European Journal of Operational Research 324 (2025) 742–758

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Discrete Optimization

Integer linear programming formulations for the maximum flow blocker problem

Isma Bentoumi [b,d], Fabio Furini [c], A. Ridha Mahjoub [a,b,*], Sébastien Martin [d]

[a] Department of Statistics and Operations Research, College of Science, Kuwait University, Kuwait
[b] LAMSADE, UMR CNRS 7243, University Paris-Dauphine, PSL, Place du Marechal de Lattre de Tassigny, 75016 Paris, France
[c] Department of Computer, Control and Management Engineering "Antonio Ruberti", Sapienza University of Rome, Via Ariosto 25 00185, Roma, Italy
[d] Huawei Technologies France, 19 Quai du Point du Jour, 92100 Boulogne-Billancourt, France

ARTICLE INFO

Keywords:
Combinatorial optimization
Bilevel problems
Blocker problem
Interdiction problem
Maximum-flow problem

ABSTRACT

Given a network with capacities and blocker costs associated with its arcs, we study the maximum flow blocker problem (FB). This problem seeks to identify a minimum-cost subset of arcs to be removed from the network, ensuring that the maximum flow value from the source to the destination in the remaining network does not exceed a specified threshold. The FB finds applications in telecommunication networks and monitoring of civil infrastructures, among other domains. We undertake a comprehensive study of several new integer linear programming (ILP) formulations designed for the FB. The first type of model, featuring an exponential number of constraints, is solved through tailored Branch-and-Cut algorithms. In contrast, the second type of ILP model, with a polynomial number of variables and constraints, is solved using a state-of-the-art ILP solver. The latter formulation establishes a structural connection between the FB and the maximum flow interdiction problem (FI), introducing a novel approach to obtaining solutions for each problem from the other. The ILP formulations proposed for solving the FB are evaluated thanks to a theoretical analysis assessing the strength of their LP relaxations. Additionally, the exact methods presented in this paper undergo a thorough comparison through an extensive computational campaign involving a set of real-world and synthetic instances. Our tests aim to evaluate the performance of the exact algorithms and identify the features of instances that can be solved with proven optimality.

1. Introduction

Let $N = (V, A)$ be a network where A is a set of m arcs and V is a set of n vertices containing two special vertices: the source $s \in V$ and the destination $t \in V$. Each arc $a \in A$ is given a positive integer *capacity* $c_a \in \mathbb{Z}_+$. We also denote an arc a as (u, v) where $u \in V$ is the tail and $v \in V$ is the head of the arc. Let y_a be a non-negative continuous variable representing the value of the flow passing through an arc $a \in A$. A *feasible flow* from the source to the destination, or simply a flow, has to respect the following two typologies of constraints. The first m constraints are called the *capacity constraints* for the arcs:

$$0 \leq y_a \leq c_a, \qquad a \in A, \tag{1}$$

and the second set of n constraints are called the *flow-conservation constraints* for the vertices:

$$\sum_{a \in \delta^+(u)} y_a - \sum_{a \in \delta^-(u)} y_a = 0, \qquad u \in V, \tag{2}$$

where $\delta^+(u) = \{(u, v) \in A : v \in V \setminus \{u\}\}$ (the outgoing arcs of vertex u) and $\delta^-(u) = \{(v, u) \in A : v \in V \setminus \{u\}\}$ (the entering arcs of vertex u). Without loss of generality, we consider networks in which the source has only one entering arc from the destination, i.e., $\delta^-(s) = \{(t, s)\}$ and the destination has only one outgoing arc, i.e., $\delta^+(t) = \{(t, s)\}$.

The *maximum flow problem* (MF) asks to determine the maximum feasible flow of a network from the source to the destination and accordingly, the maximum flow value. By construction, this value corresponds to the maximum flow value outgoing from the source and entering the destination, as well as the maximum flow value on the arc (t, s). It is one of the fundamental problems in optimization, largely studied since its introduction in the 1950s, see e.g., Schrijver (2002). A min–max relation links the MF to the *minimum cut problem* (MC). This relationship establishes an equivalence between the two problems, see e.g., Ahuja et al. (1993), stating that the optimal solution values of the two problems are the same. Given a subset of vertices $U \subseteq V$ such that $s \in U$ and $t \in V \setminus U$, an $s - t$ *cut* (or simply a cut) of N,

* Corresponding author at: Department of Statistics and Operations Research, College of Science, Kuwait University, Kuwait.
E-mail addresses: isma.bentoumi@dauphine.eu (I. Bentoumi), fabio.furini@uniroma1.it (F. Furini), ridha.mahjoub@ku.edu.kw, ridha.mahjoub@lamsade.dauphine.fr (A.R. Mahjoub), sebastien.martin@huawei.com (S. Martin).

https://doi.org/10.1016/j.ejor.2025.02.013
Received 15 January 2024; Accepted 8 February 2025
Available online 17 February 2025

denoted by $\delta(U)$, is the subset of arcs having the tail in U and the head in $V \setminus U$. Accordingly, the MC asks for finding a cut of minimum value which corresponds to the maximum flow value in the network. We refer the interested reader to Ahuja et al. (1993) for further details on the MF and the MC, in particular, for the Linear Programming (LP) formulations of the two problems which feature totally unimodular system of constraints (see Schrijver, 2003) and for the polynomial-time algorithms designed to find their optimal solutions.

In this article, we propose the first Integer Linear Programming (ILP) formulations and exact algorithms for the *blocker variant* of the MF in which each arc $a \in A$ is also given a positive integer *blocker cost* $r_a \in \mathbb{Z}_+$. This problem, called the *maximum flow blocker problem* (FB), consists in finding a minimum-cost subset of arcs to be removed from the network N, i.e., *blocked*, in such a way that the maximum flow value between s and t in the remaining network is not larger than a given threshold. The threshold is called the *target flow*, a positive integer value denoted by $\Phi \in \mathbb{Z}_+$.

We introduce two vectors of m positive integer values, the arc-capacity vector $\boldsymbol{c} \in \mathbb{Z}_+^m$ and the blocker cost vector $\boldsymbol{r} \in \mathbb{Z}_+^m$, containing respectively the capacities and blocker costs associated with arcs of the network N. Accordingly, an instance of the FB can be represented as a tuple $(N, \boldsymbol{c}, \boldsymbol{r}, \Phi)$ and we denote by $\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi))$ its optimal solution value. Moreover, we denote by $f_m(N, \boldsymbol{c})$ the maximum flow value of a network with a given arc-capacity vector. Without loss of generality, we consider instances in which $\Phi < f_m(N, \boldsymbol{c})$, otherwise, clearly, an optimal FB solution is the empty subset of blocked arcs. Finally, it is worth noticing that the FB is $\mathcal{NP}$-hard and we provide in Section 3.4 a reduction from the *maximum flow interdiction problem* (FI), which is $\mathcal{NP}$-hard as shown in Wood (1993).

1.1. Graphical illustration of FB optimal solutions

We illustrate in this section the features of optimal FB solutions thanks to an example network N with $n = 9$ vertices and $m = 17$ arcs shown in Figs. 1 and 2. This network is initially used to show optimal MF and MC solutions. For clarity of these two figures and the following ones, we do not report the arc (t, s), assuming that the values $c_{(t,s)}$ and $r_{(t,s)}$ of its capacity and blocker cost, respectively, are sufficiently large to have no impact on the maximum flow value of the network nor the optimal solutions of the FB. In Fig. 1, we report on each arc two values separated by the symbol "/": the first one, in blue, is the flow on the arc; the second one, in black, is the capacity of the arc. Thicker bold red arcs represent the arcs of an optimal MC solution. In this example, the maximum flow value $f_m(N, \mathbf{c})$ between s and t is 36 which, as a consequence of the *max-flow min-cut theorem*, coincides with the value of the minimum cut. In this example, the minimum cut $\delta(U)$ is given by the set of arcs (v_1, v_5), (v_4, v_6), (v_4, v_7) and (v_3, v_7), U is the set of vertices $\{s, v_1, v_2, v_3, v_4\}$ and $V \setminus U = \{v_5, v_6, v_7, t\}$. The gray shaded surface surrounds, on both sides of the minimum cut, the set of vertices U and $V \setminus U$. By removing all the arcs of the minimum cut $\delta(U)$, in a blocker optic, no flow can be routed from the source s to the destination t. In other words, an optimal MC solution provides a set of arcs of minimum capacity whose removal does not allow any flow from the source to the destination.

The same network is then used to illustrate an optimal FB solution in Fig. 2. We report on each arc its blocker cost shown in red above the flow of the arc in the remaining network and the arc capacity. We consider a target flow $\Phi = 23$ ($\approx$ 64% of the maximum flow value $f_m(N, \mathbf{c})$). In the optimal FB solution shown, two arcs are blocked, (v_1, v_5) and (v_2, v_4), with a minimum total blocker cost equal to 6. The blocked arcs are depicted with dashed blue lines. The maximum flow value in the remaining network is $18 \leq \Phi = 23$. The figure shows with thicker bold red lines the arcs of the minimum cut in the remaining network. It is worth noticing that neither these arcs nor the blocked arcs are contained in the minimum cut of N (see Section 3 for further comments on this important point).

1.2. FB applications

The FB has applications in several domains. In telecommunication networks, the flow routed on an arc represents the amount of data sent from the source to the destination and passing through that arc. In this context, FB optimal solutions allow analyzing the network resilience in case of malfunctioning arcs. Malfunctioning arcs can be modeled by reducing the capacity of the arc to 0 or by removing the arc from the network. These situations are due to anomalies, failures, or packet loss caused by congestion. We refer the interested reader to Junior et al. (2019) for further details on telecommunication network anomalies.

