $p_{ij} \ge 0$ can be collected if customer $j \in J$ is served by the open facility $i \in I$. The leader can provide an incentive $a_j \ge 0$ to a customer $j \in J$ to convince her not to switch the service, but there is a limited (interdiction) budget $b > 0$ to do so. The follower aims to maximize its revenue, assuming that all customers that are not "interdicted" by the leader will switch to the follower (if it is able to provide the service). The revenue for company $B$ is defined as the sum of collected profits minus the costs for opening the facilities. The leader defines the interdiction policy using binary variables $(v, x)$. Each variable $v_j$ takes value 1 iff customer $j \in J$ receives an incentive from company $A$. In addition, there are auxiliary variables $x_{ij}$ that are set to one (for all $i \in I$) whenever customer $j$ receives an incentive from $A$. The $x$ variables are used to possibly interdict some $y$ variables in the follower, where $y_{ij} = 1$ iff customer $j \in J$ is served by facility $i \in I$ of company $B$. Finally, the follower also defines variables $z_i$ to denote the set of facilities that company $B$ has to open. We obtain the following IG formulation:

$$\min_{(x,v)} \max_{(y,z)} \sum_{i\in I}\sum_{j\in J} p_{ij}y_{ij} - \sum_{i\in I} f_i z_i \tag{10}$$

$$x_{ij} - v_j = 0, \qquad \forall i\in I, j\in J \tag{11}$$

$$\sum_{j\in J} a_j v_j \le b \tag{12}$$

$$y_{ij} \le 1 - x_{ij}, \qquad \forall i\in I, j\in J \tag{13}$$

$$y_{ij} - z_i \le 0, \qquad \forall i\in I, j\in J \tag{14}$$

$$\sum_{i\in I} y_{ij} \le 1, \qquad \forall j\in J \tag{15}$$

$$x_{ij}, y_{ij} \in \{0,1\}, \qquad \forall i\in I, j\in J \tag{16}$$

$$v_j \in \{0,1\}, \qquad \forall j\in J \tag{17}$$

$$z_i \in \{0,1\}. \qquad \forall i\in I \tag{18}$$

Thus, in this example, allocation variables $y_{ij}$ (that are the only ones that can be interdicted by the leader) do satisfy the monotonicity property, whereas the remaining variables at the follower level ($z_i$) contribute to the objective function but are not subject to interdiction and as such, do not need to satisfy the monotonicity property.

## 2.3. Prize-Collecting Problems

There are many problems in literature that are of a *prize-collecting* type, including the PCTSP (Balas 1989, 2007, Bienstock et al. 1993), the *prize-collecting Steiner tree problem* (Ljubić et al. 2006, Prodon et al. 2010, Fischetti et al. 2017b), and various variants of the *orienteering problem* (Vansteenwegen et al. 2011). For example, in the PCTSP, we are given a complete graph $G = (V, E)$ with positive prizes $p_i$ associated with nodes $i \in V$ and positive costs $c_e$ with the edges $e \in E$. We are looking for a subset of nodes, such that the revenue (i.e., the difference between the sum of the prizes in this subset and the cost of a tour on it) is maximized. In other words, a salesperson is looking for the most profitable tour through a subset of nodes, taking into account both the prize that she can collect from each visited client and the travel cost between clients. Similar to the previous example, assume that there are two salespersons $A$ and $B$, where $A$ dominates the market, whereas $B$ wants to enter it. Each client $i \in V$ only allows for one salesperson to collect its prize. Again, $A$ can provide some incentive $a_i \ge 0$ to each client $i$ not to switch to the competitor subject to a limited budget $b \ge 0$. The goal of $A$ is now to find the best subset of clients to offer incentives (i.e., to interdict), such that the revenue of $B$ is minimized, assuming that all clients to which $A$ offered the incentive do not want the service of $B$ (i.e., even if $B$ visits such a client in her tour, $B$ does not collect its prize).

Let the leader binary variable $x_i$ be one iff $A$ interdicts $i \in V$, and let the follower binary variable $y_i$ by one iff $B$ collects the prize of client $i \in V$ (i.e., iff client $i$ is visited by $B$ and is not interdicted by $A$). We also introduce additional follower binary variables $z_e^E = 1$ iff $B$ travels on edge $e \in E$ and additional follower binary variables $z_i^V = 1$ iff $B$ visits client $i \in V$. The described problem can then be modeled as follows:

$$\min_x \max_{(y,z^E,z^V)} \sum_{i\in V} p_i y_i - \sum_{e\in E} c_e z_e^E \tag{19}$$

$$\sum_{i\in V} a_i x_i \le b \tag{20}$$

$$y_i \le 1 - x_i, \qquad \forall i\in V \tag{21}$$

$$y_i - z_i^V \le 0, \qquad \forall i\in V \tag{22}$$

$$(z^E, z^V) \in F \tag{23}$$

$$x_i, y_i \in \{0,1\}, \qquad \forall i\in V \tag{24}$$

$$z_e^E \in \{0,1\}, \qquad \forall e\in E \tag{25}$$

$$z_i^V \in \{0,1\} \qquad \forall i\in V, \tag{26}$$

where $F$ denotes the set of the feasible follower solutions (i.e., $F$ contains the incidence vectors $S(z^E, z^V)$ of all simple cycles of $G$ and the corresponding visited nodes). (In a variant of the problem, all such cycles can be required to satisfy additional conditions; e.g., they must visit a certain "depot" node.)

Constraints (22) ensure that $B$ can only collect the prize of client $i$ if she visits $i$, whereas constraint (23) states follower feasibility in a generic way—this could be modeled, for example, by using subtour elimination constraints (e.g., Bienstock et al. 1993). The interdiction actions of $A$ are modeled by (21). Note that, under the very reasonable assumption that edge costs $c_e$ satisfy the triangle inequality, the optimal follower cycle will only visit noninterdicted clients.

It is easy to see that variables $y_i$ (which are the only interdictable follower variables) fulfill the monotonicity property, because the system $Qy \le Q_0$ is just (22) in this case; hence, $Q_N = I \ge 0$.

Observe that the above formulation gives a very general recipe to formulate interdiction problems fulfilling the monotonicity property, because the generic constraint (23) can be replaced with any other set of constraints (e.g., if one wants to consider the prize-collecting Steiner tree as the follower problem, constraint (23) can be replaced by a set of constraints specifying that $(z^E, z^V)$ describes a suitable Steiner tree).

## 2.4. Other Problems

Other relevant problems from the literature that fall into the category of IGs under monotonicity are the set-packing interdiction problem (Dinitz and Gupta 2013), the maximum weight-matching interdiction problem

(Zenklusen 2010), and the maximum weight-independent set interdiction problem (Bazgan et al. 2011). Observe that the independent set problem is an example of a *hereditary graph problem* (Halldórsson 2000). A graph is said to possess the hereditary property $\Pi$ if every subgraph induced by its node subsets also possesses the same property. For a property $\Pi$, the corresponding hereditary graph problem is then defined to find the maximum node-weighted subgraph satisfying $\Pi$. It is easy to see that feasible solutions to such a problem define an independent system (with respect to the nodes); thus, node interdiction variants of these problems fall within our setting.

However, some other important problems in network interdiction, including the interdiction of shortest paths (Israeli and Wood 2002, Song and Shen 2016) or multicommodity flows (Lim and Smith 2007), do not fit in our framework.

## 3. Interdiction Cuts

In this section, we first recall the idea of reformulating IGs as single-level problems with an exponential number of constraints called interdiction cuts. This idea has been frequently used in the interdiction literature (e.g., the seminal paper by Israeli and Wood (2002) or the survey by Wood (2010)). However, in most of the cases, the quality of derived cuts is not satisfactory, because large big-*M* coefficients (or indicator constraints) must be used. In the remainder of this section, we show that big-*M* values can be avoided (resulting in much tighter interdiction cuts) under the assumption that the follower satisfies the monotonicity property. We then provide a counterexample that shows that these specific interdiction cuts are not valid if the monotonicity property is violated. We finally conclude this section by providing additional theoretical results for strengthening and lifting the basic form of interdiction cuts.

### 3.1. Single-Level Reformulation

For a given $x \in X$, we define the *value function* as follows:

$$\Phi(x) = \max_{y \in \mathbb{R}^{n_2}} d^T y \tag{27}$$

$$Qy \le Q_0 \tag{28}$$

$$0 \le y_j \le u_j(1 - x_j), \quad \forall j \in N \tag{29}$$

$$y_j \text{ integer}, \quad \forall j \in J_y, \tag{30}$$

so that problem (1)–(4) can be restated in the $\mathbb{R}^{n_1+1}$ space as

$$\min_{x \in \mathbb{R}^{n_1}, w \in \mathbb{R}} w \tag{31}$$

$$w \ge \Phi(x) \tag{32}$$

$$Ax \le b \tag{33}$$

$$x_j \text{ integer}, \quad \forall j \in J_x \tag{34}$$

$$x_j \in \{0, 1\}, \quad \forall j \in N. \tag{35}$$

Constraint (32) can be rewritten in the following different form (e.g., Wood 2010). We consider an alternative formulation of the follower subproblem (27)–(30), in which interdiction constraints (29) are removed and a penalization term $-\sum_{j \in N} M_j x_j y_j$ is added to the objective function. For sufficiently large values of multipliers $M_j$, this penalty term guarantees that any optimal solution of the follower has $x_j y_j = 0 \ \forall j \in N$, no matter the choice of $x$. For a given $x$, the follower subproblem can then be rewritten as

$$\Phi(x) = \max\left\{ d^T y - \sum_{j \in N} M_j x_j y_j : y \in Y \right\}, \tag{36}$$

where

$$Y = \{y \in \mathbb{R}^{n_2} : Qy \le Q_0, \quad 0 \le y_j \le u_j \quad \forall j \in N, \\ y_j \text{ integer } \forall j \in J_y\}.$$

Note that, using the reformulation above, the feasible space $Y$ of the follower does not depend on the interdiction policy $x$ anymore. Furthermore, for a given $x$, the objective function is linear, which means that its optimal solution corresponds to a vertex of $\mathrm{conv}(Y)$. Consequently, the follower subproblem can be restated as

$$\Phi(x) = \max\left\{ d^T y - \sum_{j \in N} M_j x_j y_j : y \in \hat{Y} \right\}, \tag{37}$$

where $\hat{Y}$ contains all extreme points of $\mathrm{conv}(Y)$.

One can, therefore, derive a reformulation of the IG as a single-level MILP akin to Benders decomposition (with the follower variables $y$ being projected out of the model), namely

$$\min_{x \in \mathbb{R}^{n_1}, w \in \mathbb{R}} w \tag{38}$$

$$w \ge d^T \hat{y} - \sum_{j \in N} M_j x_j \hat{y}_j \quad \forall \hat{y} \in \hat{Y} \tag{39}$$

$$Ax \le b \tag{40}$$

$$x_j \text{ integer}, \quad \forall j \in J_x \tag{41}$$

$$x_j \text{ binary}, \quad \forall j \in N. \tag{42}$$

In the following, we refer to (39) as interdiction cuts.

The above reformulation projects $y$ variables out from the model and allows for the application of a B&C procedure, in which interdiction cuts are initially removed from the model and then dynamically added through the following separation procedure. Given an optimal (possibly fractional) solution $(w^*, x^*)$ at the current B&C node, the follower subproblem is solved for $x = x^*$ to obtain an optimal point $y^* \in \hat{Y}$. If the current solution violates the interdiction cut (39) associated with $\hat{y} = y^*$, then this globally valid cut is added to the current formulation; otherwise, no interdiction cut needs to be generated for $(w^*, x^*)$.

The single-level reformulation above has already been used in the literature within an iterative cutting-plane procedure (e.g., the procedure called CP in Caprara et al. 2016 or Israeli and Wood 2002 and Wood 2010). In all of these approaches, however, every time a single interdiction cut is added, the current model is solved as an MILP before the new cut is separated in a cutting-plane fashion.

### 3.2. Interdiction Cuts for Followers with the Property of Monotonicity

A crucial point for the effectiveness of the proposed reformulation is how to determine appropriate values for $M_j$ to guarantee tight lower bounds—the smaller these coefficients, the better the formulation. The choice of $M_j$ is problem dependent (e.g., Wood 2010). For the KIP, it has been observed by Ralphs (2015) and Caprara et al. (2016) that the values can be set as $M_j = d_j$ for all $j \in N$, although no formal proof for this result has been stated explicitly. In the following, we prove validity of these tightened constraints not only for the KIP but also, for the broader family of IGs satisfying the property of monotonicity—allowing, in particular, for $R \neq \emptyset$.

**Theorem 2.** *Under Assumption 1, the following interdiction cuts are valid for (31)–(35):*

$$w \geq \sum_{j\in R} d_j \hat{y}_j + \sum_{j\in N} d_j \hat{y}_j (1 - x_j) \qquad \forall \hat{y} \in \hat{Y}. \tag{43}$$

**Proof.** Let $\hat{y} \in \hat{Y}$, and take any feasible solution $(w, x)$ to (31)-(35). Define a follower solution $y' = (y'_N, \hat{y}_R)$, where $y'_j = \hat{y}_j(1 - x_j)$ for all $j \in N$. Because of Assumption 1, $y'$ is a feasible follower solution for the given $x$; hence,

$$\begin{aligned} w \geq \Phi(x) \geq d^T y' &= d_R^T y'_R + d_N^T y'_N \\ &= \sum_{j\in R} d_j \hat{y}_j + \sum_{j\in N} d_j \hat{y}_j (1 - x_j), \end{aligned}$$

as claimed. □

Note that the point $\hat{y} \in \hat{Y}$ in the theorem above does not depend on $x$ (i.e., it does not have to satisfy any complementarity condition of the form (29)). Furthermore, we observe that interdiction cuts (43) are valid not only for extreme points $\hat{y} \in \hat{Y}$ but also, for any arbitrary point in $Y$.