In the case where all blocker costs are equal to one, a network is said to be *resilient* to f simultaneous arc failures, concerning a target flow Φ, if after the removal of any set of at most f arcs, there exists a flow of value larger than Φ. An optimal FB solution $\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi))$ is a set of arcs of minimum cardinality such that the network is not resilient to $\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi))$ simultaneous arc failures concerning a target flow Φ. By considering $f = \zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi)) - 1$, since $\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi))$ is the minimum size of the set of arcs, thus, removing any set of arcs of size at most $\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi)) - 1$ (maximum simultaneous malfunctioning arcs) ensures that the maximum flow value in the remaining network is larger than Φ. Let us consider again the network, the capacities shown in Fig. 2 and the associated FB with all blocker costs equal to one and $\Phi = 10$, the optimal solution value is $\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi)) = 2$ and the arcs (v_6, t) and (v_7, t) are an optimal FB solution. The consequence of removing these arcs results in a network containing a maximum flow value of 7. Accordingly, for any single malfunctioning arc, the remaining network has a maximum flow value larger than $\Phi = 10$.

Moreover, always in telecommunication networks with general blocker costs, FB solutions determine the minimum-cost subset of arcs to monitor all flows of value strictly larger than Φ. More precisely, for each subset of arcs guaranteeing a flow strictly greater than Φ, at least one of them is monitored. The blocker costs in this case represent the costs of installing the monitor devices on the arcs. We refer the interested reader to Ghafir et al. (2016) for further details on network monitoring.

A second application of the FB is in the context of monitoring civil infrastructures, where the aim is to monitor the transportation links to reduce illegal traffic (see e.g., Wood, 1993). FB solutions determine the links that need to be controlled to have an illegal flow no larger than a given threshold. It is worth noticing that the blocker costs can also be seen as protection costs. In this context, FB solutions identify the arcs that need to be protected to preserve the flow in the network.

1.3. Related problems and literature review

To the best of our knowledge, there are no articles on the FB and, for this reason, we focus this section on related problems studied in the literature. Given a network, together with arc-capacities, positive integer *interdiction costs* $q_a \in \mathbb{Z}_+$ on the arcs $a \in A$ and an *interdiction budget* denoted by Ψ, the *maximum flow interdiction problem* (FI) consists in finding a subset of arcs of total interdiction cost no larger than Ψ to be removed from the network, i.e., *interdicted*, in such a way that the maximum flow value in the remaining network is minimized. We introduce the interdiction cost vector $\boldsymbol{q} \in \mathbb{Z}_+^m$ of m positive integer values containing the interdiction costs associated with the arcs of N. Accordingly, an instance of the FI can be represented as a tuple $(N, \boldsymbol{c}, \boldsymbol{q}, \Psi)$ and we denote by $\zeta(\text{FI}(N, \boldsymbol{c}, \boldsymbol{q}, \Psi))$ its optimal solution value.

The FI is the closest problem to the FB since they share a similar bilevel structure, see Smith and Song (2020) for a survey on bilevel optimization. The FI has a long history of research, see e.g., Wollmer (1964), McMasters and Mustin (1970), and Wood (1993). Recent studies, including Lei et al. (2017), demonstrate its relevance and application in several domains such as in contrasting illegal traffic, protecting civil infrastructures, and for military planning.

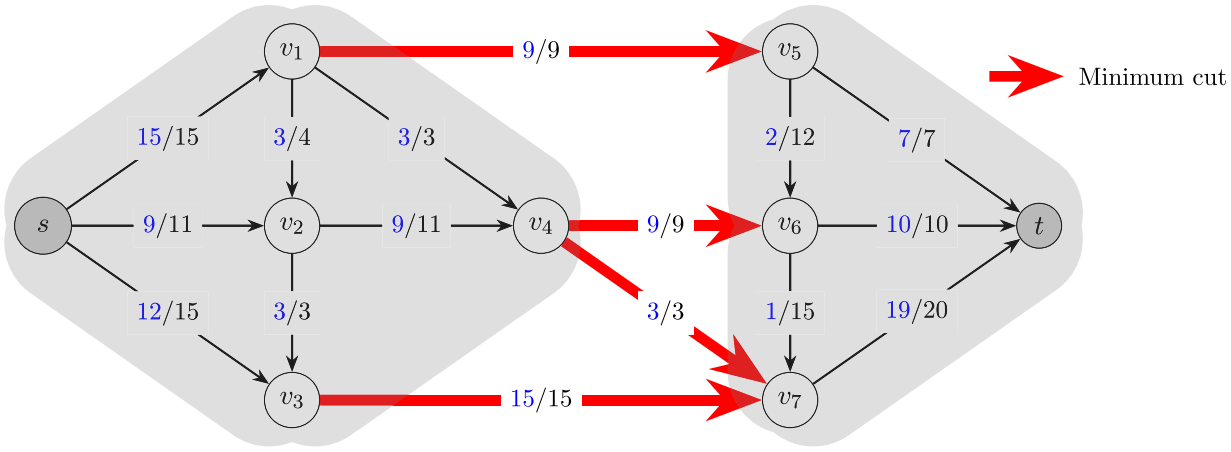

Fig. 1. An example network showing optimal MF and MC solutions.

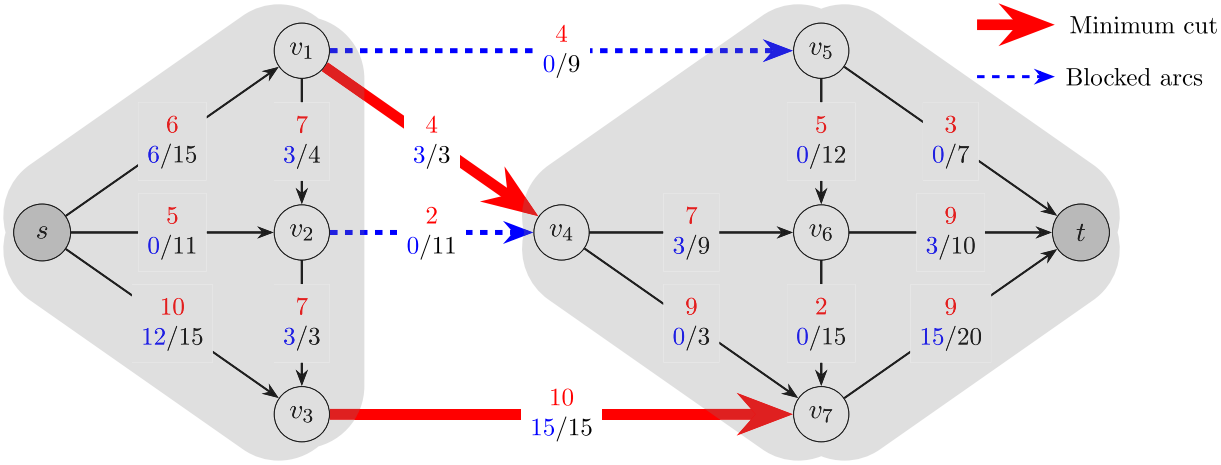

Fig. 2. An optimal FB solution with a target flow $\Phi = 23$.

The seminal work of Wollmer (1964) provides the first exact algorithm to solve the FI on planar networks and with a cardinality constraint, i.e., with a maximum number of arcs that can be interdicted. Another exact algorithm has been developed in McMasters and Mustin (1970) to solve the FI for the case in which the capacities of the arcs can be reduced. In Wood (1993), the first bilevel model for the FI is proposed as well as a technique for obtaining a single-level ILP formulation (see Section 3 for further details). In Royset and Wood (2007), the authors study a bi-objective version of the FI. The first objective is to minimize the total interdiction cost while the second objective aims to minimize the maximum flow remaining in the network after the interdiction of the arcs. The authors propose an algorithm that solves a sequence of FIs through Lagrangian relaxation. This approach identifies the efficient frontier which is computed using a specialized branch-and-bound algorithm. In Wood (2011), a generic Benders decomposition algorithm is presented to solve bilevel interdiction problems, and a specific tailoring of this procedure is developed for the FI based on its particular combinatorial structure. Specifically, valid inequalities are derived to improve the performance of the model. Finally, Altner et al. (2010) present results on the approximability of maximum flow interdiction problems. After introducing two classes of valid inequalities for the FI, they demonstrate that the integrality gap of the linear programming relaxation of the ILP model proposed by Wood (1993) is not bounded by a constant, even with the inclusion of the two families of inequalities. Furthermore, they offer insights into the hardness of approximating the FI using a reduction from a simpler interdiction problem, the R-Interdiction Covering Problem.

Blocker and interdiction bilevel problems have received a large amount of attention in the last few years, in particular for combinatorial optimization problems in networks. We refer the interested reader to, e.g., Pajouh et al. (2014), Furini et al. (2019, 2021) for recent studies on maximum clique interdiction and blocker problems. In Pajouh et al. (2014), the authors study the minimum vertex blocker clique problem, which involves removing a subset of vertices of minimum cardinality in a weighted network such that the maximum weight of a clique remaining in the network does not exceed a given threshold. This problem is modeled as a linear 0–1 program with an exponential number of constraints and is solved using a row generation approach. The vertex interdiction variant of the maximum clique problem is investigated in Furini et al. (2019), where an ILP formulation with an exponential number of constraints, called *Clique Interdiction Cuts*, is proposed, along with a polyhedral analysis and efficient bounds. In Furini et al. (2021), the authors focus on the edge interdiction variant, where edges are removed from the network instead of vertices. The approach developed employs a set-covering formulation with an exponential number of constraints, known as *clique-covering inequalities*, and solved via a Branch-and-Cut algorithm. Several other problems have also been investigated in their interdiction and blocker variants, see e.g., Magnouche and Martin (2020) and Israeli and Wood (2002) for the shortest path blocker and interdiction problems. Israeli and Wood (2002) present a bilevel formulation to solve the shortest path interdiction problem and introduce two different methods based on decomposition algorithms. The first method employs Benders decomposition, where cutting planes are added to the master problem. The

second method involves reformulating the master problem into a set-covering problem. Magnouche and Martin (2020) studied the shortest path blocker problem under the name of the most vital vertices for the shortest path problem. The authors present an ILP formulation with an exponential number of constraints, solved using a Branch-and-Cut algorithm. They also conducted a polyhedral analysis to enhance the model. In Wei and Walteros (2022), the authors study a general class of interdiction and blocker problems in which discrete sets characterize the leader and the follower and they propose generic methods to solve these kinds of problems.

1.4. Contributions and outline of the paper

After introducing a bilevel model for the FB, we present in Section 2 the first three new ILP formulations for the problem. These models use natural variables associated with the arcs of the network and include families of constraints with exponential size. To address this, we develop a tailored Branch-and-Cut algorithm for each model, based on separation subroutines for both integer and fractional solutions. Additionally, we provide complexity results for the separation problems and theoretically compare the strength of these families of constraints. In Section 3, we introduce a fourth and novel ILP formulation for the FB, which is a compact model featuring a polynomial number of variables and constraints. This model establishes a structural relationship between the FB and the FI, demonstrating that optimal solutions for one problem can be effectively obtained using optimal solutions from the other. To the best of our knowledge, this is the first time such a link between a blocker and an interdiction problem has been obtained. In conclusion to this section, we also conduct a theoretical comparison to assess the strength of the LP relaxation of the fourth model. Furthermore, we provide complexity results for both the FB and the FI, including a special case in which all arcs have the same blocker or interdiction cost and the same capacity. In Section 4, we present the results of an extensive computational campaign, comparing the performance of the four proposed ILP formulations with a state-of-the-art bilevel solver. These results allow us to identify the best exact approach for solving the FB and to determine the features of the FB instances that can be solved to proven optimality. Finally, in Section 5, we summarize our conclusions and outline potential future research directions.

2. Natural ILP models for the FB

In this section we introduce a FB model that belongs to the class of blocker models, a special class of bilevel optimization formulations, see e.g., Dempe (2020). This model and its structural properties are the base of the natural ILP formulations developed in Sections 2.2 and 2.4, each featuring an exponential family of constraints. For each family of constraints, we describe a separation procedure along with its complexity (Sections 2.3 and 2.5). We also compare the strength of the LP relaxations of the first two formulations (see Section 2.6). We then propose in Section 2.7 a third ILP formulation featuring the two families of constraints together.

2.1. A bilevel formulation

There are two types of variables in blocker models, the first-level decision variables associated with the so-called *leader problem* which affect the second-level decision variables associated with the so-called *follower problem*. For the FB, the leader determines a set of blocked arcs to be removed from the network and the follower determines the maximum flow value in the remaining network. The leader anticipates the optimal follower's solution to choose the minimum cost subset of arcs to be blocked that results in a remaining network having a maximum flow value no larger than Φ.

Let us introduce a vector $\boldsymbol{x} \in \{0,1\}^m$ of m binary first-level variables, each variable x_a is associated to an arc $a \in A$ and takes the value 1 if and only if the arc a is blocked, that is, removed from the network N. A bilevel model for the FB reads as follows:

$$\zeta(\mathrm{FB}(N,\boldsymbol{c},\boldsymbol{r},\Phi)) = \min_{\boldsymbol{x}\in\{0,1\}^m} \left\{ \sum_{a\in A} r_a\, x_a : \ \vartheta(\boldsymbol{x}) \le \Phi \right\}. \tag{3}$$

The link between the leader problem and the follower problem is established by the *value function* $\vartheta(\boldsymbol{x})$ which returns the maximum flow value in the network containing only the non-blocked arcs. This value is determined by the follower's objective function which also corresponds to the maximum flow value in the network where the capacity of blocked arcs is set to 0. As far as the leader problem is concerned, the objective function of (3) minimizes the total cost of the blocked arcs under the constraint imposing that the maximum flow value $\vartheta(\boldsymbol{x})$ is no larger than Φ.

Let us introduce a second vector $\boldsymbol{y} \in \mathbb{Q}_+^m$ of m non-negative second-level variables, each variable $y_a \ge 0$ is associated to an arc $a \in A$ and it represents the value of the flow on the arc. Therefore, the bilevel model (3) for the FB can be rewritten as follows:

$$\zeta(\mathrm{FB}(N,\boldsymbol{c},\boldsymbol{r},\Phi)) = \min_{\boldsymbol{x}\in\{0,1\}^m} \sum_{a\in A} r_a\, x_a \tag{4a}$$

$$\vartheta(\boldsymbol{x}) \le \Phi, \tag{4b}$$

$$\text{where} \quad \vartheta(\boldsymbol{x}) = \max_{\boldsymbol{y}\in\mathbb{Q}_+^m} \sum_{a\in\delta^+(s)} y_a \tag{4c}$$

$$\sum_{a\in\delta^+(u)} y_a - \sum_{a\in\delta^-(u)} y_a = 0, \quad u \in V, \tag{4d}$$

$$y_a \le c_a\,(1-x_a), \quad a \in A. \tag{4e}$$

Constraints (4d) are the flow conservation constraints, see (2), of the vertices. Constraints (4e) model the capacity constraints of the arcs, see (1), and they impose, at the same time, a flow of value 0 on blocked arcs.

A binary realization $\boldsymbol{x} \in \{0,1\}^m$ of the first-level variables is called a *blocker policy* and it generates a *non-blocked network* $\mathcal{N}_{NB}(\boldsymbol{x}) = \big(V, \mathcal{A}_{NB}(\boldsymbol{x})\big)$, i.e., the network with the same vertex set of N and only the non-blocked arcs $a \in A$ with $x_a = 0$ (denoted $\mathcal{A}_{NB}(\boldsymbol{x})$). It is worth noticing that $\vartheta(\boldsymbol{x})$ corresponds to the maximum flow value in the non-blocked network $\mathcal{N}_{NB}(\boldsymbol{x})$.

In Theorem 2 of Cormican et al. (1998), an important structural property of the follower problem is established. In particular, given a blocker policy $\boldsymbol{x}$, it is proven that the follower problem can be restated as follows:

$$\vartheta(\boldsymbol{x}) = \max_{\boldsymbol{y}\in\mathbb{Q}_+^m} \left\{ \sum_{a\in\delta^+(s)} y_a - \sum_{a\in A} x_a\, y_a : \ (1),\ (2) \right\}. \tag{5}$$

In this LP reformulation, constraints (4e) are replaced with the "standard" capacity constraints (1) for the arcs. The constraints of the follower do not depend anymore on the first-level variables and a penalization term is added to the new objective function to ensure that $\vartheta(\boldsymbol{x})$ is the maximum flow value in the non-blocked network $\mathcal{N}_{NB}(\boldsymbol{x})$. In other words, Theorem 2 of Cormican et al. (1998) states that the maximum flow value in $\mathcal{N}_{NB}(\boldsymbol{x})$ is equal to the outgoing flow from the source minus the total flow on the blocked arcs. It is worth noticing that the follower problem is an LP formulation featuring a totally unimodular system of constraints. Since all arcs $a \in A$ have an integer capacity $c_a \in \mathbb{Z}_+$, its variables consequently take integer values in its optimal solutions. Moreover, Model (5) remains clearly valid also for fractional solution $\boldsymbol{x} \in [0,1]^m$.

2.2. A first single-level ILP model for the FB

In this section, we introduce the first ILP formulation for the FB using only the vector $\boldsymbol{x}$ of binary variables, i.e., the natural variables associated with the arcs whose binary realizations represent the blocker

policies. The ILP model is obtained starting from the bilevel model (4) and projecting out the variables $\boldsymbol{y}$ of the follower model (5). This method is based on a Benders decomposition approach derived from the work of Wood (2011) for general bilevel interdiction problems. In this section, we adapt the approach to address the blocker variant of the MF. Additionally, we introduce an exact algorithm designed specifically to solve the ILP formulation.

The polytope of feasible solutions for the follower subproblem, which does not depend on the leader variables as shown in (5), can be defined as follows:

$$\mathcal{P}_f = \left\{ \boldsymbol{y} \in \mathbb{Q}_+^m : \sum_{a \in \delta^+(s)} y_a \geq \boldsymbol{\Phi} + 1, \quad (1), (2). \right\}. \tag{6}$$

We remark that only flows of value strictly larger than $\boldsymbol{\Phi}$ are associated with the non-dominated constraint (4b). This can be imposed by a "$\geq \boldsymbol{\Phi} + 1$" constraint since all the capacities of the arcs are integer values. For this reason, $\mathcal{P}_f$ includes the constraint imposing the lower bound on the value of the flow outgoing from the source. The model (5) is valid for any (fractional) vector $\boldsymbol{x} \in [0,1]^m$ and, since the objective function is linear, it is sufficient to optimize over the set of extreme points $\boldsymbol{y}$ of $\mathcal{P}_f$ (denoted $ext(\mathcal{P}_f)$. The constraints (4b) can then be restated as follows:

$$\vartheta(\boldsymbol{x}) = \max_{\boldsymbol{y} \in ext(\mathcal{P}_f)} \left\{ \sum_{a \in \delta^+(s)} y_a - \sum_{a \in A} x_a \, y_a \right\} \leq \boldsymbol{\Phi}. \tag{7}$$

Accordingly, by applying a Benders-like decomposition to the bilevel model (4), we obtain the following single-level ILP reformulation for the FB:

$$\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \boldsymbol{\Phi})) = \min_{\boldsymbol{x} \in \{0,1\}^m} \sum_{a \in A} r_a \, x_a \tag{8a}$$

$$\sum_{a \in \delta^+(s)} y_a - \sum_{a \in A} x_a \, y_a \leq \boldsymbol{\Phi}, \qquad \boldsymbol{y} \in ext(\mathcal{P}_f). \tag{8b}$$

where constraints (4b) are replaced with constraints (8b), called *Benders cuts*, an exponential-size family of constraints, one for each extreme point of $\mathcal{P}_f$. This ILP model is called *natural formulation* since it features only the natural variables associated with the arcs and it is denoted by n-ILP$_{\text{B}}$ in the remainder of the article.