It is worth observing that, in case $R \neq \emptyset$, the above proof remains valid even if Assumption 1 is relaxed as follows: "if $\hat{y} = (\hat{y}_N, \hat{y}_R)$ is a feasible follower solution for a given $x$ and $y'_N$ satisfies constraints (4) and $0 \leq y'_N \leq \hat{y}_N$, then there exists $y'_R$ with $d^T y'_R \geq d^T \hat{y}_R$, such that $y' = (y'_N, y'_R)$ is a feasible follower solution for $x$."

**Theorem 3.** *Under Assumption 1, IG (31)–(35) can be reformulated by replacing constraint (32) with the family of (linear) interdiction cuts (43).*

**Proof.** Observe that there are exponentially many interdiction cuts (43). We have to show that, for any feasible interdiction policy $x$, these inequalities imply $w \geq \Phi(x)$. Indeed, the interdiction inequality for $\hat{y} = y^*(x)$ reads

$$\begin{aligned} w \geq \sum_{j\in R} d_j \hat{y}_j + \sum_{j\in N} d_j y_j^*(x)(1 - x_j) = \sum_{j\in R} d_j \hat{y}_j + \sum_{j\in N} d_j y_j^*(x) \\ = \Phi(x), \end{aligned}$$

where the first equality follows from the fact that, for all $j \in N$, $y_j^*(x) \cdot x_j = 0$ owing to (29). □

**Definition 1.** A follower solution $\hat{y} = (\hat{y}_N, \hat{y}_R) \in \hat{Y}$ is *maximal* if there is no $(y'_N, \hat{y}_R) \in \hat{Y} \backslash \{\hat{y}\}$, such that $y'_N \geq \hat{y}_N$.

The following result shows that, among all extreme points $\hat{y} \in \hat{Y}$, it is in fact sufficient to consider only maximal solutions. This fact can be computationally exploited to avoid the generation of useless interdiction cuts; Section 4 has additional details.

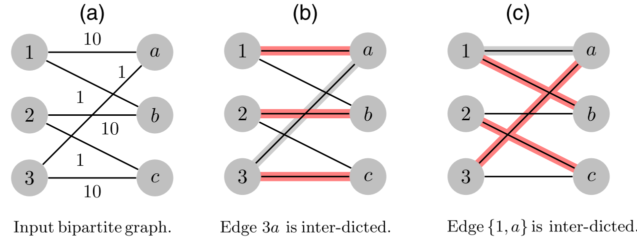
**Theorem 4.** *Let $\hat{y} = (\hat{y}_N, \hat{y}_R) \in \hat{Y}$ be nonmaximal, and let $y' = (y'_N, \hat{y}_R) \in \hat{Y} \backslash \{\hat{y}\}$ be such that $y'_N \geq \hat{y}_N$. Then, under Assumption 1, the interdiction inequality (43) for $\hat{y}$ is dominated by that for $y'$.*

**Proof.** Obvious as, for all $j \in N$, $x_j \in [0, 1]$ implies $y'_j(1 - x_j) \geq \hat{y}_j(1 - x_j)$. □

In the following example, we show that, by dropping our assumption that the follower solutions satisfy the monotonicity property, the resulting interdiction cuts (43) are not valid. To this end, consider a problem instance, in which the follower solves the maximum weight assignment problem (i.e., a perfect matching on a bipartite graph) and the leader tries to minimize its outcome by interdicting some of the edges of the input bipartite graph.

Consider the graph depicted in Figure 1, and assume that the interdiction budget allows the leader to interdict at most one edge. If the leader interdicts

**Figure 1.** (Color online) Assignment Interdiction Problem and Two Possible Solutions




*Notes.* (a) Example of the assignment interdiction problem. Weight of the horizontal edges is equal to 10; weight of the remaining edges is 1. Panels (b) and (c) show two possible solutions $y^*(x) \in \hat{Y}$ for two feasible interdiction policies $x \in X$: gray edges are interdicted by the leader, and dark gray (red online) edges are chosen by the follower.

edge $3a$, we have $\Phi(x) = 30$ and $y^*(x) = \chi_{\{1a,2b,3c\}}$. The resulting interdiction cut for $\hat{y} = y^*(x)$ would be $w \geq 30 - 10x_{1a} - 10x_{2b} - 10x_{3c}$, which is, however, violated by the feasible leader policy $x'$, in which the leader interdicts edge $1a$ for which $y^*(x') = \chi_{\{1b,2c,3a\}}$ and $w' = \Phi(x') = 3$. Note that the above cut would instead be valid for a nonperfect variant of the problem allowing for isolated nodes—that would in fact satisfy the monotonicity property.

### 3.3. New Classes of Cuts

In this subsection, we address the questions of how to modify the basic form of interdiction cuts (43) to derive further valid inequalities and how to lift them (in a computationally inexpensive way if possible) to improve the performance of the resulting B&C algorithm. We first propose a new class of modified interdiction cuts, then we introduce a lifting procedure for interdiction cuts, and finally, we present a new family of cuts based on dominance relationships among items. For the validity of the new cuts, we impose an additional assumption.

**Assumption 2.** *All follower variables $y_N$ are binary (i.e., $N \subseteq J_y$ and $u = 1$).*

**Theorem 5.** *For any $\hat{y} \in \hat{Y}$, let $S_a = \{a_1, \ldots, a_K\} \subset N$ and $S_b = \{b_1, \ldots, b_K\} \subset N$ be two distinct collections of items, such that $\hat{y}_{a_k} = 1$, $\hat{y}_{b_k} = 0$, and $Q_{a_k} \geq Q_{b_k}$ for $k = 1, \ldots, K$. Under Assumptions 1 and 2, the following modified interdiction cut is valid for (31)–(35):*

$$w \geq \sum_{j \in R} d_j \hat{y}_j + \sum_{j \in N} d_j \hat{y}_j (1 - x_j) + \sum_{k=1}^{K} d_{b_k}(x_{a_k} - x_{b_k}). \tag{44}$$

**Proof.** For any given $\hat{y} \in \hat{Y}$, we have to show that (44) is satisfied by any given feasible interdiction policy $x$. In case $x_{a_k} - x_{b_k} \leq 0$ for each $k = 1, \ldots, K$, this is obvious, because $x$ satisfies the interdiction inequality (43). Otherwise, let $\bar{\mathcal{K}} = \{k \in \{1, \ldots, K\} : x_{a_k} - x_{b_k} = 1\}$ (i.e., $x_{a_k} = 1$ and $x_{b_k} = 0$ for each $k \in \bar{\mathcal{K}}$). Consider the alternative follower solution $y'$ obtained from $\hat{y}$ by flipping, for each $k \in \bar{\mathcal{K}}$, $\hat{y}_{a_k}$ and $\hat{y}_{b_k}$ (i.e., by setting $y'_{a_k} = 0$ and $y'_{b_k} = 1$) and leaving the remaining entries unchanged. Under the assumption $Q_{a_k} \geq Q_{b_k}$, one has $Qy' \leq Q_0$ (i.e., $y' \in \hat{Y}$); hence, $x$ satisfies the interdiction inequality associated with $y'$, namely

$$\begin{aligned} w &\geq \sum_{j \in R} d_j y'_j + \sum_{j \in N} d_j y'_j (1 - x_j) = \\ &= \sum_{j \in R} d_j \underbrace{y'_j}_{=\hat{y}_j} + \sum_{j \in N \setminus \{a_k, b_k : k \in \bar{\mathcal{K}}\}} d_j \underbrace{y'_j}_{=\hat{y}_j} (1 - x_j) \\ &\quad + \sum_{k \in \bar{\mathcal{K}}} \left( d_{a_k} \underbrace{y'_{a_k}}_{=0} (1 - x_{a_k}) + d_{b_k} \underbrace{y'_{b_k}}_{=1} (1 - x_{b_k}) \right). \end{aligned} \tag{45}$$

Rewrite (44) in a similar way to obtain

$$\begin{aligned} w \geq \sum_{j \in R} d_j \hat{y}_j &+ \sum_{j \in N \setminus \{a_k, b_k : k \in \bar{\mathcal{K}}\}} d_j \hat{y}_j (1 - x_j) \\ &+ \sum_{k \in \bar{\mathcal{K}}} \left( d_{a_k} \underbrace{\hat{y}_{a_k}}_{=1} (1 - x_{a_k}) + d_{b_k} \underbrace{\hat{y}_{b_k}}_{=0} (1 - x_{b_k}) \right. \\ &\left. + d_{b_k}(x_{a_k} - x_{b_k}) \right) + \sum_{k \in \mathcal{K} \setminus \bar{\mathcal{K}}} d_{b_k}(x_{a_k} - x_{b_k}). \end{aligned} \tag{46}$$

Because (45) is a valid inequality and the left-hand sides of both (45) and (46) are the same, it remains to be shown that the right-hand side of (46) is smaller than or equal to the right-hand side of (45). To this end, it is enough to subtract the right-hand side of (46) from the right-hand side of (45) to obtain

$$\begin{aligned} &\underbrace{\sum_{k \in \bar{\mathcal{K}}} \left( d_{b_k}(1 - \underbrace{x_{b_k}}_{=0}) - d_{a_k}(1 - \underbrace{x_{a_k}}_{=1}) - d_{b_k}(\underbrace{x_{a_k} - x_{b_k}}_{=1}) \right)}_{=0} \\ &\quad - \sum_{k \in \mathcal{K} \setminus \bar{\mathcal{K}}} \underbrace{d_{b_k}}_{>0} \underbrace{(x_{a_k} - x_{b_k})}_{\leq 0} \geq 0. \quad \square \end{aligned} \tag{47}$$

As the above proof shows, the modified interdiction cuts (44) can be seen as disjunctive cuts based on the disjunctions $x_{a_k} - x_{b_k} \leq 0$ or $\geq 1$, with validity that exploits the integrality of $x$. Note that, even if $d_{b_k} > 0$ by assumption, the additional terms $d_{b_k}(x_{a_k} - x_{b_k})$ in the right-hand side can be negative for some feasible $x$, meaning that these cuts do not dominate (and are not dominated by) interdiction cuts.

Interdiction cuts can also be lifted by exploiting some additional properties of $Q$, thus producing a new family of cuts that are strictly better (i.e., that dominate) the standard ones (43).

**Theorem 6.** *For a given $\hat{y} \in \hat{Y}$, let $S_a = \{a_1, \ldots, a_K\} \subset N$ and $S_b = \{b_1, \ldots, b_K\} \subset N$ be two distinct collections of items, such that $\hat{y}_{a_k} = 1$, $\hat{y}_{b_k} = 0$, $d_{a_k} < d_{b_k}$, and $Q_{a_k} \geq Q_{b_k}$ for each $k \in \{1, \ldots, K\}$. Under Assumptions 1 and 2, the following lifted interdiction cut is valid for (31)–(35):*

$$w \geq \sum_{j \in R} d_j \hat{y}_j + \sum_{j \in N} d_j \hat{y}_j (1 - x_j) + \sum_{k=1}^{K} (d_{b_k} - d_{a_k})(1 - x_{b_k}). \tag{48}$$

**Proof.** We have to show that (48) is satisfied by any given feasible interdiction policy $x$. In case $x_{b_k} = 1$ for each $k = 1, \ldots, K$, this is obvious, because $x$ satisfies the interdiction inequality (43). Otherwise, denote by $\bar{\mathcal{K}} = \{k \in \{1, \ldots, K\} : x_{b_k} = 0\}$ the subset of indices associated with items in $S_b$ that are not selected in policy $x$.

Consider the alternative follower solution $y'$ obtained from $\hat{y}$ by flipping, for each $k \in \bar{\mathcal{K}}$, $\hat{y}_{a_k}$ and $\hat{y}_{b_k}$ (i.e., by setting $y'_{a_k} = 0$ and $y'_{b_k} = 1$) and leaving the remaining entries unchanged. Under the assumption $Q_{a_k} \geq Q_{b_k}$, one has $Qy' \leq Q_0$ (i.e., $y' \in \hat{Y}$); hence, $x$ satisfies the interdiction inequality associated with $y'$, namely

$$\begin{aligned} w \geq & \sum_{j\in R} d_j y'_j + \sum_{j\in N} d_j y'_j (1 - x_j) = \\ = & \sum_{j\in R} d_j \underbrace{y'_j}_{=\hat{y}_j} + \sum_{j\in N\setminus\{a_k, b_k : k\in\bar{\mathcal{K}}\}} d_j \underbrace{y'_j}_{=\hat{y}_j} (1 - x_j) \\ & + \sum_{k\in\bar{\mathcal{K}}} \left( d_{a_k} \underbrace{y'_{a_k}}_{=0} (1 - x_{a_k}) + d_{b_k} \underbrace{y'_{b_k}}_{=1} (1 - x_{b_k}) \right). \end{aligned} \tag{49}$$

Rewrite (48) in a similar way to obtain

$$\begin{aligned} w \geq & \sum_{j\in R} d_j \hat{y}_j + \sum_{j\in N\setminus\{a_k, b_k : k\in\bar{\mathcal{K}}\}} d_j \hat{y}_j (1 - x_j) \\ & + \sum_{k\in\bar{\mathcal{K}}} \left( d_{a_k} \underbrace{\hat{y}_{a_k}}_{=1} (1 - x_{a_k}) + d_{b_k} \underbrace{\hat{y}_{b_k}}_{=0} (1 - x_{b_k}) \right. \\ & \left. + (d_{b_k} - d_{a_k})(1 - x_{b_k}) \right) + \sum_{k\in\mathcal{K}\setminus\bar{\mathcal{K}}} (d_{b_k} - d_{a_k}) \underbrace{(1 - x_{b_k})}_{=0}. \end{aligned} \tag{50}$$

Because (49) is a valid inequality and the left-hand sides of both (49) and (50) are the same, it remains to be shown that the right-hand side of (50) is smaller than or equal to the right-hand side of (49). To this end, subtract the right-hand side of (50) from the right-hand side of (49) to obtain

$$\sum_{k\in\bar{\mathcal{K}}} \left( d_{b_k}(1 - \underbrace{x_{b_k}}_{=0}) - d_{a_k}(1 - x_{a_k}) - (d_{b_k} - d_{a_k})(1 - \underbrace{x_{b_k}}_{=0}) \right). \tag{51}$$

For each $k \in \bar{\mathcal{K}}$, the corresponding term in (51) is zero if $x_{a_k} = 0$, whereas it is equal to $d_{a_k} > 0$ if $x_{a_k} = 1$. Thus, the sum is nonnegative, which concludes the proof. □

Notice that items in $S_a$ and $S_b$ may often be paired in different ways, still satisfying the requirements of the theorem above; thus, they produce different lifted inequalities. Our specific recipe for their separation will be provided in the next section.

Finally, the following theorem introduces a new family of valid inequalities that exploits dominance relationships between pairs of items.

**Theorem 7.** *Let $i, s \in N$ be two distinct items, such that $A_i \leq A_s$, $Q_i \leq Q_s$, and $d_i \geq d_s$. Under Assumption 1 and 2, the following dominance inequality*

$$x_s \leq x_i \tag{52}$$

*is satisfied by at least one optimal solution to problem (31)–(35).*

**Proof.** We provide a constructive proof of the existence of an optimal solution of (31)–(35) that is not cut off by (52). Let $x^*$ be an optimal solution that violates (52) (if any; i.e., such that $x^*_s = 1$ while $x^*_i = 0$). Define an alternative leader solution $x'$ obtained from $x^*$ by flipping its components indexed by $\{s, i\}$, namely

$$x'_j = \begin{cases} x^*_j, & j \notin \{s, i\} \\ 0, & j = s \\ 1, & j = i \end{cases} \qquad j \in \{1, \ldots, n_1\}.$$

Solution $x'$ clearly satisfies (52) and is feasible because of assumption $A_i \leq A_s$. It remains to be shown that $x'$ is also optimal for (31)–(35) (i.e., that $\Phi(x') \leq \Phi(x^*)$). To this end, let $y' = y^*(x')$ denote an optimal follower solution for $x'$ (where $y'_i = 0$ as $x'_i = 1$), and define an alternative follower solution $\hat{y}$ obtained from $y'$ by flipping its entries indexed by $\{s, i\}$ in case $y'_s = 1$, while $\hat{y} = y'$ otherwise. By definition, one has $\hat{y}_s = 0$ in both cases. In addition, because of assumption $Q_i \leq Q_s$, $\hat{y}$ is a feasible follower solution for $x^*$; hence,

$$\Phi(x') = d^T y' \leq d^T \hat{y} \leq \Phi(x^*),$$

where the first inequality follows from assumption $d_s \leq d_i$. □

It is worth noting that there are only $O(|N|^2)$ dominance inequalities, and therefore, they can be statically added to the original model formulation without the need to design a runtime separation procedure. To avoid dominance loops, in case items $i$ and $s$ are identical (i.e., $A_i = A_s$, $Q_i = Q_s$, and $d_i = d_s$), we skip one of two inequalities—namely, that for $i < s$.

## 4. A B&C Approach for Monotone IGs

We have designed a B&C approach that works in the $(w, x)$ space and dynamically adds the cuts described in the previous section. We next give implementation details about our approach.

### 4.1. Separation of Interdiction Cuts

Let $(w^*, x^*)$ be the solution of the LP relaxation at a B&C node. The separation problem for (43) consists of solving the following problem:

$$\max\left\{ \sum_{j\in R} d_j y_j + \sum_{j\in N} d^*_j y_j : y \in Y \right\}, \tag{53}$$

where $d^*_j = d_j(1 - x^*_j)$ for all $j \in N$. Let $z^*$ be the optimal solution value of such a problem, and let $y^*$ be the solution found. If $w^* < z^*$, then $y^*$ gives a maximally violated interdiction cut (43); otherwise, no violated cut exists.

Note that entries $x^*_j = 1$ produce zero coefficients $d^*_j$ in the objective function of the separation problem (53), possibly yielding an optimal solution $y^*$ that is nonmaximal. In this case, there could be some other

$y'_N \neq y^*_N$ with $y'_N \geq y^*_N$ and $y'_R = y^*_R$, which is an alternative optimal solution of the separation problem. According to Theorem 4, the interdiction cut associated with $y'$ dominates the one associated with $y^*$. Thus, to favor maximal solutions, in our implementation we actually solve separation problem (53) with a perturbed objective function $\sum_{j\in N} d^*_j y_j$, where each $d^*_j = 0$ with $j \in N$ is replaced by $\epsilon\, d_j$ for a very small $\epsilon > 0$ ($\epsilon = 0.001$ was used).

In the case that the follower is a single-(integer) knapsack problem, the separation problem can be solved using the well-known dynamic programming algorithm for knapsack problems (e.g., Martello and Toth 1990) running in pseudopolynomial time. Otherwise, the separation problem is solved using a general purpose MILP solver. In both cases, separation is an NP-hard problem, which can make exact separation time consuming. However, the correctness of our approach requires us to apply exact separation of interdiction cuts (43) only in the case that $x^*$ is integer. For fractional $x^*$, to speed up execution, we heuristically solve the separation problem as follows. If the follower subproblem is a single-knapsack problem, a simple greedy heuristic is applied (Martello and Toth 1990): items are ordered according to nonincreasing values of $d^*_i/q_{i0}$, and a solution is constructed by collecting items until no more fit into the knapsack. In the case that the follower subproblem involves multiple constraints, a general purpose MILP solver is used, and the run is interrupted after the root node is finished (if no feasible solution is found, no cut is added).

### 4.2. Separation of Modified Interdiction Cuts

We have implemented a heuristic to separate modified interdiction cuts (44). The heuristic takes on input that the (possibly nonviolated) interdiction cut produced by the separation routine described in Section 4.1 and tries to modify it to obtain a violated cut (44) in a greedy way. At each iteration, the next item that is a candidate to enter set $S_a$ is the item $a \in N$ with $\hat{y}_a = 1$ and maximum $d_a$. Given $a$, its "twin" item $b$ is selected among those with $\hat{y}_b = 0$ and $Q_a \geq Q_b$ as the one with largest value $d_b(x^*_a - x^*_b)$: if $d_b(x^*_a - x^*_b) > 0$, items $a$ and $b$ are inserted into sets $S_a$ and $S_b$, respectively, and then removed from any further consideration.

### 4.3. Separation of Lifted Interdiction Cuts

In our algorithm, lifted interdiction cuts are separated in a heuristic way as well. The separation procedure is very similar to the one described in the previous subsection to obtain a modified interdiction cut. Given an interdiction cut (43), we heuristically try to lift it to an inequality (48) in a greedy way. We consider the items $a \in N$, such that $\hat{y}_a = 1$ (that are the only candidate to be included in $S_a$), according to nonincreasing $d_a$ values. For each such item $a$, every item $b$ with $\hat{y}_b = 0$ is checked for creating a possible lifting pair $(a, b)$. More precisely, we scan all such items $b$ with $\hat{y}_b = 0$, $d_b > d_a$, and $Q_a \geq Q_b$ (if any) and pick the one with minimum $d^*_j = d_j(1 - x^*_j)$ value. If such an item pair $(a, b)$ is found, items $a$ and $b$ are inserted into sets $S_a$ and $S_b$, respectively, and then removed from any further consideration. In preliminary computational tests, we experimented with alternative procedures for selecting the item pairs to lift, but the simple heuristic above turned out to be the most effective.

### 4.4. A Heuristic for General IGs

We next introduce a quite general heuristic for (possibly nonmonotone) IGs, which is based on the idea of adding invalid leader constraints on the $x$ variables that allow the optimal follower solution be expressed analytically as an a priori linear function of $x$.

To be more specific, let $N^+ = \{j \in N : d_j > 0\}$ (recall that $d_j$ can be nonpositive in the nonmonotone case), and assume that $R = \emptyset$ (i.e., all follower variables $y_j$ appear in a constraint (3)). We introduce the invalid leader constraints

$$\sum_{j\in N^+} Q_j u_j (1 - x_j) \leq Q_0, \tag{54}$$

stipulating that all of the noninterdicted items (those with $x_j = 0$) with positive profit $d_j$ can be selected by the follower (at their highest possible level) in a feasible solution. As a consequence, an optimal follower solution $y^*(x)$ always exists with

$$y^*_j(x) = \begin{cases} u_j(1 - x_j), & \text{if } j \in N^+, \\ 0, & \text{otherwise.} \end{cases}$$

The restricted IG (i.e., problem (31)–(35) with the addition of constraints (54)) can, therefore, be reformulated exactly as the following (compact) single-level MILP:

$$(HEU\_REF) \quad \min \sum_{j\in N^+} d_j u_j (1 - x_j) \tag{55}$$

$$Ax \leq b \tag{56}$$

$$\sum_{j\in N^+} Q_j u_j (1 - x_j) \leq Q_0 \tag{57}$$

$$x_j \text{ integer}, \quad \forall j \in J_x \tag{58}$$

$$x_j \in \{0, 1\}, \quad \forall j \in N, \tag{59}$$

where the $y$ variables have been projected out as their optimal value is known. The above MILP can of course be infeasible. If this is not the case, its optimal solution provides a valid upper bound $UB$ (say) for the original IG owing to the obvious fact that the feasible solution set of (55)–(59) is a subset of that of the original problem (31)–(35) caused by the addition of constraints (54).

If a finite *UB* is obtained, one can modify the original model (31)–(35) by adding the objective cutoff constraint

$$w \le UB - \epsilon \tag{60}$$

for a sufficiently small $\epsilon > 0$ ($\epsilon = 1$ in the case of integer $d$). In addition, one can impose a disjunction stating that at least one of the constraints in (54) must be violated. In our implementation, this is done through the following (possibly weak) single linear constraint

$$\sum_{j \in N^+} \max_i \{q_{ij}\}\, u_j(1 - x_j) \ge \min_i \{q_{i0} + \epsilon\}, \tag{61}$$

where in case $Q_0 > 0$, the single inequalities have been normalized to get $q_{i0} = 1$ for all $i$.

According to our computational experience, the addition of constraints (60) and (61) to the original model (31)–(35) often reduces solution time in a very significant way. This is true, in particular, when the resulting problem turns out to be infeasible, meaning that one is able to quickly prove that *UB* gives an optimal solution of the original IG as well.

## 5. Computational Results

To assess the efficiency of our approach, we implemented it in Python using the commercial solver IBM ILOG CPLEX 12.6 as underlying B&C framework. All CPLEX parameters were left at their default values in our runs, and a time limit of 3,600 seconds for each run was set. The runs were made in sequential (single-thread) mode on an Intel Xeon E3-1220V2 @3.1-GHz computer with 3 GB of RAM.

### 5.1. Benchmark

We tested our approach on instances from the literature for the KIP instances as well as on new instances with multiple leader and/or follower constraints based on multidimensional knapsack interdiction problem (MKIP) instances. We decided instead not to address the other applications mentioned in Section 2, for which we expect that a customized code (still based on our interdiction cuts but using, for example, a specialized preprocessing and a runtime separation procedure for subtour elimination constraints) is required to get the best-possible performance.

**5.1.1. KIP Instances from the Literature.** Our first data set includes the following 360 KIP instances from literature.

- Instance set CCLW has been introduced in Caprara et al. (2016). The follower data have been created using the knapsack instance generator of Martello et al. (1999); profits $d_i$ and weights $q_i$ are uncorrelated integers in range $[0, 100]$, and the follower budget is set to $q_0 = \lceil \frac{INS}{10} \sum_{i \in N} w_i \rceil$, where *INS* is the number of the instance with $1 \le INS \le 10$. The leader coefficients $a_i$ are integers chosen uniformly random in $[0, 100]$, whereas the leader budget $a_0$ is taken from $[q_0 - 10, q_0 + 10]$. Ten instances are created for $|N| \in \{35, 40, 45, 50, 55\}$ for a total of 50 instances.
- Instance set TRS has been proposed by Tang et al. (2015). The interdiction budget is a cardinality constraint allowing at most $k$ items to be interdicted. Item weights and profits are random integers from $[1, 100]$. Ten instances for pairs $(|N|, k)$ with $|N| \in \{20, 22, 25, 28, 30\}$ and three different values of $k$ have been constructed for a total of 150 instances.
- Instance set D has been introduced in DeNegre (2011). This class is based on bicriteria knapsack instances from the *multiple criteria decision-making library*: the first objective of the bicriteria problem is used to define the follower objective function, whereas the second objective defines the interdiction budget constraint of the leader. The interdiction budget of an instance is $\lceil \sum_{i=1}^{n_1} a_i/2 \rceil$, where $a_i$ is the cost of interdicting item $i$. The instances have $|N| \in \{10, 20, \ldots 50\}$, with two additional sets with 11 and 12 items. For every number of items, there are 20 instances, except for the 10-item case, for which there are 40 instances. Thus, there are 160 instances in this class.