To solve n-ILP$_{\text{B}}$, we develop a Branch-and-Benders-Cut approach, i.e., a Branch-and-Cut algorithm where Benders cuts (8b) are separated in the nodes of the branching tree for integer and fractional solutions. This exact algorithm requires defining a *relaxed master problem* (RMP) where the binary variables are replaced with continuous variables taking values between 0 and 1. Only a subset of constraints are included in the RMP in the initialization phase. To check that RMP solutions respect all the Benders cuts or to determine one or more violated constraints which are then added to the RMP, we propose the separation procedure described in the next section.

2.3. Separation of the Benders cuts

Given a (fractional) solution $\boldsymbol{x} \in [0,1]^m$ of the RMP in a Branch-and-Cut node, the *separation problem* for the Benders cuts (8b) requires finding a vector $\boldsymbol{y} \in ext(\mathcal{P}_f)$ such that:

$$\sum_{a \in \delta^+(s)} y_a - \sum_{a \in A} x_a \, y_a > \boldsymbol{\Phi} \tag{9}$$

or to prove that such a vector does not exist, i.e., that all Benders cuts are satisfied by the solution $\boldsymbol{x}$. Inequality (9) can be equivalently rewritten as follows:

$$\sum_{a \in \delta^+(s)} (x_a - 1) \, y_a + \sum_{a \in A \setminus \delta^+(s)} x_a \, y_a < -\boldsymbol{\Phi} \tag{10}$$

and accordingly, it is necessary to find a vector $\boldsymbol{y} \in ext(\mathcal{P}_f)$ leading to the minimum value of the left-hand-side of (10). Let us introduce the *minimum cost circulation problem* (MCCP). The MCCP requires a vector $\mathbf{g} \in \mathbb{Q}^m$ where each element g_a is the cost per unit of flow on the arc $a \in A$ and a vector $\mathbf{b} \in \mathbb{Q}^m$ where each element b_a is the lower bound on the flow value passing on the arc $a \in A$. The MCCP aims to find a minimum cost flow respecting the capacity constraints (1), the flow conservation constraints (2), and the lower bounds imposed on the flow of the arcs. By using the vector of continuous variables $\boldsymbol{y} \in \mathbb{Q}_+^m$ for the flow values on the arcs, the MCCP can be modeled by the following LP formulation:

$$\min_{\boldsymbol{y} \in \mathbb{Q}_+^m} \left\{ \sum_{a \in A} g_a \, y_a : (2), \; b_a \leq y_a \leq c_a, \quad a \in A \right\}. \tag{11}$$

The constraints of this formulation imply the constraints of $\mathcal{P}_f$ and accordingly, the MCCP is used to characterize the complexity of the separation problem in the proof of the following proposition.

Proposition 1. *The separation problem for the Benders cuts (8b) can be solved in strongly polynomial time for (fractional) solutions $\boldsymbol{x} \in [0,1]^m$ of the RMP.*

Proof. We construct a MCCP instance by setting $g_a = (x_a - 1)$ for the arcs $a \in \delta^+(s)$, $g_a = x_a$ for the arcs $a \in A \setminus \delta^+(s)$, $b_{(t,s)} = \boldsymbol{\Phi} + 1$, $b_a = 0$ for every arc $a \in A \setminus \{(t,s)\}$ and the same capacities on the arcs. An optimal solution $\boldsymbol{y}$ of this MCCP instance respects, by construction, the capacity constraints (1), the flow conservation constraints (2) and by setting $b_{(t,s)} = \boldsymbol{\Phi} + 1$, we have $\sum_{a \in \delta^+(s)} y_a \geq \boldsymbol{\Phi} + 1$, i.e., all the constraints of $\mathcal{P}_f$ are satisfied. If the optimal MCCP solution value is strictly smaller than $-\boldsymbol{\Phi}$, then the Benders cut of maximum violation is found and it is associated with $\mathbf{y}$. Otherwise, all Benders cuts are satisfied by solution $\boldsymbol{x}$. The MCCP can be solved in strongly polynomial time, see e.g., Tardos (1985). Accordingly, the separation problem of the Benders cuts (8b) can be solved in strongly polynomial time for any (fractional) solution $\boldsymbol{x}$. □

2.4. A second single-level ILP model for the FB

In this section, we introduce a second ILP formulation using as previously only the natural variables $\boldsymbol{x}$ associated with the arcs. The ILP model presented is a set-covering type formulation that can be used for solving interdiction and blocker problems. We refer the interested reader to Wei and Walteros (2022) for set-covering formulations of the shortest path interdiction problem, the maximum clique vertex interdiction problem, and the minimum spanning tree interdiction. We then describe the exact algorithm that can be developed to solve this second ILP formulation to proven optimality.

For a given vector $\boldsymbol{y} \in ext(\mathcal{P}_f)$, we define the subset of arcs $\mathcal{A}_S(\boldsymbol{y}) \subseteq A$ routing a strictly positive flow in the extreme point $\boldsymbol{y}$ as follows:

$$\mathcal{A}_S(\boldsymbol{y}) = \left\{ a \in A : y_a > 0 \right\}.$$

These arcs induce the *support network* $\mathcal{N}_S(\boldsymbol{y}) = (V, \mathcal{A}_S(\boldsymbol{y}))$ in which, by construction, the maximum flow value $f_m(\mathcal{N}_S(\boldsymbol{y}), \mathbf{c})$ is larger than or equal to $\boldsymbol{\Phi} + 1$. We now introduce the following set of constraints called the *target-flow inequalities*:

$$\sum_{a \in \mathcal{A}_S(\boldsymbol{y})} x_a \geq 1, \qquad \boldsymbol{y} \in ext(\mathcal{P}_f). \tag{12}$$

For any vector $\boldsymbol{y} \in ext(\mathcal{P}_f)$, the associated constraint (12) is valid since it imposes to block at least one arc in the subset $\mathcal{A}_S(\boldsymbol{y}) \subset A$. In other words, the constraint prevents having a flow of value strictly larger than $\boldsymbol{\Phi}$ in the support network $\mathcal{N}_S(\boldsymbol{y})$. Clearly, it is an exponential family of constraints, one for each extreme point $\boldsymbol{y} \in ext(\mathcal{P}_f)$. Accordingly, we obtain the following second natural formulation for the FB, denoted by n-ILP$_{\text{TF}}$. This formulation reads as follows:

$$\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \boldsymbol{\Phi})) = \min_{\boldsymbol{x} \in \{0,1\}^m} \sum_{a \in A} r_a \, x_a \tag{13a}$$

$$\sum_{a \in \mathcal{A}_S(\boldsymbol{y})} x_a \geq 1, \qquad \boldsymbol{y} \in ext(\mathcal{P}_f). \tag{13b}$$

It is worth noticing that the target-flow inequalities share a similar combinatorial structure with the *super-valid inequalities* (SVI) for bilevel network interdiction models, proposed in Wood (2011). A technique to lift the right-hand-side (RHS) of the SVI is described in Wood (2011). A similar technique can be used to lift the RHS of the target-flow inequalities (12), as well. For a given $\boldsymbol{y} \in ext(\mathcal{P}_f)$, in case the following inequality holds:

$$\sum_{a \in \delta^+(s)} y_a - \max_{a,b \in \mathcal{A}_S(\boldsymbol{y})} \{y_a + y_b\} \geq \Phi + 1, \tag{14}$$

the RHS of the target-flow inequality can be set to three. This is because blocking the two arcs routing the maximum flow values in $\mathcal{A}_S(\boldsymbol{y})$ is not sufficient to have a maximum flow value $f_m(\mathcal{N}_S(\boldsymbol{y}), \mathbf{c})$ smaller than or equal to Φ. If inequality (14) holds by subtracting only the maximum flow value of the arcs, then the RHS can be set to two. This lifting can be extended to subsets of three or more arcs. However, in our computational experiments (see Section 4), only subsets of at most two arcs are found with the desired properties. Since finding the largest two values in $\boldsymbol{y}$ can be done in linear time, the lifting can be performed in a very efficient manner but our extensive preliminary results show that it has a limited beneficial impact on the computational performance.

In Wood (2011), the separation problem of the SVI (which is similar to the separation problem of the target-flow inequalities) is not addressed nor is its computational complexity. For this reason, in the next section, we present the separation problem of the target-flow inequalities (12) and characterize its computational complexity. In the computational Section 4, we discuss the impact on the performance of n-ILP$_{\text{TF}}$ determined by different exact and heuristic separation procedures based on the separation problem of the target-flow inequalities.

2.5. Separation of the target-flow inequalities

Given a (fractional) solution $\boldsymbol{x} \in [0,1]^m$ of the RMP in a Branch-and-Cut node, the *separation problem* for the target-flow inequalities (12) requires finding a vector $\boldsymbol{y} \in ext(\mathcal{P}_f)$ such that:

$$\sum_{a \in \mathcal{A}_S(\boldsymbol{y})} x_a < 1, \tag{15}$$

or to prove that such a vector does not exist, i.e., that all target-flow inequalities are satisfied by the solution $\boldsymbol{x}$. Thus, it is necessary to find a vector $\boldsymbol{y} \in ext(\mathcal{P}_f)$ leading to the minimum value of the left-hand-side of (15). We distinguish two cases. The first one is for integer RMP solutions and the second one is for fractional solutions.