**5.1.2. MKIP Instances from SAC-94 Library (Khuri et al. 1994).** The SAC-94 library (Khuri et al. 1994) is a benchmark library containing 0/1 Multidimensional Knapsack Instances from Freville and Plateau (1990) (instances hp* and pb*), Petersen (1967) (instances pet*), Senju and Toyoda (1968) (instances sento*), Shih (1979) (instances weish*), and Weingartner and Ness (1967) (instances weing*). Starting with these 54 instances, we generated 144 new instances of the MKIP as follows.

The instances have 2–30 constraints and 10–90 items. For each instance of this data set, we constructed three different interdiction instances by considering

- the first constraint as the leader constraint and the remaining constraints as follower constraints (these instances are denoted by –0 in the name);
- the first 50% of constraints (rounded up) as leader constraints and the remaining ones as follower constraints (denoted by –50); and
- all but the last constraint as leader constraints (denoted by -100).

Thus, in the -0 and -50 instances, the follower problem is a multidimensional knapsack problem, whereas instances of type -100 have a single knapsack as follower problem. Moreover, in instances of type -50 and -100, there are multiple leader constraints. Of course, when the underlying multidimensional knapsack instances have just two constraints, all three transformations give the same instance with one leader and one follower constraint (i.e., a single knapsack as follower problem). These instances are weing* and instance pb4.

Thus, we obtained 54 instances of type -100 and 45 instances of type -0 and -50 for a total of 144 instances. Details on the numbers of variables and leader/follower constraints and on the obtained optimal solution are presented in Tables 4–6. All instances are available online at https://msinnl.github.io/pages/instancescodes.html.

## 5.2. Analyzing the Influence of the Individual Ingredients

To assess the influence of the various ingredients proposed in our framework, we tested six different settings of our B&C code.

-: this is our basic setting, in which only basic interdiction cuts (43) are separated using the exact algorithm;

M: as before, with the addition of the heuristic separation for modified interdiction cuts described in Section 4.2;

MH: as before but using the heuristic separation procedure for interdiction cuts (43) described in Section 4.1 (instead of the exact separation algorithm);

MHD: as before but all dominance inequalities (52) are statically added to the initial model;

MHDL: as before but instead of adding the basic interdiction cut associated with a heuristic follower solution $\hat{y}$, we perform the lifting procedure described in Section 4.3 to $\hat{y}$ and only generate the associated lifted interdiction cut; and

MHDLP: as before but a preprocessing step is applied that invokes the heuristic of Section 4.4 and possibly adds the associated invalid cuts (60) and (61) to the model formulation.

In all settings, only maximal follower solutions are considered for separation. Furthermore, both fractional and integer solutions are separated. Observe that, by construction, each execution of the separation algorithm returns (at most) a single violated (lifted) interdiction cut (possibly plus a modified interdiction cut); hence, we did not impose any limit on the number of generated cuts at each separation call.

Figure 2 plots the root node gap and the runtime to optimality for the KIP instances from literature, whereas Figure 3 gives the same information for MKIP instances. The root gap is calculated as $100 \cdot (BestObj - RootBound)/(10^{-10} + |BestObj|)$, where *BestObj* is the best objective value found by all settings and *RootBound* is the root node lower bound produced by the setting. Observe that setting MHDLP may prove optimality of the heuristic solution *UB* by proving infeasibility of the problem after preprocessing; in case such infeasibility is already proven at the root node, we report a gap of zero.

Both Figures 2(a) and 3(a) show that using the modified interdiction cuts slightly improves the root gap, whereas heuristic separation provides a root gap that is similar to that obtained using an exact procedure for separation. (Actually, for some MKIP instances, the gap with heuristic separation is even slightly better—this can be explained by the fact the CPLEX additionally generates internal MILP cuts that may affect the final bound in an unpredictable way.)

For the KIP instances from the literature, a clear effect on the bounds can be observed when using the dominance inequalities (52) (setting MHD), resulting in about 170 instances solved to optimality at the root node to be compared with about 100 for settings - and H. Additionally, using lifting (setting MHDL) and the preprocessing heuristic (setting MHDLP) improves the gap even further so that about 220 instances can be solved to optimality at the root node. A similar trend, however less pronounced, can be observed for the MKIP instances. For both classes of instances, the gap at the root node for more than two of three of the instances is below 10% when using settings MHDL∗.

Turning our attention to the runtime to optimality (i.e., to Figures 2(b) and 3(b)), we see that the noticeable difference in root gap between the settings does not directly translate into a similar difference in runtime for the KIP instances. In particular, although setting M slightly improves the root gap, the runtime to optimality is nearly identical to the basic setting -. However, additionally using the heuristic separation (H) gives a big improvement in runtime. The explanation of this behavior is that a much higher node throughput can be achieved in branch and bound when using heuristic separation, whereas slightly improved bounds (M) may not be crucial to quickly solve an instance to optimality. For MKIP, this effect is less pronounced, which is because of the fact that the greedy heuristic used in the KIP case is more efficient than the MILP-based heuristic for the MKIP case.

Dominance inequalities and lifted interdiction cuts are both very important ingredients for harder instances. For example, for KIP instances, when using MH, the most difficult instance takes about 1,000 seconds, whereas setting MHD drastically reduces runtime to about 160 seconds and MHDL* reduces it to about 80 seconds. Finally, using the primal heuristic in a preprocessing step as described in Section 4.4 further improves performance, especially for easy instances. This improvement may be crucial if such problems have to be solved in a real-time setting.

We conclude by observing that the setting where all ingredients of our framework are used, namely MHDLP, gives the best overall performance. In particular, it solves all KIP instances from the literature in at most 84 seconds (the most challenging problem being instance 55-3 of set CCLW; see the next section), and only four of 360 instances took more than 10 seconds.

In view of the above, MHDLP is chosen as our default setting and will be simply denoted by B&C in what follows.

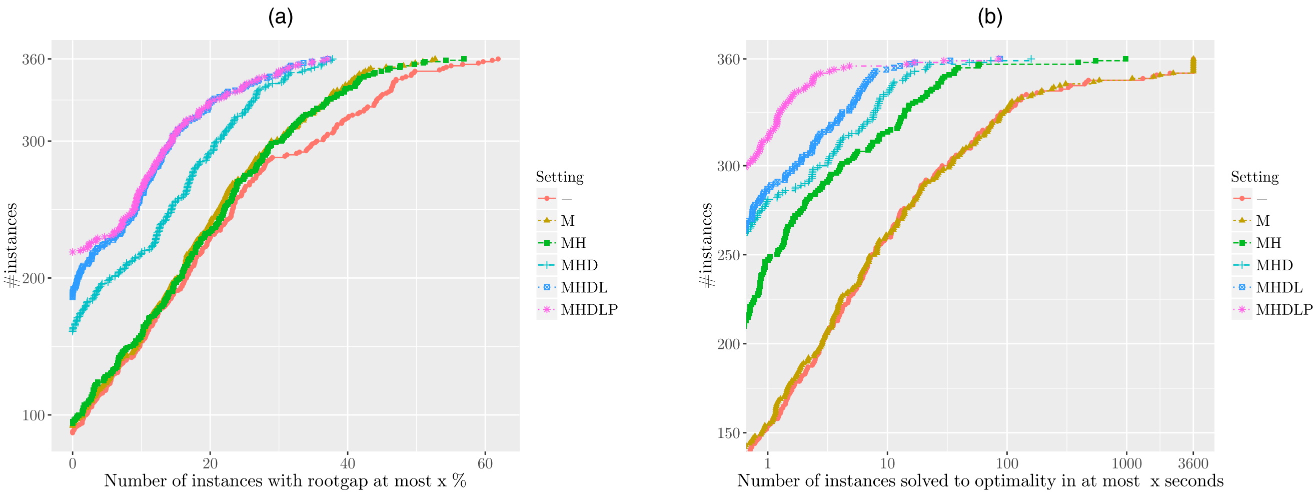
## 5.3. Results for Instance Set CCLW

Table 1 gives a comparison of the results achieved by B&C (i.e., by our approach using its most advanced

INFORMS Journal on Computing, 2019, vol. 31, no. 2, pp. 390–410, © 2019 INFORMS



**Figure 2.** (Color online) Root Gap and Runtime to Optimality for the KIP Instances from Literature and for Different Settings



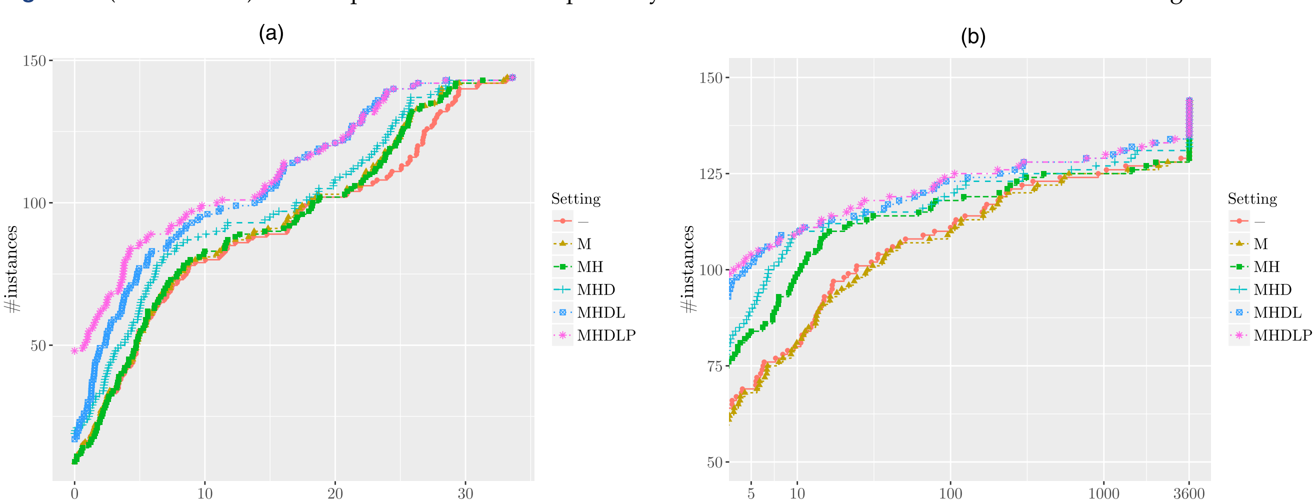

*Notes.* (a) Root gap. (b) Runtime to optimality.

setting MHDLP) with those of the integer cutting-plane approach using interdiction cuts (column CP) and the specialized CCLW algorithm, both presented in Caprara et al. (2016). The results for CP and CCLW in Caprara et al. (2016) have been obtained on a four-core Intel Xeon @2.6 GHz. Column $z^*$ gives the optimal solution value, whereas the remaining columns provide the runtime to optimality (in seconds) for the respective approaches. Entries TL in this column indicate runs for which the time limit of 3,600 seconds has been reached. It may be observed that many instances of this data set are very easy for both CCLW and B&C, and they are solved in around one second of computing time, whereas they are much harder for CP.

Recalling that the instances of set CCLW are constructed in such a way that a larger instance number means larger budget (for leader and follower, because the two budgets are set in a correlated way), one can observe that there is a peak of difficulty for CCLW for instances numbered three and four for all sizes. For B&C, this can only be observed for the largest set with 55 items.

Turning our attention to the hardest instances of the set, we see that B&C outperforms CCLW by up to three orders of magnitudes. Notably, B&C finds the optimal solution for the two unsolved instances 55-3 and 55-4 in just 84 and 16 seconds, respectively. Moreover, B&C solves instance 50-2 in just two seconds, whereas CCLW takes as long as 1,520 seconds.

**Figure 3.** (Color online) Root Gap and Runtime to Optimality for the MKIP Instances and for Different Settings




*Notes.* (a) Root gap. (b) Runtime to optimality.

**Table 1.** Runtime to Optimality, in Seconds, for Our Approach (B&C) vs. the Cutting Plane (CP) and CCLW Approaches from Caprara et al. (2016)

| Size and instance | $z^*$ | CP | CCLW | B&C |
|---|---|---|---|---|
| 35 | | | | |
| 1 | 279 | 0.34 | 0.79 | 0.12 |
| 2 | 469 | 1.59 | 2.57 | 0.21 |
| 3 | 448 | 55.61 | 40.39 | 0.66 |
| 4 | 370 | 495.50 | 1.48 | 0.87 |
| 5 | 467 | TL | 0.72 | 0.93 |
| 6 | 268 | 71.43 | 0.06 | 0.11 |
| 7 | 207 | 144.46 | 0.06 | 0.07 |
| 8 | 41 | 0.50 | 0.04 | 0.07 |
| 9 | 80 | 0.97 | 0.03 | 0.07 |
| 10 | 31 | 0.12 | 0.03 | 0.08 |
| 40 | | | | |
| 1 | 314 | 0.66 | 1.06 | 0.16 |
| 2 | 472 | 6.67 | 7.50 | 0.36 |
| 3 | 637 | 324.61 | 162.80 | 1.02 |
| 4 | 388 | 1,900.03 | 0.34 | 0.82 |
| 5 | 461 | TL | 0.22 | 0.58 |
| 6 | 399 | 2,111.85 | 0.09 | 0.13 |
| 7 | 150 | 83.59 | 0.05 | 0.08 |
| 8 | 71 | 1.73 | 0.04 | 0.09 |
| 9 | 179 | 137.16 | 0.08 | 0.09 |
| 10 | 0 | 0.03 | 0.03 | 0.04 |
| 45 | | | | |
| 1 | 427 | 1.81 | 2.37 | 0.23 |
| 2 | 633 | 13.03 | 11.64 | 0.37 |
| 3 | 548 | TL | 344.01 | 1.81 |
| 4 | 611 | TL | 38.90 | 3.30 |
| 5 | 629 | TL | 3.42 | 2.78 |
| 6 | 398 | 3,300.76 | 0.07 | 0.17 |
| 7 | 225 | 60.43 | 0.04 | 0.09 |
| 8 | 157 | 60.88 | 0.05 | 0.10 |
| 9 | 53 | 0.83 | 0.05 | 0.10 |
| 10 | 110 | 0.40 | 0.05 | 0.11 |
| 50 | | | | |
| 1 | 502 | 2.86 | 4.55 | 0.21 |
| 2 | 788 | 1,529.16 | 1520.56 | 2.38 |
| 3 | 631 | TL | 105.59 | 2.40 |
| 4 | 612 | TL | 3.64 | 1.27 |
| 5 | 764 | TL | 0.60 | 4.82 |
| 6 | 303 | 1,046.85 | 0.05 | 0.14 |
| 7 | 310 | 2,037.01 | 0.09 | 0.11 |
| 8 | 63 | 2.79 | 0.05 | 0.12 |
| 9 | 234 | 564.97 | 0.10 | 0.12 |
| 10 | 15 | 0.09 | 0.04 | 0.13 |
| 55 | | | | |
| 1 | 480 | TL | 18.57 | 0.46 |
| 2 | 702 | TL | 443.53 | 1.50 |
| 3 | 778 | TL | TL | 84.83 |
| 4 | 889 | TL | TL | 16.75 |
| 5 | 726 | TL | 0.24 | 1.36 |
| 6 | 462 | TL | 0.09 | 0.16 |
| 7 | 370 | TL | 0.08 | 0.12 |
| 8 | 387 | TL | 0.10 | 0.13 |
| 9 | 104 | TL | 0.06 | 0.13 |
| 10 | 178 | TL | 0.06 | 0.14 |

### 5.4. Results for Instance Set TRS

Table 2 gives the results for instance set TRS. We compare the results of our B&C with the results obtained by the best-performing approach presented in Tang et al. (2015) (columns TRS), where this data set has been proposed. We also benchmark our results against the best-performing setting of a state-of-the-art general purpose bilevel mixed integer programming solver, namely the exact approach presented in Fischetti et al. (2017a) (see column MIX++ in Table 2). The results of Tang et al. (2015) have been obtained on "a PC with 3.30 GHz using CPLEX 12.5," whereas the results of Fischetti et al. (2017a) have been obtained with four-thread runs on the same machine that we used for the runs in this paper. Results are given as averages over the 10 instances per each $(|N|, k)$ pair. For each approach, column $t[s]$ reports runtime (in seconds); for TRS, we also provide the number of instances that were not solved to optimality within the time limit of one hour (column $N^*$ in Table 2).

We observe that, for all $(|N|, k)$ pairs, our approach needs an average runtime of at most 0.3 seconds for computing a provably optimal solution. These computing times are smaller than those for MIX++ by up to three orders of magnitude. Furthermore, most of the instances with 22 or more items were unsolved by the approach of Tang et al. (2015) within one hour of computing time, whereas all of them are just trivial for our algorithm. Finally, note that all approaches except B&C are very sensitive to the value of $k$ (i.e., to the number of items that can be interdicted): the most challenging instances for MIX++ are those with small $k$ values, whereas medium values of $k$ produce the hardest instances for the approach by Tang et al. (2015).

**Table 2.** Results for Instance Set TRS Compared with Results Obtained by the Best Algorithm Presented in Tang et al. (2015) (TRS) and the State-of-the-Art General Purpose Bilevel Solver Presented in Fischetti et al. (2017a) (MIX++)

| | | TRS | | | |
|---|---|---|---|---|---|
| $\|N\|$ | $k$ | $t$ (s) | $N^*$ | MIX++ $t$ (s) | B&C $t$ (s) |
| 20 | 5 | 721.4 | 0 | 5.4 | 0.1 |
| 20 | 10 | 2,992.6 | 3 | 1.7 | 0.1 |
| 20 | 15 | 129.5 | 0 | 0.2 | 0.1 |
| 22 | 6 | 1,281.2 | 6 | 10.3 | 0.1 |
| 22 | 11 | 3,601.8 | 10 | 2.3 | 0.1 |
| 22 | 17 | 248.2 | 0 | 0.2 | 0.1 |
| 25 | 7 | 3,601.4 | 10 | 33.6 | 0.2 |
| 25 | 13 | 3,602.3 | 10 | 8.0 | 0.2 |
| 25 | 19 | 1,174.6 | 0 | 0.4 | 0.1 |
| 28 | 7 | 3,601.0 | 10 | 97.9 | 0.3 |
| 28 | 14 | 3,602.5 | 10 | 22.6 | 0.3 |
| 28 | 21 | 3,496.9 | 8 | 0.5 | 0.1 |
| 30 | 8 | 3,601.0 | 10 | 303.0 | 0.3 |
| 30 | 15 | 3,602.3 | 10 | 31.8 | 0.3 |
| 30 | 23 | 3,604.5 | 10 | 0.6 | 0.1 |

*Notes.* Every row reports average results over 10 instances. $N^*$ gives the number of instances not solved to proven optimality by TRS. $t$ (s), runtime in seconds.

**Table 3.** Results for Instance Set D Compared with Results Obtained by the State-of-the-Art General Purpose Bilevel Solver Presented in Fischetti et al. (2017a)

| | MIX++ from Fischetti et al. (2017a) | | | | B&C | | | |
|---|---|---|---|---|---|---|---|---|
| Instance | BestSol | LB | %Gap | $t$ (s) | BestSol | LB | %Gap | $t$ (s) |
| K5030W01 | 2,956 | 2,956.000 | 0.00 | 39.88 | 2,956 | CUTOFF | 0.00 | 0.72 |
| K5030W02 | 3,529 | 3,529.000 | 0.00 | 86.77 | 3,529 | CUTOFF | 0.00 | 0.59 |
| K5030W03 | 2,706 | 2,706.000 | 0.00 | 43.01 | 2,706 | CUTOFF | 0.00 | 0.61 |
| K5030W04 | 3,201 | 3,201.000 | 0.00 | 73.32 | 3,201 | CUTOFF | 0.00 | 0.65 |
| K5030W05 | 4,861 | 4,861.000 | 0.00 | 569.09 | 4,861 | CUTOFF | 0.00 | 2.37 |
| K5030W06 | 1,997 | 1,997.000 | 0.00 | 12.12 | 1,997 | CUTOFF | 0.00 | 0.47 |
| K5030W07 | 2,270 | 2,270.000 | 0.00 | 18.99 | 2,270 | CUTOFF | 0.00 | 0.45 |
| K5030W08 | 4,902 | 4,902.000 | 0.00 | 1,077.40 | 4,902 | 4,902.000 | 0.00 | 3.58 |
| K5030W09 | 2,201 | 2,201.000 | 0.00 | 14.06 | 2,201 | CUTOFF | 0.00 | 0.50 |
| K5030W10 | 2,668 | 2,668.000 | 0.00 | 19.00 | 2,668 | CUTOFF | 0.00 | 0.75 |
| K5030W11 | 2,013 | 2,013.000 | 0.00 | 28.67 | 2,013 | CUTOFF | 0.00 | 0.50 |
| K5030W12 | 2,534 | 2,534.000 | 0.00 | 11.42 | 2,534 | CUTOFF | 0.00 | 0.33 |
| K5030W13 | 3,152 | 3,152.000 | 0.00 | 21.57 | 3,152 | CUTOFF | 0.00 | 0.53 |
| K5030W14 | 2,184 | 2,184.000 | 0.00 | 23.05 | 2,184 | CUTOFF | 0.00 | 0.43 |
| K5030W15 | 2,841 | 2,841.000 | 0.00 | 53.60 | 2,841 | CUTOFF | 0.00 | 0.58 |
| K5030W16 | 2,102 | 2,102.000 | 0.00 | 12.57 | 2,102 | CUTOFF | 0.00 | 0.47 |
| K5030W17 | 3,553 | 3,553.000 | 0.00 | 98.74 | 3,553 | CUTOFF | 0.00 | 0.50 |
| K5030W18 | 2,602 | 2,602.000 | 0.00 | 19.66 | 2,602 | CUTOFF | 0.00 | 0.50 |
| K5030W19 | 5,015 | 5,015.000 | 0.00 | 710.57 | 5,015 | CUTOFF | 0.00 | 2.44 |
| K5030W20 | 2,496 | 2,496.000 | 0.00 | 11.95 | 2,496 | 2,496.000 | 0.00 | 0.79 |
| K5040W01 | 4,254 | 3,204.000 | 24.68 | TL | 4,254 | CUTOFF | 0.00 | 2.00 |
| K5040W02 | 4,423 | 4,423.000 | 0.00 | 2,533.36 | 4,423 | CUTOFF | 0.00 | 1.02 |
| K5040W03 | 3,440 | 3,440.000 | 0.00 | 1,578.91 | 3,440 | CUTOFF | 0.00 | 0.68 |
| K5040W04 | 3,574 | 3,574.000 | 0.00 | 1,158.20 | 3,574 | CUTOFF | 0.00 | 1.04 |
| K5040W05 | 4,646 | 3,363.630 | 27.60 | TL | 4,529 | CUTOFF | 0.00 | 1.32 |
| K5040W06 | 2,606 | 2,606.000 | 0.00 | 233.58 | 2,606 | CUTOFF | 0.00 | 0.99 |
| K5040W07 | 3,244 | 3,244.000 | 0.00 | 600.63 | 3,244 | CUTOFF | 0.00 | 1.11 |
| K5040W08 | 6,345 | 2,870.000 | 54.77 | TL | 6,174 | 6,173.558 | 0.00 | 14.44 |
| K5040W09 | 3,154 | 3,154.000 | 0.00 | 410.00 | 3,154 | CUTOFF | 0.00 | 0.57 |
| K5040W10 | 4,382 | 4,382.000 | 0.00 | 3,099.20 | 4,382 | CUTOFF | 0.00 | 1.41 |
| K5040W11 | 3,389 | 3,389.000 | 0.00 | 1,120.76 | 3,389 | CUTOFF | 0.00 | 0.81 |
| K5040W12 | 3,817 | 3,817.000 | 0.00 | 593.61 | 3,817 | CUTOFF | 0.00 | 0.56 |
| K5040W13 | 4,174 | 4,174.000 | 0.00 | 1,126.49 | 4,174 | CUTOFF | 0.00 | 0.85 |
| K5040W14 | 3,374 | 3,374.000 | 0.00 | 1,090.91 | 3,374 | CUTOFF | 0.00 | 0.75 |
| K5040W15 | 3,925 | 3,164.837 | 19.37 | TL | 3,925 | CUTOFF | 0.00 | 0.56 |
| K5040W16 | 2,605 | 2,605.000 | 0.00 | 194.39 | 2,605 | CUTOFF | 0.00 | 1.20 |
| K5040W17 | 3,996 | 3,996.000 | 0.00 | 2,645.77 | 3,996 | CUTOFF | 0.00 | 1.05 |
| K5040W18 | 3,342 | 3,342.000 | 0.00 | 918.78 | 3,342 | CUTOFF | 0.00 | 0.52 |
| K5040W19 | 5,299 | 3,167.000 | 40.23 | TL | 5,233 | CUTOFF | 0.00 | 1.68 |
| K5040W20 | 2,875 | 2,875.000 | 0.00 | 267.87 | 2,875 | CUTOFF | 0.00 | 0.89 |
| K5050W01 | 4,244 | 2,610.229 | 38.50 | TL | 4,189 | CUTOFF | 0.00 | 1.20 |
| K5050W02 | 5,280 | 2,559.000 | 51.53 | TL | 5,106 | CUTOFF | 0.00 | 1.20 |
| K5050W03 | 5,483 | 2,530.028 | 53.86 | TL | 4,769 | CUTOFF | 0.00 | 1.39 |
| K5050W04 | 3,999 | 2,401.000 | 39.96 | TL | 3,723 | CUTOFF | 0.00 | 1.20 |
| K5050W05 | 5,109 | 2,408.000 | 52.87 | TL | 4,998 | CUTOFF | 0.00 | 4.31 |
| K5050W06 | 3,558 | 2,691.930 | 24.34 | TL | 3,558 | CUTOFF | 0.00 | 1.55 |
| K5050W07 | 4,521 | 2,355.000 | 47.91 | TL | 4,390 | CUTOFF | 0.00 | 2.34 |
| K5050W08 | 8,215 | 2,706.517 | 67.05 | TL | 7,862 | 7,862.000 | 0.00 | 29.65 |
| K5050W09 | 4,775 | 2,521.000 | 47.20 | TL | 4,620 | CUTOFF | 0.00 | 1.21 |
| K5050W10 | 5,575 | 2,682.103 | 51.89 | TL | 5,047 | CUTOFF | 0.00 | 2.13 |
| K5050W11 | 3,855 | 2,287.000 | 40.67 | TL | 3,778 | CUTOFF | 0.00 | 1.63 |
| K5050W12 | 4,885 | 2,738.073 | 43.95 | TL | 4,562 | CUTOFF | 0.00 | 1.61 |
| K5050W13 | 4,926 | 2,816.000 | 42.83 | TL | 4,778 | CUTOFF | 0.00 | 1.27 |
| K5050W14 | 5,055 | 2,249.000 | 55.51 | TL | 4,544 | CUTOFF | 0.00 | 1.19 |
| K5050W15 | 4,757 | 2,240.701 | 52.90 | TL | 4,610 | CUTOFF | 0.00 | 1.17 |
| K5050W16 | 4,039 | 2,222.000 | 44.99 | TL | 3,979 | CUTOFF | 0.00 | 1.52 |

**Table 3.** (Continued)

| | MIX++ from Fischetti et al. (2017a) | | | | B&C | | | |
|---|---|---|---|---|---|---|---|---|
| Instance | BestSol | LB | %Gap | *t* (s) | BestSol | LB | %Gap | *t* (s) |
| K5050W17 | 5,666 | 2,672.093 | 52.84 | TL | 5,218 | CUTOFF | 0.00 | 1.24 |
| K5050W18 | 4,591 | 2,858.000 | 37.75 | TL | 4,591 | CUTOFF | 0.00 | 1.13 |
| K5050W19 | 6,022 | 2,717.526 | 54.87 | TL | 5,858 | CUTOFF | 0.00 | 2.06 |
| K5050W20 | 4,303 | 2,247.000 | 47.78 | TL | 4,303 | CUTOFF | 0.00 | 2.57 |

*Note.* BestSol, value of the best solution found; LB, best lower bound; %gap, associated optimality gap; *t* (s), runtime in seconds.