For an integer solution $\boldsymbol{x} \in \{0,1\}^m$, i.e., for a blocker policy, the separation problem can be restated as an MF and the next proposition characterizes its computational complexity:

Proposition 2. *The separation problem for the target-flow inequalities (12) can be performed in strongly polynomial time for integer solutions $\boldsymbol{x} \in \{0,1\}^m$ of the RMP.*

Proof. Since $x_a \in \{0,1\}$, $a \in A$, a violated target-flow inequality can be found if and only if an extreme point $\boldsymbol{y} \in ext(\mathcal{P}_f)$ exists such that the subset of arcs $\mathcal{A}_S(\boldsymbol{y}) \subset A$ does not contain any blocked arcs, i.e., arcs $a \in A$ for which $x_a = 1$. Accordingly, finding a violated target-flow inequality can be done by solving the MF in the non-blocked network $\mathcal{N}_{NB}(\boldsymbol{x}) = \big(V, \mathcal{A}_{NB}(\boldsymbol{x})\big)$. In case the maximum flow value $f_m(\mathcal{N}_{NB}(\boldsymbol{x}), \mathbf{c})$ is $\geq \Phi + 1$, an extreme point $\boldsymbol{y} \in ext(\mathcal{P}_f)$ associated to a maximally violated target-flow inequality is found since the left-hand-side of (12) is equal to 0. Otherwise, target flow inequalities are not violated by $\boldsymbol{x}$. The MF can be solved in strongly polynomial time, see e.g., Ahuja et al. (1993) and Olver and Végh (2016), and accordingly, the separation problem of the target-flow inequalities (12) can be solved in strongly polynomial time for any integer RMP solution $\boldsymbol{x}$. □

For fractional RMP solutions $\boldsymbol{x}$, let $\boldsymbol{z} \in \{0,1\}^m$ be a vector of binary variables where each variable z_a is equal to 1 if and only if the arc a is in $\mathcal{A}_S(\boldsymbol{y})$, i.e., if the arc a is used to route a flow in $\boldsymbol{y} \in ext(\mathcal{P}_f)$. The separation problem for the target-flow inequalities can be modeled by the following ILP problem:

$$\min_{\substack{\boldsymbol{z} \in \{0,1\}^m, \\ \boldsymbol{y} \in ext(\mathcal{P}_f)}} \left\{ \sum_{a \in A} x_a\, z_a \;:\; y_a \leq c_a\, z_a, \quad a \in A \right\}. \tag{16}$$

The objective function minimizes the left-hand side of (15). The constraints are the ones of $\mathcal{P}_f$ with the additional constraints to impose to select an arc if it routes a flow in $\boldsymbol{y}$. If the optimal solution value is strictly smaller than 1, a target-flow inequality maximally violated by $\boldsymbol{x}$ is found. Otherwise, no target-flow inequalities are violated by $\boldsymbol{x}$.

Let us introduce the *minimum edge-cost flow problem* (MECF), which is used to characterize the complexity of the separation problem for fractional RMP solutions. Given an *arc-price* vector $\mathbf{p} \in \mathbb{Z}_+^m$ and a *flow-value bound* $R \in \mathbb{Z}_+$, the MECF requires finding a minimum-price flow from s to t with a flow value larger than or equal to R. The MECF is $\mathcal{NP}$-complete in its decision version, see Garey and Johnson (1979), and accordingly, it is $\mathcal{NP}$-hard in its optimization version. It is worth noticing that the MECF remains $\mathcal{NP}$-hard also when p_a is in $[0,1]$. By reducing the MECF to the separation problem, the next proposition characterizes the computational complexity of the latter one.

Proposition 3. *The separation problem for the target-flow inequalities (12) is $\mathcal{NP}$-hard for fractional solutions $\boldsymbol{x} \in [0,1]^m$ of the RMP.*

Proof. Starting from a MECF instance, we set $x_a = p_a$ (a value in $[0,1]$) for every arc $a \in A$ and $\Phi = R - 1$. Once the separation problem (16) is solved, its optimal solution $(\boldsymbol{y}, \boldsymbol{z})$ corresponds to an optimal MECF solution. Since $\boldsymbol{y}$ is in $ext(\mathcal{P}_f)$, all the MECF constraints are satisfied, i.e., the flow conservation constraints, the capacity constraints and the requirement of having a flow value larger than or equal to R. The variable values $\boldsymbol{z}$ correspond to the arcs with a minimum-price flow from s to t. □

Following the idea proposed in Wood (2011), a violated target-flow inequality (12) can be derived from a violated Benders cut (8b), which are both associated with a vector $\boldsymbol{y} \in ext(\mathcal{P}_f)$. More precisely, if $\sum_{a \in \mathcal{A}_S(\boldsymbol{y})} x_a > 1$, a violated target-flow inequality is found and can be added to the RMP. This method for generating violated target-flow inequalities for fractional and integer RMP solutions based on a violated Benders cut is computationally effective, as shown by the tests reported in Section 4.

2.6. Comparison of the strength of the LP relaxations of the natural formulations

This section compares the strength of the LP relaxations of the two natural formulations n-ILP$_{\text{B}}$ and n-ILP$_{\text{TF}}$, for the FB, presented in the previous section. Let us introduce the two polytopes $\mathcal{P}_{n,\text{B}}$ and $\mathcal{P}_{n,\text{TF}}$, defined respectively by the Benders cuts (8b) and the target-flow inequalities (12), which are formally defined as follows:

$$\mathcal{P}_{n,\text{B}} = \{\boldsymbol{x} \in [0,1]^m : \boldsymbol{x} \text{ satisfies (8b)}\}, \quad \mathcal{P}_{n,\text{TF}} = \{\boldsymbol{x} \in [0,1]^m : \boldsymbol{x} \text{ satisfies (12)}\}.$$

The next proposition shows that the lower bounds on $\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi))$ provided by the optimal solution value of the LP relaxations of n-ILP$_{\text{B}}$ and n-ILP$_{\text{TF}}$ do not dominate each other.

Proposition 4. *There are instances of the FB for which, $\mathcal{P}_{n,\text{B}} \not\subset \mathcal{P}_{n,\text{TF}}$, and other ones for which $\mathcal{P}_{n,\text{TF}} \not\subset \mathcal{P}_{n,\text{B}}$.*

Proof. We consider the network N represented in Fig. 2 and we exhibit two instances, one for which the LP relaxation of n-ILP$_{\text{B}}$ is strictly

stronger than the one of n-ILP$_{\text{TF}}$ and another for which the opposite holds.

Setting $\Phi = 7$ ($\approx 19\%$ of the maximum flow), the optimal solution value of the LP relaxation of n-ILP$_{\text{TF}}$ is equal to 16 with $x_{(s,v_3)} = x_{(v_1,v_5)} = x_{(v_2,v_4)} = 1$ and $x_a = 0$ for all other arcs, which corresponds to an optimal FB solution. For the same instance, the optimal solution value of the LP relaxation of n-ILP$_{\text{B}}$ is equal to $\frac{133}{10} < 16$ with $x_{(v_1,v_5)} = x_{(v_2,v_4)} = 1$, $x_{(v_3,v_7)} = \frac{73}{100}$ and $x_a = 0$ for all other arcs. Therefore, this instance shows that $\mathcal{P}_{n,\text{B}} \not\subseteq \mathcal{P}_{n,\text{TF}}$.

We now consider the network N with different blocker costs and arc capacities. We associate with the arcs (v_1, v_5) and (v_3, v_7) a capacity equal to 5. We associate with the arcs $(s, v_1), (s, v_2)$ and (s, v_3) a blocker cost equal to 100. All other arcs have unitary blocker costs and capacities. The maximum flow value is now 3. Setting $\Phi = 1$ ($\approx 34\%$ of the maximum flow), the optimal solution value of the LP relaxation of n-ILP$_{\text{TF}}$ is equal to $\frac{3}{2}$, with $x_{(v_5,t)} = x_{(v_6,t)} = x_{(v_7,t)} = \frac{1}{2}$ and $x_a = 0$ for all other arcs, while the optimal solution value of the LP relaxation of n-ILP$_{\text{B}}$ is equal to $\frac{39}{20} > \frac{3}{2}$ with $x_{(v_5,t)} = x_{(v_6,t)} = x_{(v_7,t)} = \frac{13}{20}$ and $x_a = 0$ for all other arcs. This second instance shows that $\mathcal{P}_{n,\text{TF}} \not\subseteq \mathcal{P}_{n,\text{B}}$. □

2.7. A third single-level ILP model for the FB

A third natural formulation for the FB can be obtained by using both families of constraints, i.e, the Benders cuts (8b) and the target-flow inequalities (12). This formulation is denoted by n-ILP$_{\text{B+TF}}$ and it reads as follows:

$$\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi)) = \min_{\boldsymbol{x} \in \{0,1\}^m} \left\{ \sum_{a \in A} r_a\, x_a \;:\; (8b), (12) \right\}. \tag{17}$$

The performance of all three ILP formulations is empirically evaluated and the results are discussed in the computational Section 4 of this manuscript. Moreover, the strength of the natural formulation is discussed and compared with the additional formulation introduced in the next paragraph in Section 3.3.

3. A compact ILP model for the FB

In this section, we first present a compact ILP formulation for the FB. This formulation, featuring a polynomial number of variables and constraints, is obtained through the "dualize-and-combine" method (see Wei & Walteros, 2022) applied to the bilevel model introduced in Section 2.1, i.e., Model (4) and it exploits the nature of the follower problem. We then present a property that establishes a structural link between the solutions of the FB and the solutions of the maximum flow interdiction problem (FI). This result allows us to derive some complexity results for the FB and the FI.

3.1. A fourth compact ILP model for the FB

By using the vector of binary variables $\boldsymbol{x} \in \{0,1\}^m$ representing the blocked arcs, we derive from Model (4), a single-level ILP formulation exploiting the nature of the value function $\vartheta(\boldsymbol{x})$, as defined by Model (5), in which the dependency on the $\boldsymbol{x}$ variables is in the objective function of the model.