No dependency with respect to $k$ can instead be observed in B&C.

## 5.5. Results for Instance Set D

Table 3 gives results for instance set D. These instances have been introduced in DeNegre 2011, where computational results have only been presented for the smallest problems with at most 30 items. Because a much better general purpose bilevel solver has been recently proposed by Fischetti et al. (2017a), in Table 3, we compare only the best setting of Fischetti et al. (2017a) (namely, MIX++) with our own B&C solver. Table 3 reports the value of the best solution found (column BestSol in Table 3) and for each approach, the best lower bound, the associated optimality gap, and the runtime in seconds. For B&C, in case the heuristic solution obtained during preprocessing is optimal, we report CUTOFF in column LB in Table 3. We report only the results for the larger instances with 30 to 50 items, because the smaller instances with up to 20 items were solved to optimality by both approaches in less than 10 seconds (in most cases, in less than 1 second).

The table shows that our B&C gives a speedup of two to three orders of magnitude compared with MIX++ for most of the instances (note, however, that the latter solver, although better than any previous method on these instances, is not specialized for interdiction). The speedup becomes more pronounced as the number of items grows. Furthermore, none of the instances with 50 items could be solved by MIX++ within one hour, whereas B&C solves all instances except K5040W08 and K5050W08 within four seconds; however, for K5040W08 and K5050W08, it takes 14 and 29 seconds, respectively. Interestingly, all but four instances are solved right after preprocessing by proving infeasibility after the addition of the cutoff constraint.

## 5.6. Results for the MKIP Instances Based on the SAC-94 Library

Tables 4–6 compare the results obtained by B&C with the results obtained by the best setting of the general purpose bilevel solver presented in Fischetti et al. (2017a) (MIX++). Both solvers have been run on the same machine with a time limit of one hour, although MIX++ used four (instead of one) threads. As in Table 3, the tables report the value of the best solution found, the lower bound, the optimality gap, and the runtime in seconds. Additionally, the number of items ($|N|$), leader constraints, and follower constraints are given.

Table 4 reports results for instances of type -100 (i.e., with single-knapsack follower). We see that, depending on the underlying instance from which they have been created, they pose different difficulties to MIX++. For example, instances weing* are solved in less than three seconds (except weing8). Instances weish* are particularly hard for MIX++, more than one-half remaining unsolved within the time limit. Looking at hp* and pb* also reveals that the performance of MIX++ is highly influenced by the number of variables and constraints. However, our B&C approach manages to solve all instances to optimality in at most four seconds, thus greatly outperforming MIX++ for every instance. For about one-half of the instances, the heuristic solution obtained during preprocessing is the optimal one.

Table 5 addresses instances of type -50. For MIX++, they do not seem much more difficult than instances of type -100: one-half of the instances based on weish* cannot be solved within the time limit, and for the remaining ones, runtimes are similar to those of the associated instances of type -100. Thus, for MIX++, the underlying instance seems to have a bigger impact on runtime than the number of follower constraints. For our B&C approach, instead, these instances are more difficult than the ones of type -100. This is not too surprising, because these instances have a multidimensional knapsack as follower problem; thus, the preprocessing procedure is less effective. Moreover, the solution of the follower problem now consists of heuristically solving an MILP instead of a single-knapsack problem. In any case, our approach manages to solve all but three of the instances to optimality—in more than 50% of the cases within four seconds. Again, for about one-half of the instances, the solution found in the first phase of the heuristic is the optimal one.

The three unsolved instances (within the time limit of 3600 seconds) are weish22, weish27, and weish29. For these three instances, the gap is at most 1.15% compared with a gap of up to 73% for MIX++. We reran

**Table 4.** Results for Instance Set SAC Compared with Results Obtained by the State-of-the-Art General Purpose Bilevel Solver Presented in Fischetti et al. (2017a)

| | | | | MIX++ Fischetti et al. (2017a) | | | | B&C | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | \|N\| | #LC | #FC | BestSol | LB | %Gap | t (s) | BestSol | LB | %Gap | t (s) |
| hp1-100 | 28 | 3 | 1 | 1,536 | 1,536.000 | 0.00 | 13.06 | 1,536 | CUTOFF | 0.00 | 0.05 |
| hp2-100 | 35 | 3 | 1 | 3,015 | 3,015.000 | 0.00 | 304.14 | 3,015 | 3,015.000 | 0.00 | 2.55 |
| pb1-100 | 27 | 3 | 1 | 1,536 | 1,536.000 | 0.00 | 11.08 | 1,536 | CUTOFF | 0.00 | 0.04 |
| pb2-100 | 34 | 3 | 1 | 1,902 | 1,902.000 | 0.00 | 251.04 | 1,902 | CUTOFF | 0.00 | 0.10 |
| pb4-100 | 29 | 1 | 1 | 52,329 | 52,329.000 | 0.00 | 5.42 | 52,329 | CUTOFF | 0.00 | 0.04 |
| pb5-100 | 20 | 9 | 1 | 1,799 | 1,799.000 | 0.00 | 37.29 | 1,799 | 1,799.000 | 0.00 | 1.66 |
| pb6-100 | 40 | 29 | 1 | 1,389 | 1,389.000 | 0.00 | 250.25 | 1,389 | 1,389.000 | 0.00 | 0.85 |
| pb7-100 | 37 | 29 | 1 | 670 | 565.000 | 15.67 | TL | 656 | 656.000 | 0.00 | 0.59 |
| pet2-100 | 10 | 9 | 1 | 38,833 | 38,833.000 | 0.00 | 0.97 | 38,833 | CUTOFF | 0.00 | 0.03 |
| pet3-100 | 15 | 9 | 1 | 1,080 | 1,080.000 | 0.00 | 1.90 | 1,080 | CUTOFF | 0.00 | 0.04 |
| pet4-100 | 20 | 9 | 1 | 2,505 | 2,505.000 | 0.00 | 4.07 | 2,505 | CUTOFF | 0.00 | 0.09 |
| pet5-100 | 28 | 9 | 1 | 3,025 | 3,025.000 | 0.00 | 18.06 | 3,025 | CUTOFF | 0.00 | 0.06 |
| pet6-100 | 39 | 4 | 1 | 3,936 | 3,936.000 | 0.00 | 419.09 | 3,936 | CUTOFF | 0.00 | 0.07 |
| pet7-100 | 50 | 4 | 1 | 5,935 | 5,166.000 | 12.96 | TL | 5,723 | CUTOFF | 0.00 | 0.11 |
| sento1-100 | 60 | 29 | 1 | 1,686 | 1,225.000 | 27.34 | TL | 1,610 | 1,610.000 | 0.00 | 3.84 |
| sento2-100 | 60 | 29 | 1 | 752 | 457.000 | 39.23 | TL | 738 | CUTOFF | 0.00 | 0.46 |
| weing1-100 | 28 | 1 | 1 | 6,205 | 6,205.000 | 0.00 | 0.70 | 6,205 | CUTOFF | 0.00 | 0.05 |
| weing2-100 | 28 | 1 | 1 | 16,705 | 16,705.000 | 0.00 | 0.89 | 16,705 | CUTOFF | 0.00 | 0.05 |
| weing3-100 | 28 | 1 | 1 | 37,936 | 37,936.000 | 0.00 | 1.28 | 37,936 | 37,936.000 | 0.00 | 0.13 |
| weing4-100 | 28 | 1 | 1 | 42,958 | 42,958.000 | 0.00 | 0.65 | 42,958 | CUTOFF | 0.00 | 0.06 |
| weing5-100 | 28 | 1 | 1 | 6,205 | 6,205.000 | 0.00 | 0.52 | 6,205 | CUTOFF | 0.00 | 0.05 |
| weing6-100 | 28 | 1 | 1 | 8,103 | 8,103.000 | 0.00 | 0.60 | 8,103 | CUTOFF | 0.00 | 0.05 |
| weing7-100 | 105 | 1 | 1 | 15,646 | 15,646.000 | 0.00 | 2.70 | 15,646 | CUTOFF | 0.00 | 0.34 |
| weing8-100 | 105 | 1 | 1 | 212,854 | 212,854.000 | 0.00 | 151.36 | 212,854 | 212,854.000 | 0.00 | 1.31 |
| weish01-100 | 30 | 4 | 1 | 1,121 | 1,121.000 | 0.00 | 14.61 | 1,121 | 1,121.000 | 0.00 | 0.19 |
| weish02-100 | 30 | 4 | 1 | 1,293 | 1,293.000 | 0.00 | 16.76 | 1,293 | CUTOFF | 0.00 | 0.08 |
| weish03-100 | 30 | 4 | 1 | 1,601 | 1,601.000 | 0.00 | 10.12 | 1,601 | 1,601.000 | 0.00 | 0.20 |
| weish04-100 | 30 | 4 | 1 | 1,268 | 1,268.000 | 0.00 | 5.18 | 1,268 | CUTOFF | 0.00 | 0.13 |
| weish05-100 | 30 | 4 | 1 | 1,315 | 1,315.000 | 0.00 | 5.08 | 1,315 | CUTOFF | 0.00 | 0.11 |
| weish06-100 | 40 | 4 | 1 | 1,369 | 1,369.000 | 0.00 | 335.60 | 1,369 | CUTOFF | 0.00 | 0.22 |
| weish07-100 | 40 | 4 | 1 | 1,407 | 1,407.000 | 0.00 | 574.35 | 1,407 | CUTOFF | 0.00 | 0.12 |
| weish08-100 | 40 | 4 | 1 | 1,369 | 1,369.000 | 0.00 | 210.40 | 1,369 | CUTOFF | 0.00 | 0.12 |
| weish09-100 | 40 | 4 | 1 | 1,645 | 1,645.000 | 0.00 | 88.94 | 1,645 | 1,645.000 | 0.00 | 0.51 |
| weish10-100 | 50 | 4 | 1 | 2,146 | 2,146.000 | 0.00 | 809.66 | 2,146 | 2,146.000 | 0.00 | 0.43 |
| weish11-100 | 50 | 4 | 1 | 2,827 | 2,827.000 | 0.00 | 331.23 | 2,827 | 2,827.000 | 0.00 | 0.88 |
| weish12-100 | 50 | 4 | 1 | 2,146 | 2,146.000 | 0.00 | 621.03 | 2,146 | 2,146.000 | 0.00 | 0.38 |
| weish13-100 | 50 | 4 | 1 | 2,369 | 2,369.000 | 0.00 | 628.74 | 2,369 | 2,369.000 | 0.00 | 0.63 |
| weish14-100 | 60 | 4 | 1 | 2,648 | 1,825.110 | 31.08 | TL | 2,625 | 2,625.000 | 0.00 | 1.67 |
| weish15-100 | 60 | 4 | 1 | 2,138 | 1,759.000 | 17.73 | TL | 2,138 | CUTOFF | 0.00 | 0.20 |
| weish16-100 | 60 | 4 | 1 | 2,336 | 1,435.153 | 38.56 | TL | 2,285 | 2,285.000 | 0.00 | 1.21 |
| weish17-100 | 60 | 4 | 1 | 1,010 | 808.556 | 19.94 | TL | 991 | CUTOFF | 0.00 | 0.15 |
| weish18-100 | 70 | 4 | 1 | 1,986 | 1,348.885 | 32.08 | TL | 1,945 | CUTOFF | 0.00 | 0.19 |
| weish19-100 | 70 | 4 | 1 | 3,874 | 1,779.000 | 54.08 | TL | 3,741 | 3,740.887 | 0.00 | 1.80 |
| weish20-100 | 70 | 4 | 1 | 2,142 | 1,310.088 | 38.84 | TL | 2,075 | CUTOFF | 0.00 | 0.24 |
| weish21-100 | 70 | 4 | 1 | 2,535 | 1,453.784 | 42.65 | TL | 2,451 | CUTOFF | 0.00 | 0.24 |
| weish22-100 | 80 | 4 | 1 | 3,719 | 1,524.107 | 59.02 | TL | 3,325 | CUTOFF | 0.00 | 0.78 |
| weish23-100 | 80 | 4 | 1 | 4,177 | 1,602.804 | 61.63 | TL | 3,906 | 3,906.000 | 0.00 | 1.15 |
| weish24-100 | 80 | 4 | 1 | 2,190 | 1,277.619 | 41.66 | TL | 2,111 | CUTOFF | 0.00 | 0.23 |
| weish25-100 | 80 | 4 | 1 | 2,445 | 1,155.601 | 52.74 | TL | 2,392 | CUTOFF | 0.00 | 0.31 |
| weish26-100 | 90 | 4 | 1 | 4,266 | 1,627.125 | 61.86 | TL | 3,799 | CUTOFF | 0.00 | 1.25 |
| weish27-100 | 90 | 4 | 1 | 4,077 | 1,545.901 | 62.08 | TL | 3,565 | CUTOFF | 0.00 | 0.99 |
| weish28-100 | 90 | 4 | 1 | 4,441 | 1,635.000 | 63.18 | TL | 3,896 | CUTOFF | 0.00 | 0.84 |
| weish29-100 | 90 | 4 | 1 | 4,514 | 1,690.245 | 62.56 | TL | 3,997 | 3,997.000 | 0.00 | 1.55 |
| weish30-100 | 90 | 4 | 1 | 2,267 | 1,504.739 | 33.62 | TL | 2,226 | CUTOFF | 0.00 | 0.30 |

*Note.* BestSol, value of the best solution found; #FC, follower constraints; LB, best lower bound; #LC, leader constraints; % gap, associated optimality gap; *t* (s), runtime in seconds.