Given $\boldsymbol{x} \in \{0,1\}^m$, the objective function of (5) can be re-written as $y_{(t,s)} - \sum_{a \in A} x_a\, y_a$. This is due to the fact that the flow on the arc (t, s) is equal to the outgoing flow from the source s. Accordingly, the dual of Model (5) reads as follows:

$$\vartheta(\boldsymbol{x}) = \min_{\substack{\boldsymbol{\gamma} \in \mathbb{Q}^n_+,\\ \boldsymbol{\mu} \in \mathbb{Q}^m_+}} \left\{ \sum_{a \in A} c_a\, \mu_a \;:\; \mu_{uv} + \gamma_u - \gamma_v \geq -x_{uv}, \right.$$
$$\left. (u,v) \in A \setminus \{(t,s)\},\ \gamma_t - \gamma_s \geq 1 \right\}. \tag{18}$$

This dual LP model features a vector $\boldsymbol{\mu} \in \mathbb{Q}^m_+$ of m dual non-negative variables associated with the capacity constraints (1) and a vector $\boldsymbol{\gamma} \in \mathbb{Q}^n_+$ of n dual non-negative variables associated with the flow conservation constraints (2). Indeed, the $\boldsymbol{\gamma}$ variables cannot take negative values since an equivalent version of Constraints (2) is given by replacing "$= 0$" with "≤ 0", obtaining in this manner a set of less-than-or-equal-to constraints (thanks to the arc $(t, s) \in A$). The dual Model (18) has a totally unimodular system of constraints and accordingly, there exists an optimal dual solution in which all variables $\boldsymbol{\mu}$ and $\boldsymbol{\gamma}$ take binary values.

By using in (4) the value function $\vartheta(\boldsymbol{x})$ as defined in (18) and two sets of binary variables $\boldsymbol{\mu} \in \{0,1\}^m$ and $\boldsymbol{\gamma} \in \{0,1\}^n$, we obtain the following bilevel reformulation of Model (4):

$$\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi)) = \min_{\boldsymbol{x} \in \{0,1\}^m} \sum_{a \in A} r_a\, x_a \tag{19a}$$
$$\vartheta(\boldsymbol{x}) \leq \Phi, \tag{19b}$$
$$\text{where} \quad \vartheta(\boldsymbol{x}) = \min_{\substack{\boldsymbol{\gamma} \in \{0,1\}^n,\\ \boldsymbol{\mu} \in \{0,1\}^m}} \sum_{a \in A} c_a\, \mu_a \tag{19c}$$
$$\mu_{uv} + \gamma_u - \gamma_v \geq -x_{uv},$$
$$(u,v) \in A \setminus \{(t,s)\} \tag{19d}$$
$$\gamma_t - \gamma_s \geq 1. \tag{19e}$$

It is worth noticing that Constraint (19b) imposes that the optimal solution value of the follower problem ((19c)–(19e)) should not exceed the target flow Φ. We can replace $\vartheta(\boldsymbol{x})$ by the objective function value in (19c), i.e., "$\sum_{a \in a} c_a \mu_a$", since by doing so, we impose that there exists at least one solution of the follower problem ((19c)–(19e)) whose value is smaller than Φ and accordingly, its optimal solution value is also smaller than Φ.

Therefore, we obtain a compact ILP formulation for the FB, which reads as follows:

$$\zeta(\text{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi)) = \min_{\substack{\boldsymbol{x}, \boldsymbol{\mu} \in \{0,1\}^m,\\ \boldsymbol{\gamma} \in \{0,1\}^n}} \sum_{a \in A} r_a\, x_a \tag{20a}$$
$$\sum_{a \in A} c_a\, \mu_a \leq \Phi, \tag{20b}$$
$$\mu_{uv} + x_{uv} + \gamma_u - \gamma_v \geq 0, \qquad (u,v) \in A \setminus \{(t,s)\}, \tag{20c}$$
$$\gamma_t - \gamma_s \geq 1. \tag{20d}$$

This ILP formulation (20), denoted by c-ILP, is called *compact formulation*, since it features a polynomial number of variables and constraints. Accordingly, it can be directly solved using an ILP solver.

It is worth noticing that in any optimal solution, for a given arc $a \in A$, we can have either $\mu_a = 1$ or $x_a = 1$ but not both. This is due to the fact that $r_a > 0$ for all arc $a \in A$. If $x_a = \mu_a = 1$, then the solution obtained by setting $\mu_a = 0$ and keeping other values unchanged is still feasible and does not increase the objective function value. For this reason and due to the nature of constraints (20c) and (20d), an optimal solution of (20) is a cut in the network N, denoted by $\delta(U^G(\boldsymbol{x}))$, which depends on an optimal blocker policy $\boldsymbol{x}$. This cut $\delta(U^G(\boldsymbol{x}))$ is given by the arcs $a \in A$ where $\mu_a = 1$ or $x_a = 1$ and it is the union of the set of non-blocked arcs such that $x_a = 0$ and $\mu_a = 1$ and the set of blocked arcs such that $x_a = 1$ and $\mu_a = 0$. If a variable γ_u is equal to 1, it indicates that vertex u is in the subset $U^G(\boldsymbol{x})$ containing the source s and if it is equal to 0, it indicates that vertex u is in the subset $V \setminus U^G(\boldsymbol{x})$ containing the destination t. In addition, any optimal solution $(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\gamma})$ of Model (20) contains the minimum cut $\delta(U^{\mathcal{N}_{NB}}(\boldsymbol{x}))$ in the non-blocked network $\mathcal{N}_{NB}(\boldsymbol{x})$ which is given by the arcs $a \in \mathcal{A}_{NB}(\boldsymbol{x})$ such that $\mu_a = 1$ and $U^{\mathcal{N}_{NB}}(\boldsymbol{x})$ is a set of vertices containing the source s. This is due to the fact that constraints (20c) of Formulation (20) can be equivalently rewritten as follows:

$$\mu_{uv} + \gamma_u - \gamma_v \geq 0, \quad (u,v) \in \mathcal{A}_{NB}(\boldsymbol{x}), \tag{21}$$

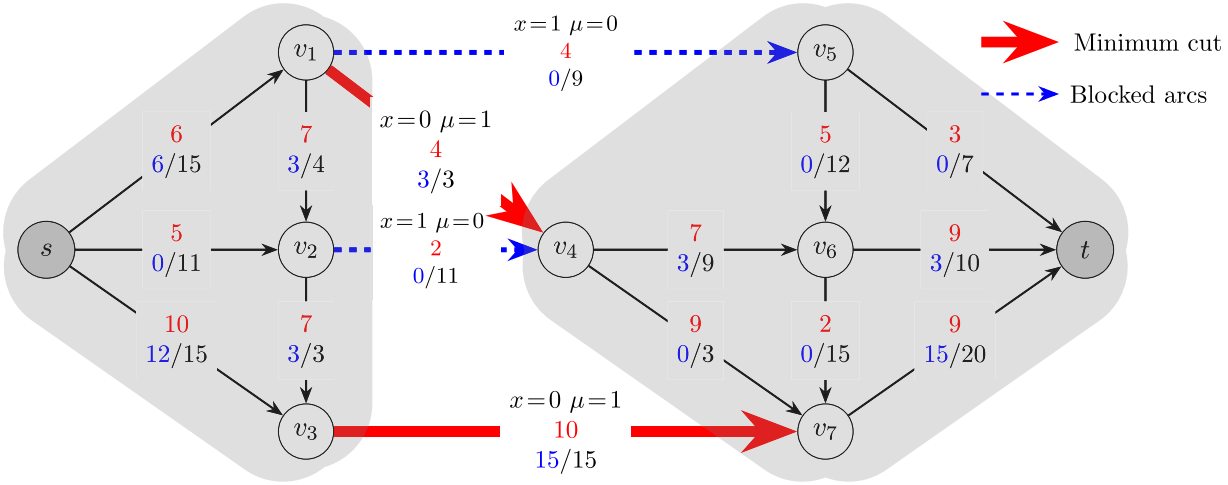

Fig. 3. The cut $\delta(U^N(\boldsymbol{x}))$ (bold red and blue dashed arcs) associated to an optimal FB solution $(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\gamma})$, composed by the blocked arcs (blue dashed arcs) and the minimum cut $\delta(U^{\mathcal{N}_{NB}}(\boldsymbol{x}))$ (red arcs) in the non-blocked network $\mathcal{N}_{NB}(\boldsymbol{x})$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\mathcal{A}_{NB}(\boldsymbol{x})$ are the arcs of $\mathcal{N}_{NB}(\boldsymbol{x})$. Together with constraints (20d), they are the standard MC constraints for the non-blocked network $\mathcal{N}_{NB}(\boldsymbol{x})$.

Fig. 3 illustrates the same optimal FB solution as Fig. 2 with a target flow $\Phi = 23$. In addition, we report on each arc the values of $\boldsymbol{\mu}$ and $\boldsymbol{x}$ variables associated with an optimal solution $(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\gamma})$ of Model (20). The arcs of the minimum cut $\delta(U^{\mathcal{N}_{NB}}(\boldsymbol{x}))$ in the non-blocked network $\mathcal{N}_{NB}(\boldsymbol{x})$ are depicted with thick red lines, i.e., the arcs $a \in A$ with $\mu_a = 1$. The arcs of $\delta(U^G(\boldsymbol{x}))$ are the arcs of $\delta(U^{\mathcal{N}_{NB}}(\boldsymbol{x}))$ together with the blocked arcs depicted with dashed blue lines, i.e., the arcs $a \in A$ with $x_a = 1$.

Fig. 3 also shows that the cut $\delta(U^N(\boldsymbol{x}))$ containing the optimal FB solution does not necessarily correspond to the minimum cut in the network. Specifically, in the depicted example, the optimal FB solution is contained in a cut having a value of 38 and composed by the arcs $(v_1, v_5), (v_1, v_4), (v_2, v_4)$, and (v_3, v_7). In contrast, the minimum cut in the same network has a value of 36 and it is composed by the arcs $(v_1, v_5), (v_4, v_6), (v_4, v_7)$, and (v_3, v_7), as illustrated in Fig. 1.

It is worth noticing that the structure of Model (20) exhibits a notable similarity to the formulation introduced in Wood (1993) for the FI. More precisely, the next section demonstrates the existence of a structural link between the solutions of the FI and the solutions of the FB.

3.2. Solving the FB via the FI

Before discussing the link between the FB and the FI, we present a compact ILP formulation for the FI, proposed in Wood (1993). To this end, let us introduce $\boldsymbol{w} \in \{0,1\}^m$, which is a vector of binary variables associated with the set of arcs A of N, each variable w_a takes value 1 if and only if the arc a is removed from N, i.e., *interdicted*. A binary realization of variables $\boldsymbol{w} \in \{0,1\}^m$ is called an *interdiction policy* and it generates a *non-interdicted network* $\mathcal{N}_{NI}(\boldsymbol{w}) = (V, \mathcal{A}_{NI}(\boldsymbol{w}))$, i.e., the network induced by the set of non-interdicted arcs $a \in A$ with $w_a = 0$ (denoted $\mathcal{A}_{NI}(\boldsymbol{w})$).

In Wood (1993), the author employs a dualize-and-combine approach to derive a compact ILP formulation for the FI starting from its bilevel formulation. However, in contrast with the approach detailed in Section 2.1, the compact formulation of the FI is obtained after computing the dual of the follower problem, which is a MF where the capacity constraints takes into account the action of the leader. In other words, as for the FB, the "standard" capacity constraints (1) of the MF are replaced by the capacity constraints "$y_a \le c_a(1 - w_a)$" for all arcs $a \in A$. By fixing values of the interdiction variables, the capacity constraints of the follower problem become independent of the first-level variables, adopting the standard form with a right-hand side of $c_a(1-w_a)$. This allows to replace the follower problem by its dual formulation, resulting in a single-level reformulation model featuring a quadratic objective function and constraints of the MC. The author then uses linearization techniques, taking advantage of the binary nature of the dual variables.

Using two additional sets of binary variables $\boldsymbol{\beta} \in \{0,1\}^m$ and $\boldsymbol{\alpha} \in \{0,1\}^n$, the compact ILP formulation for the FI of Wood (1993) reads as follows:

$$\zeta(\mathrm{FI}(N, \boldsymbol{c}, \boldsymbol{q}, \Psi)) = \min_{\substack{\boldsymbol{w},\boldsymbol{\beta} \in \{0,1\}^m, \\ \boldsymbol{\alpha} \in \{0,1\}^n}} \sum_{a \in A} c_a \beta_a \tag{22a}$$

$$\sum_{a \in A} q_a w_a \le \Psi, \tag{22b}$$

$$\beta_{uv} + w_{uv} + \alpha_u - \alpha_v \ge 0, \qquad (u,v) \in A \setminus \{(t,s)\}, \tag{22c}$$

$$\alpha_t - \alpha_s \ge 1. \tag{22d}$$

It is worth noticing that constraints (22c) and (22d) share the same structural form as constraints (20c) and (20d), respectively. For this reason, an optimal solution of (22) is also a cut in the network N, denoted by $\delta(U^G(\boldsymbol{w}))$, which depends on an optimal interdiction policy $\boldsymbol{w}$ (see also Royset and Wood (2007)). More precisely, the cut $\delta(U^G(\boldsymbol{w}))$ is given by the arcs $a \in A$ where $\beta_a = 1$ or $w_a = 1$ and it is the union of the set of non-interdicted arcs $\mathcal{A}_{NI}(\boldsymbol{w})$ such that $\beta_a = 1$ and $w_a = 0$ and the set of interdicted arcs such that $w_a = 1$ and $\beta_a = 0$. The minimum cut $\delta(U^{\mathcal{N}_{NI}}(\boldsymbol{w}))$ in the non-interdicted network $\mathcal{N}_{NI}(\boldsymbol{w})$ is given by the arcs $a \in \mathcal{A}_{NI}(\boldsymbol{w})$ such that $\beta_a = 1$. Constraint (22b) imposes that the total interdiction cost, i.e., the interdiction cost induced by the set of interdicted arcs, does not exceed the interdiction budget Ψ.

The next proposition shows how to obtain an optimal FB solution starting from an optimal FI solution, where the interdiction costs are set to the capacities of the arcs, the capacities are set to the blocker costs and the interdiction budget is set to the target flow, by solving a MC in the non-interdicted network.

Proposition 5. *Given an instance $(N, \boldsymbol{c}, \boldsymbol{r}, \Phi)$ of the FB, one of its optimal solutions is given by an optimal solution of the MC in the non-interdicted network $\mathcal{N}_{NI}(\boldsymbol{w})$ given by an optimal interdiction policy $\boldsymbol{w}$ of the FI instance $(N, \boldsymbol{r}, \boldsymbol{c}, \Phi)$.*

Proof. Let $(N, \boldsymbol{c}, \boldsymbol{r}, \Phi)$ and $(N, \boldsymbol{c}, \boldsymbol{q}, \Psi)$ be an instance of the FB and the FI, respectively. We set $q_a = c_a$ and $c_a = r_a$ for every arc $a \in A$ and $\Psi = \Phi$. Once the FI (see Model ((22a)–(22d))) is solved, its optimal solution $(\boldsymbol{w}, \boldsymbol{\beta}, \boldsymbol{\alpha})$ corresponds to an optimal FB solution. Indeed, since $\boldsymbol{w} \in \{0,1\}^m, \boldsymbol{\beta} \in \{0,1\}^m$ and $\boldsymbol{\alpha} \in \{0,1\}^n$, the constraints of the FI, i.e, constraints (22b), (22c) and (22d) become identical to the constraints of the FB, i.e, Constraints (20b), (20c) and (20d), respectively. The variable values $\boldsymbol{\beta}$ corresponds to the blocked arcs, while the variable values $\boldsymbol{w}$ corresponds to the arcs remaining in the cut $\delta(U^N(\boldsymbol{\beta}))$. According to the objective function (22a) of the FI, the total blocker cost of the blocked arcs will be minimized, leading to the optimal solution value of the FB. Moreover, when solving a FI, the variable values $\boldsymbol{\beta}$ corresponds to the arcs of the minimum cut in the non-interdicted network $\mathcal{N}_{NI}(\boldsymbol{w})$, as explained previously. Accordingly, the blocked arcs corresponds to the minimum cut in the non-interdicted network $\mathcal{N}_{NI}(\boldsymbol{w})$, which ends the proof. □

The same reasoning can be used to prove, as stated in the next proposition, that an optimal FI solution can be found starting from an optimal FB solution, where the blocker costs are set to the capacities of the arcs, the capacities are set to the interdiction costs and the target flow is set to the interdiction budget, by solving a MC in the non-blocked network.

Proposition 6. *Given an instance $(N, \boldsymbol{c}, \boldsymbol{q}, \Psi)$ of the FI, one of its optimal solution is given by an optimal solution of the MC in the non-blocked network $\mathcal{N}_{NB}(\boldsymbol{x})$ given by an optimal blocker policy $\boldsymbol{x}$ of the FB instance $(N, \boldsymbol{q}, \boldsymbol{c}, \Psi)$.*

Proof. The proof of this proposition can be readily adapted from the proof developed for Proposition 5. □

Proposition 5 and Proposition 6 establish a structural relationship between the solutions of the blocker and the interdiction variant of the MF. More precisely, it proves that the solutions of the FB can be obtained using any exact algorithm designed for the FI, and then solving a MC in the non-interdicted network, and vice versa. Finally, for the optimal solution values of the FB and the FI, we have the following corollary:

Corollary 1. *Given an instance $(N, \boldsymbol{c}, \boldsymbol{r}, \Phi)$ of the FB, its optimal solution value $\zeta(\mathrm{FB(N}, \boldsymbol{c}, \boldsymbol{r}, \Phi))$ is equal to the maximum flow value $f_m(\mathcal{N}_{NI}(\boldsymbol{w}), \mathbf{c})$ in the non-interdicted network $\mathcal{N}_{NI}(\boldsymbol{w})$ given by an optimal interdiction policy $\boldsymbol{w}$ of the FI instance $(N, \boldsymbol{r}, \boldsymbol{c}, \Phi)$, i.e., we have:*

$$\zeta(\mathrm{FB(N}, \boldsymbol{c}, \boldsymbol{r}, \Phi)) = f_m(\mathcal{N}_{NI}(\boldsymbol{w}), \mathbf{c}). \tag{23}$$

Given an instance $(N, \boldsymbol{c}, \boldsymbol{q}, \Psi)$ of the FI, its optimal solution value $\zeta(\mathrm{FI(N}, \boldsymbol{c}, \boldsymbol{q}, \Psi))$ is equal to the maximum flow value $f_m(\mathcal{N}_{NB}(\boldsymbol{x}), \mathbf{c})$ in the non-blocked network $\mathcal{N}_{NB}(\boldsymbol{x})$ given by an optimal blocker policy $\boldsymbol{x}$ of the FB instance $(N, \boldsymbol{q}, \boldsymbol{c}, \Psi)$, i.e., we have:

$$\zeta(\mathrm{FI(N}, \boldsymbol{c}, \boldsymbol{q}, \Psi)) = f_m(\mathcal{N}_{NB}(\boldsymbol{x}), \mathbf{c}). \tag{24}$$

In Fig. 4, we represent an optimal FI solution on the same network shown in Fig. 2, where the interdiction costs equal the capacities of the arcs and the capacities of the arcs equal the blocker costs. We report on each arc $a \in A$, the interdiction cost in red, the arc capacity in black and the maximum flow value of the non-interdicted network in blue. For the arcs in the cut $\delta(U^N(\boldsymbol{w}))$ of N associated to an optimal FI solution $(\boldsymbol{w}, \boldsymbol{\beta}, \boldsymbol{\alpha})$ with an interdiction budget $\Psi = 23$, we also report the values of the $\boldsymbol{w}$ and $\boldsymbol{\beta}$ variables (without reporting the arcs). The arcs of the minimum cut $\delta(U^{\mathcal{N}_{NI}}(\boldsymbol{w}))$ in $\mathcal{N}_{NI}(\boldsymbol{w})$ are depicted with thick (bold) red lines, i.e., the arcs (v_1, v_5) and (v_2, v_4). The arcs of $\delta(U^G(\boldsymbol{w}))$ are the arcs of $\delta(U^{\mathcal{N}_{NI}}(\boldsymbol{w}))$ together with the interdicted arcs depicted with dashed blue lines, i.e., the arcs (v_1, v_4) and (v_3, v_7).