**Table 5.** Results for Instance Set SAC Compared with Results Obtained by the State-of-the-Art General Purpose Bilevel Solver Presented in Fischetti et al. (2017a)

| | | | | MIX++ from Fischetti et al. (2017a) | | | | B&C | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | \|N\| | #LC | #FC | BestSol | LB | %Gap | t (s) | BestSol | LB | %Gap | t (s) |
| hp1-50 | 28 | 2 | 2 | 1,536 | 1,536.000 | 0.00 | 21.21 | 1,536 | CUTOFF | 0.00 | 0.08 |
| hp2-50 | 35 | 2 | 2 | 2,912 | 2,912.000 | 0.00 | 263.13 | 2,912 | CUTOFF | 0.00 | 20.95 |
| pb1-50 | 27 | 2 | 2 | 1,536 | 1,536.000 | 0.00 | 12.90 | 1,536 | CUTOFF | 0.00 | 0.86 |
| pb2-50 | 34 | 2 | 2 | 1,787 | 1,787.000 | 0.00 | 181.58 | 1,787 | CUTOFF | 0.00 | 1.72 |
| pb5-50 | 20 | 5 | 5 | 1,625 | 1,625.000 | 0.00 | 48.58 | 1,625 | 1,624.985 | 0.00 | 301.03 |
| pb6-50 | 40 | 15 | 15 | 634 | 634.000 | 0.00 | 119.05 | 634 | 634.000 | 0.00 | 26.02 |
| pb7-50 | 37 | 15 | 15 | 423 | 423.000 | 0.00 | 1,073.40 | 423 | 423.000 | 0.00 | 106.16 |
| pet2-50 | 10 | 5 | 5 | 38,833 | 38,833.000 | 0.00 | 1.38 | 38,833 | CUTOFF | 0.00 | 0.07 |
| pet3-50 | 15 | 5 | 5 | 905 | 905.000 | 0.00 | 1.23 | 905 | CUTOFF | 0.00 | 0.07 |
| pet4-50 | 20 | 5 | 5 | 2,445 | 2,445.000 | 0.00 | 3.67 | 2,445 | 2,445.000 | 0.00 | 0.92 |
| pet5-50 | 28 | 5 | 5 | 3,025 | 3,025.000 | 0.00 | 45.83 | 3,025 | CUTOFF | 0.00 | 0.38 |
| pet6-50 | 39 | 3 | 2 | 3,936 | 3,936.000 | 0.00 | 474.10 | 3,936 | CUTOFF | 0.00 | 0.98 |
| pet7-50 | 50 | 3 | 2 | 5,873 | 5,031.324 | 14.33 | TL | 5,723 | CUTOFF | 0.00 | 14.05 |
| sento1-50 | 60 | 15 | 15 | 1,102 | 1,102.000 | 0.00 | 2,235.55 | 1,102 | 1102.000 | 0.00 | 76.06 |
| sento2-50 | 60 | 15 | 15 | 522 | 338.000 | 35.25 | TL | 503 | 503.000 | 0.00 | 10.32 |
| weish01-50 | 30 | 3 | 2 | 1,097 | 1,097.000 | 0.00 | 23.81 | 1,097 | 1097.000 | 0.00 | 1.22 |
| weish02-50 | 30 | 3 | 2 | 1,293 | 1,293.000 | 0.00 | 28.99 | 1,293 | CUTOFF | 0.00 | 0.20 |
| weish03-50 | 30 | 3 | 2 | 619 | 619.000 | 0.00 | 9.78 | 619 | 619.000 | 0.00 | 0.30 |
| weish04-50 | 30 | 3 | 2 | 1,027 | 1,027.000 | 0.00 | 5.57 | 1,027 | 1027.000 | 0.00 | 0.21 |
| weish05-50 | 30 | 3 | 2 | 1,215 | 1,215.000 | 0.00 | 8.22 | 1,215 | 1215.000 | 0.00 | 0.20 |
| weish06-50 | 40 | 3 | 2 | 1,369 | 1,369.000 | 0.00 | 373.87 | 1,369 | CUTOFF | 0.00 | 1.55 |
| weish07-50 | 40 | 3 | 2 | 1,407 | 1,407.000 | 0.00 | 804.42 | 1,407 | CUTOFF | 0.00 | 0.52 |
| weish08-50 | 40 | 3 | 2 | 1,369 | 1,369.000 | 0.00 | 372.00 | 1,369 | CUTOFF | 0.00 | 0.32 |
| weish09-50 | 40 | 3 | 2 | 1,568 | 1,568.000 | 0.00 | 108.72 | 1,568 | 1,568.000 | 0.00 | 0.54 |
| weish10-50 | 50 | 3 | 2 | 785 | 785.000 | 0.00 | 232.30 | 785 | 785.000 | 0.00 | 2.20 |
| weish11-50 | 50 | 3 | 2 | 584 | 584.000 | 0.00 | 58.67 | 584 | 584.000 | 0.00 | 2.16 |
| weish12-50 | 50 | 3 | 2 | 778 | 778.000 | 0.00 | 242.39 | 778 | 778.000 | 0.00 | 2.64 |
| weish13-50 | 50 | 3 | 2 | 742 | 742.000 | 0.00 | 140.11 | 742 | 742.000 | 0.00 | 2.25 |
| weish14-50 | 60 | 3 | 2 | 1,041 | 811.173 | 22.08 | TL | 1,020 | 1,020.000 | 0.00 | 40.00 |
| weish15-50 | 60 | 3 | 2 | 1,931 | 1,931.000 | 0.00 | 3,110.28 | 1,931 | 1,931.000 | 0.00 | 4.99 |
| weish16-50 | 60 | 3 | 2 | 2,198 | 1,474.000 | 32.94 | TL | 2,172 | 2,172.000 | 0.00 | 7.45 |
| weish17-50 | 60 | 3 | 2 | 991 | 819.052 | 17.35 | TL | 991 | CUTOFF | 0.00 | 0.06 |
| weish18-50 | 70 | 3 | 2 | 2,113 | 948.000 | 55.13 | TL | 1,945 | CUTOFF | 0.00 | 0.15 |
| weish19-50 | 70 | 3 | 2 | 1,194 | 599.807 | 49.76 | TL | 1,095 | 1,095.000 | 0.00 | 202.63 |
| weish20-50 | 70 | 3 | 2 | 2,274 | 1,021.000 | 55.10 | TL | 2,075 | CUTOFF | 0.00 | 0.42 |
| weish21-50 | 70 | 3 | 2 | 2,601 | 1,263.000 | 51.44 | TL | 2,451 | CUTOFF | 0.00 | 0.80 |
| weish22-50 | 80 | 3 | 2 | 1,522 | 504.071 | 66.88 | TL | 1,372 | 1,358.866 | 0.96 | TL |
| weish23-50 | 80 | 3 | 2 | 1,309 | 522.000 | 60.12 | TL | 1,236 | 1,236.000 | 0.00 | 1,026.43 |
| weish24-50 | 80 | 3 | 2 | 2,360 | 889.443 | 62.31 | TL | 2,111 | CUTOFF | 0.00 | 0.23 |
| weish25-50 | 80 | 3 | 2 | 2,576 | 920.311 | 64.27 | TL | 2,392 | CUTOFF | 0.00 | 0.42 |
| weish26-50 | 90 | 3 | 2 | 1,384 | 402.000 | 70.95 | TL | 1,243 | 1,242.896 | 0.00 | 2,913.69 |
| weish27-50 | 90 | 3 | 2 | 1,470 | 391.000 | 73.40 | TL | 1,290 | 1275.212 | 1.15 | TL |
| weish28-50 | 90 | 3 | 2 | 1,513 | 444.510 | 70.62 | TL | 1,358 | 1,357.908 | 0.00 | 2,079.98 |
| weish29-50 | 90 | 3 | 2 | 1,401 | 405.000 | 71.09 | TL | 1,205 | 1,196.822 | 0.68 | TL |
| weish30-50 | 90 | 3 | 2 | 2,356 | 1,221.892 | 48.14 | TL | 2,226 | CUTOFF | 0.00 | 0.09 |

*Notes.* The optimal solution value (obtained with a larger time limit) for weish22-50 is 1,372, for weish27-50 is 1,290, and for weish29-50 is 1,205. BestSol, value of the best solution found; #FC, follower constraints; LB, best lower bound; #LC, leader constraints; %Gap, associated optimality gap; $t$ (s), runtime in seconds.

these three instance with a larger time limit, and all of them could be solved to optimality within 3,900 seconds.

There seems to be no clear influence of the number of items and constraints on the performance of our approach (e.g., pb5 with 20 items and five leader and follower constraints takes 12 times as long as pb6 that has 40 items and 15 leader and follower constraints). Solver MIX++ turns out to be faster than B&C only for instance pb5-50 (49 versus 301 seconds).

Finally, Table 6 reports results for type −0. For MIX++, the results are very similar to the previous ones, and in 16 of 30 instances, weish* can be solved within the time limit of 3,600 seconds. Our approach B&C manages to outperform MIX++ for every instance, although it is not able to solve to optimality seven instances. However, for these unsolved instances (weish22, weish23, weish25, weish26, weish27, weish28, and weish29), the gap is at most 9.12% compared with

**Table 6.** Results for Instance Set SAC Compared with Results Obtained by the State-of-the-Art General Purpose Bilevel Solver Presented in Fischetti et al. (2017a)

| | | | | MIX++ from Fischetti et al. (2017a) | | | | B&C | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | \|*N*\| | #LC | #FC | BestSol | LB | %Gap | *t* (s) | BestSol | LB | %Gap | *t* (s) |
| hp1-0 | 28 | 1 | 3 | 1,467 | 1,467.000 | 0.00 | 12.86 | 1,467 | CUTOFF | 0.00 | 0.66 |
| hp2-0 | 35 | 1 | 3 | 2,278 | 2,278.000 | 0.00 | 377.55 | 2,278 | 2,278.000 | 0.00 | 4.47 |
| pb1-0 | 27 | 1 | 3 | 1,467 | 1,467.000 | 0.00 | 11.39 | 1,467 | CUTOFF | 0.00 | 0.70 |
| pb2-0 | 34 | 1 | 3 | 1,784 | 1,784.000 | 0.00 | 145.71 | 1,784 | CUTOFF | 0.00 | 4.14 |
| pb5-0 | 20 | 1 | 9 | 1,417 | 1,417.000 | 0.00 | 26.44 | 1,417 | CUTOFF | 0.00 | 16.63 |
| pb6-0 | 40 | 1 | 29 | 292 | 292.000 | 0.00 | 19.42 | 292 | 292.000 | 0.00 | 14.37 |
| pb7-0 | 37 | 1 | 29 | 185 | 185.000 | 0.00 | 31.28 | 185 | CUTOFF | 0.00 | 2.35 |
| pet2-0 | 10 | 1 | 9 | 25,295 | 25,295.000 | 0.00 | 0.39 | 25,295 | CUTOFF | 0.00 | 0.06 |
| pet3-0 | 15 | 1 | 9 | 905 | 905.000 | 0.00 | 0.66 | 905 | CUTOFF | 0.00 | 0.22 |
| pet4-0 | 20 | 1 | 9 | 1,935 | 1,935.000 | 0.00 | 2.71 | 1,935 | 1,935.000 | 0.00 | 1.04 |
| pet5-0 | 28 | 1 | 9 | 2,195 | 2,195.000 | 0.00 | 9.17 | 2,195 | CUTOFF | 0.00 | 0.13 |
| pet6-0 | 39 | 1 | 4 | 3,683 | 3,683.000 | 0.00 | 330.08 | 3,683 | CUTOFF | 0.00 | 1.43 |
| pet7-0 | 50 | 1 | 4 | 5,636 | 4,986.000 | 11.53 | TL | 5,459 | CUTOFF | 0.00 | 9.51 |
| sento1-0 | 60 | 1 | 29 | 552 | 552.000 | 0.00 | 856.95 | 552 | 552.000 | 0.00 | 78.60 |
| sento2-0 | 60 | 1 | 29 | 226 | 226.000 | 0.00 | 226.93 | 226 | CUTOFF | 0.00 | 1.07 |
| weish01-0 | 30 | 1 | 4 | 923 | 923.000 | 0.00 | 13.91 | 923 | 923.000 | 0.00 | 0.84 |
| weish02-0 | 30 | 1 | 4 | 1,108 | 1,108.000 | 0.00 | 18.34 | 1,108 | 1,108.000 | 0.00 | 0.56 |
| weish03-0 | 30 | 1 | 4 | 619 | 619.000 | 0.00 | 4.90 | 619 | 619.000 | 0.00 | 0.26 |
| weish04-0 | 30 | 1 | 4 | 465 | 465.000 | 0.00 | 5.49 | 465 | 465.000 | 0.00 | 1.80 |
| weish05-0 | 30 | 1 | 4 | 443 | 443.000 | 0.00 | 4.71 | 443 | 443.000 | 0.00 | 1.82 |
| weish06-0 | 40 | 1 | 4 | 1,283 | 1,283.000 | 0.00 | 340.99 | 1,283 | 1,283.000 | 0.00 | 7.69 |
| weish07-0 | 40 | 1 | 4 | 1,185 | 1,185.000 | 0.00 | 184.75 | 1,185 | 1,185.000 | 0.00 | 4.63 |
| weish08-0 | 40 | 1 | 4 | 1,283 | 1,283.000 | 0.00 | 387.82 | 1,283 | 1,283.000 | 0.00 | 5.55 |
| weish09-0 | 40 | 1 | 4 | 532 | 532.000 | 0.00 | 83.48 | 532 | 532.000 | 0.00 | 6.63 |
| weish10-0 | 50 | 1 | 4 | 785 | 785.000 | 0.00 | 280.40 | 785 | 785.000 | 0.00 | 10.96 |
| weish11-0 | 50 | 1 | 4 | 584 | 584.000 | 0.00 | 92.28 | 584 | 584.000 | 0.00 | 2.15 |
| weish12-0 | 50 | 1 | 4 | 778 | 778.000 | 0.00 | 314.01 | 778 | 778.000 | 0.00 | 3.36 |
| weish13-0 | 50 | 1 | 4 | 742 | 742.000 | 0.00 | 218.14 | 742 | 742.000 | 0.00 | 3.62 |
| weish14-0 | 60 | 1 | 4 | 1,046 | 786.000 | 24.86 | TL | 1,020 | 1,020.000 | 0.00 | 85.27 |
| weish15-0 | 60 | 1 | 4 | 759 | 759.000 | 0.00 | 2,484.18 | 759 | 759.000 | 0.00 | 57.82 |
| weish16-0 | 60 | 1 | 4 | 876 | 644.000 | 26.48 | TL | 828 | 828.000 | 0.00 | 278.70 |
| weish17-0 | 60 | 1 | 4 | 36 | 36.000 | 0.00 | 0.76 | 36 | CUTOFF | 0.00 | 0.08 |
| weish18-0 | 70 | 1 | 4 | 2,139 | 749.000 | 64.98 | TL | 1,927 | CUTOFF | 0.00 | 22.28 |
| weish19-0 | 70 | 1 | 4 | 1,170 | 600.301 | 48.69 | TL | 1,095 | 1,095.000 | 0.00 | 1,283.08 |
| weish20-0 | 70 | 1 | 4 | 1,086 | 526.000 | 51.57 | TL | 964 | 964.000 | 0.00 | 773.93 |
| weish21-0 | 70 | 1 | 4 | 988 | 569.000 | 42.41 | TL | 904 | 903.948 | 0.00 | 1,770.06 |
| weish22-0 | 80 | 1 | 4 | 1,465 | 485.962 | 66.83 | TL | 1,374 | 1,292.736 | 5.91 | TL |
| weish23-0 | 80 | 1 | 4 | 1,361 | 485.000 | 64.36 | TL | 1,248 | 1,161.442 | 6.94 | TL |
| weish24-0 | 80 | 1 | 4 | 2,401 | 571.713 | 76.19 | TL | 2,094 | CUTOFF | 0.00 | 27.41 |
| weish25-0 | 80 | 1 | 4 | 1,181 | 406.000 | 65.62 | TL | 1,090 | 1,001.539 | 8.12 | TL |
| weish26-0 | 90 | 1 | 4 | 1,484 | 365.237 | 75.39 | TL | 1,243 | 1,145.653 | 7.83 | TL |
| weish27-0 | 90 | 1 | 4 | 1,431 | 412.000 | 71.21 | TL | 1,296 | 1,177.850 | 9.12 | TL |
| weish28-0 | 90 | 1 | 4 | 1,482 | 434.000 | 70.72 | TL | 1,358 | 1,280.728 | 5.69 | TL |
| weish29-0 | 90 | 1 | 4 | 1,368 | 385.000 | 71.86 | TL | 1,206 | 1,110.184 | 7.94 | TL |
| weish30-0 | 90 | 1 | 4 | 829 | 314.210 | 62.10 | TL | 724 | CUTOFF | 0.00 | 91.42 |

*Notes.* The optimal solution value (obtained with a larger time limit) for weish22-0 is 1,372, for weish23-0 is 1,236, for weish25-0 is 1,079, for weish26-0 is 1,243, for weish27-0 is 1,290, for weish28-0 is 1,358, and for weish29-0 is 1,205. BestSol, value of the best solution found; #FC, follower constraints; LB, best lower bound; #LC, leader constraints; %gap, associated optimality gap; *t* (s), runtime in seconds.

gaps of 25%–75% for MIX++. In general, instances of class -0 seem more difficult than instances of class -50 (and of course, also class -100); thus, the number/ratio of leader/follower constraints seem to influence the difficulty of the problem. This effect seems not just to be restricted to the case in which the follower has just a single-knapsack constraint.

Moreover, the number of follower constraints also seem to influence the effectiveness of the heuristic, because for type -0, only for 15 of 45 instances was the solution of the heuristic the optimal one. Again, we reran the unsolved -0 instances with a larger time limit, and all of them could be solved to proven optimality within 19,000 seconds, except weish27-0, which required about 30,000 seconds.

## 6. Conclusions

In this article, we have considered IGs in which the follower subproblem satisfies a certain monotonicity property. We have shown that this property is fulfilled by important classes of IGs, including the (single and multiple) knapsack problem, the facility location problem, and the PCTSP (or Steiner tree problem)—just to mention a few.

For this large and important family of problems, we have proposed a new class of interdiction cuts that generalize those previously used in the literature. Building on these cuts, we have developed a Benders-like framework with some important enhancing ingredients. We have discussed additional families of modified/lifted interdiction cuts as well as new dominance-based valid inequalities. For all classes of cuts, we have proposed exact and/or heuristic separation procedures, and we have used them to develop an effective solver. Finally, we have introduced a preprocessing procedure based on a new heuristic single-level compact MILP formulation.

We have computationally shown that our new solver significantly outperforms very recent methods from the literature. In particular, we have tested our approach on 360 knapsack interdiction instances from the recent literature and have proved the optimality for all of them—including for the 27 previously unsolved ones. Our algorithm needs at most 84 seconds for solving any of these instances (for only four of these 360 instances, it takes more than 10 seconds), outperforming previous approaches from the literature by up to four orders of magnitude. Computational tests on new random instances based on 0/1 multidimensional knapsack problems have also been performed to assess the dependency of our approach on the number of leader and follower constraints. Also for this kind of instance, our approach outperforms by orders of magnitude the state-of-the art general bilevel solver recently proposed in Fischetti et al. (2017a).

Future work should address the extension of our approach to the nonmonotone case as well as the customization of our solution method to special classes of monotone IGs, including the facility location and prize-collecting applications outlined in Section 2.

### Acknowledgments

The authors thank two anonymous referees for helpful comments.

### References

Balas E (1989) The prize collecting traveling salesman problem. *Networks* 19(6):621–636.

Balas E (2007) The prize collecting traveling salesman problem and its applications. Gutin G, Punnen AP, eds. *The Travelling Salesman Problem and Its Variations,* Combinatorial Optimization, vol. 12 (Springer, Boston), 663–695.

Bazgan C, Toubaline S, Tuza Z (2011) The most vital nodes with respect to independent set and vertex cover. *Discrete Appl. Math.* 159(17):1933–1946.

Bienstock D, Goemans MX, Simchi-Levi D, Williamson D (1993) A note on the prize collecting traveling salesman problem. *Math. Programming* 59(1–3):413–420.

Brown G, Carlyle M, Salmeron J, Wood K (2005) Analyzing the vulnerability of critical infrastructure to attack and planning defenses. Smith JC, ed. *Emerging Theory, Methods, and Applications,* TutORials in Operations Research (INFORMS, Catonsville, MD), 102–123.

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

Brown G, Carlyle M, Harney R, Skroch E, Wood K (2009) Interdicting a nuclear-weapons project. *Oper. Res.* 57(4):866–877.

Caprara A, Carvalho M, Lodi A, Woeginger GJ (2014) A study on the computational complexity of the bilevel knapsack problem. *SIAM J. Optim.* 24(2):823–838.

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

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

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

Dinitz M, Gupta A (2013) Packing interdiction and partial covering problems. Goemans M, Correa J, eds. *Proc. 16th Internat. Conf. Integer Programming Combin. Optim.* (Springer-Verlag, Berlin), 157–168.

Fischetti M, Ljubić I, Monaci M, Sinnl M (2016) Intersection cuts for bilevel optimization. Louveaux Q, Skutella M, eds. *Proc. 18th Internat. Conf. Integer Programming Combin. Optim.* (Springer International Publishing, Cham, Switzerland), 77–88.

Fischetti M, Ljubić I, Monaci M, Sinnl M (2017a) An improved branch-and-cut algorithm for mixed-integer bilevel linear programs. *Oper. Res.* 65(6):1615–1637.

Fischetti M, Leitner M, Ljubić I, Luipersbeck M, Monaci M, Resch M, Salvagnin D, Sinnl M (2017b) Thinning out Steiner trees: A node-based model for uniform edge costs. *Math. Programming Comput.* 9(2):203–229.

Freville A, Plateau G (1990) Hard 0-1 multiknapsack test problems for size reduction methods. *Investigation Oper.* 1:251–270.

Halldórsson MM (2000) Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms Appl.* 4(1):1–16.

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

Khuri S, Baeck T, Heitkoetter J (1994) SAC94 suite: Collection of multiple knapsack problems. Accessed October 31, 2016, http://www.cs.cmu.edu/Groups/AI/areas/genetic/ga/test/sac/0.html.

Kleniati PM, Adjiman CS (2015) A generalization of the branch-and-sandwich algorithm: From continuous to mixed-integer nonlinear bilevel problems. *Comput. Chemical Engrg.* 72:373–386.

Lim C, Smith J (2007) Algorithms for discrete and continuous multicommodity flow network interdiction problems. *IIE Trans.* 39(1):15–26.

Ljubić I, Weiskircher R, Pferschy U, Klau GW, Mutzel P, Fischetti M (2006) An algorithmic framework for the exact solution of the prize-collecting steiner tree problem. *Math. Programming* 105(2-3):427–449.

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

Martello S, Toth P (1990) *Knapsack Problems: Algorithms and Computer Implementations* (John Wiley & Sons, Chichester, NY).

Martello S, Pisinger D, Toth P (1999) Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Sci.* 45(3):414–424.

Moore J, Bard J (1990) The mixed integer linear bilevel programming problem. *Oper. Res.* 38(5):911–921.

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

Petersen CC (1967) Computational experience with variants of the Balas algorithm applied to the selection of R&D projects. *Management Sci.* 13(9):736–750.

Prodon A, DeNegre S, Liebling TM (2010) Locating leak detecting sensors in a water distribution network by solving prize-collecting steiner arborescence problems. *Math. Programming* 124(1/2):119–141.

Ralphs T (2015) Bilevel integer optimization: Theory and algorithms. *Proc. 22nd Internat. Sympos. Math. Programming, Pittsburgh,* 1–44.

Senju S, Toyoda Y (1968) An approach to linear programming with 0-1 variables. *Management Sci.* 15(4):B196–B207.

Shih W (1979) A branch and bound method for the multiconstraint zero-one knapsack problem. *J. Oper. Res. Soc.* 30(4):369–378.

Smith JC, Lim C (2008) *Algorithms for Network Interdiction and Fortification Games* (Springer, New York), 609–644.

Song Y, Shen S (2016) Risk averse shortest path interdiction. *INFORMS J. Comput.* 28(3):527–539.

Tang Y, Richard JPP, Smith JC (2015) A class of algorithms for mixed-integer bilevel min–max optimization. *J. Global Optim.* 66(2):225–262.

Vansteenwegen P, Souffriau W, Van Oudheusden D (2011) The orienteering problem: A survey. *Eur. J. Oper. Res.* 209(1):1–10.

von Stackelberg H (1952) *The Theory of the Market Economy* (Oxford University Press, London, UK).

Washburn A, Wood K (1995) Two-person zero-sum games for network interdiction. *Oper. Res.* 43(2):243–251.

Weingartner HM, Ness DN (1967) Methods for the solution of the multidimensional 0/1 knapsack problem. *Oper. Res.* 15(1): 83–103.

Wood RK (2010) *Bilevel Network Interdiction Models: Formulations and Solutions* (John Wiley & Sons, Chichester, NY).

Xu P (2012) Three essays on bilevel optimization algorithms and applications. PhD thesis, Iowa State University, Ames.

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.

Zeng B, An Y (2014) Solving bilevel mixed integer program by reformulations and decomposition. Accessed August 20, 2016, http://www.optimization-online.org/DB_FILE/2014/07/4455.pdf.

Zenklusen R (2010) Matching interdiction. *Discrete Appl. Math.* 158(15):1676–1690.