Fig. 4 illustrates Proposition 5. Indeed, the minimum cut $\delta(U^{\mathcal{N}_{NI}}(\boldsymbol{w}))$ corresponds to the set of blocked arcs in the optimal FB solution represented in Fig. 2. In this FB example, the target flow is equal to the interdiction budget, the blocker costs are equal to the arc-capacities and the arc-capacities are equal to the interdiction costs. Moreover, the optimal solution value of the FB example represented in Fig. 2 is equal to 6. As stated by Corollary 1, this value corresponds to the maximum flow remaining in the non-interdicted network shown in Fig. 4.

3.3. Comparison of the strength of the LP relaxations of the formulations

This section compares the strength of the LP relaxations of three ILP formulations for the FB, presented in the previous sections, namely n-ILP$_{\mathrm{B}}$, n-ILP$_{\mathrm{B+TF}}$ and c-ILP.

The next proposition states that there are instances for which the lower bound on $\zeta(\mathrm{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi))$ provided by the optimal solution value of the LP relaxation of c-ILP is stronger than the lower bound provided by the optimal solution value of the LP relaxation of n-ILP$_{\mathrm{B}}$.

Proposition 7. *There are instances of the FB for which the LP relaxation of c-ILP (20) is stronger than the LP relaxation of Model n-ILP$_{\mathrm{B}}$ (8).*

Proof.

To prove that the LP relaxation of c-ILP (20) can be strictly stronger than the LP relaxation of n-ILP$_{\mathrm{B}}$ (8), we consider the example network of Fig. 2 with a target flow $\Phi = 14$. Let $\boldsymbol{x}$ be an optimal solution of the linear relaxation of the natural formulation n-ILP$_{\mathrm{B}}$ (8), with $x_{(v_1,v_5)} = \frac{19}{20}$, $x_{(v_2,v_4)} = \frac{81}{100}$, $x_{(v_7,t)} = \frac{1}{5}$ and $x_a = 0$ for all other arcs $a \in A$. The value of this solution is equal to $\frac{361}{50}$. On the other hand, the optimal solution value of the linear relaxation provided by the compact formulation c-ILP (20) is equal to $\frac{42}{5} > \frac{361}{50}$, with $x_{(v_1,v_5)} = x_{(v_2,v_4)} = 1$ and $x_{(v_3,t)} = \frac{6}{25}$. □

Extensive preliminary results showed that in all tested instances, the LP relaxation of c-ILP (20) is at least as strong as the LP relaxation of Model n-ILP$_{\mathrm{B}}$ (8). We conjecture that, at least for the types of instances used in our tests, this is always the case. However, we have not yet managed to prove this result theoretically. In Appendix, we report a table showing the strength of both LP relaxations.

Let us introduce the two polytopes $\mathcal{P}_{n,\mathrm{B+TF}}$ and $\mathcal{P}_c$ defined by the constraints of n-ILP$_{\mathrm{B+TF}}$ (17) and c-ILP (20), respectively, which are formally defined as follows:

$$\mathcal{P}_{n,\mathrm{B+TF}} = \{\boldsymbol{x} \in [0,1]^m \ : \ \boldsymbol{x} \text{ satisfies (8b) and (12)}\},$$

$$\mathcal{P}_c = \{\boldsymbol{x} \in [0,1]^m, \boldsymbol{\mu} \in [0,1]^m, \boldsymbol{\gamma} \in [0,1]^n \ : \ \boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\gamma} \text{ satisfy (20b), (20c) and (20d)}\}.$$

The next proposition shows that the lower bounds on $\zeta(\mathrm{FB}(N, \boldsymbol{c}, \boldsymbol{r}, \Phi))$ provided by the optimal solution value of the LP relaxations of c-ILP and n-ILP$_{\mathrm{B+TF}}$ do not dominate each other.

Proposition 8. *There are instances of the FB for which, $\mathcal{P}_c \not\subset \mathcal{P}_{n,\mathrm{B+TF}}$ and other ones for which $\mathcal{P}_{n,\mathrm{B+TF}} \not\subset \mathcal{P}_c$.*

Proof. Let us first consider the network N represented in Fig. 2. Setting Φ equal to 7 ($\approx 19\%$ of the maximum flow), the optimal solution value of the LP relaxation of the compact formulation c-ILP is equal to $\frac{63}{5}$, with $x_{(v_1,v_5)} = x_{(v_2,v_4)} = 1$, $x_{(v_7,t)} = \frac{66}{5}$ and $x_a = 0$ for all other arcs. For the same instance, the optimal solution value of the LP relaxation of n-ILP$_{\mathrm{B+TF}}$ is equal to $15 > \frac{63}{5}$, with $x_{(v_1,v_5)} = x_{(v_2,v_4)} = x_{(v_7,t)} = 1$ and $x_a = 0$ for all other arcs, which corresponds to an optimal FB solution. This proves that for this FB instance, we have $\mathcal{P}_c \not\subset \mathcal{P}_{n,\mathrm{B+TF}}$.

We now consider the same network N with different values for the arc-capacities and the blocker costs. As previously (see Section 2.6), the

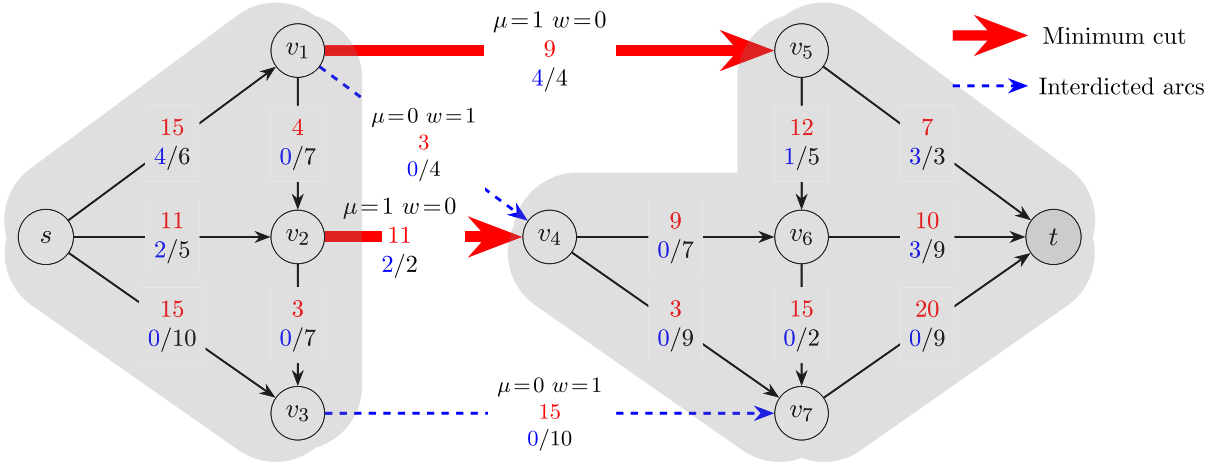

Fig. 4. The cut $\delta(U^N(\boldsymbol{w}))$ (bold red and blue dashed arcs) associated to an optimal FI solution $(\boldsymbol{w}, \boldsymbol{\beta}, \boldsymbol{\alpha})$ with an interdiction budget $\Psi = 23$ and the minimum cut $\delta(U^{\mathcal{N}_{NI}}(\boldsymbol{w}))$ (red arcs) in the non-interdicted network $\mathcal{N}_{NI}(\boldsymbol{w})$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

arcs (v_1, v_5) and (v_3, v_7) have a capacity equal to 5, the arcs $(s, v_1), (s, v_2)$ and (s, v_3) have a blocker cost equal to 100 and all other arcs have unitary blocker costs and capacities. Setting $\Phi = 1$, the optimal solution value of the LP relaxation of n-ILP$_{\text{B+TF}}$ is equal to $\frac{39}{20}$, with $x_{(v_5,t)} = x_{(v_6,t)} = x_{(v_7,t)} = \frac{13}{20}$ and $x_a = 0$ for all other arcs. For the same instance, the optimal solution value of the LP relaxation of c-ILP is equal to $2 > \frac{39}{20}$ with $x_{(v_1,v_5)} = x_{(v_6,t)} = 1$ and $x_a = 0$ for all other arcs, which corresponds to an optimal FB solution. This proves that for this second FB instance, $\mathcal{P}_{n,\text{B+TF}} \not\subset \mathcal{P}_c$. □

3.4. Complexity results for the FB and the FI

We now investigate the complexity of the FI and the FB depending on the values of the capacities, the blocker costs, and the interdiction costs on the arcs.

It is worth noticing that the FI and the FB have the same decision problem, which can be expressed as follows: Given a network N with a source $s \in V$ and a destination $t \in V$, where each arc $a \in A$ has a positive integer capacity c_a and a positive integer cost q_a, and two positive integer values Ψ and Φ, does there exist a subset of arcs $\mathcal{A} \subseteq A$ such that $\sum_{a\in\mathcal{A}} q_a \leq \Psi$ and whose removal does not allow a flow between s and t of value greater than Φ. Consequently, any complexity result for the FI also holds for the FB and vice versa.

In Wood (1993), the author proves that the decision problem of the FI is strongly $\mathcal{NP}$-complete even if all arcs have the same interdiction cost. Accordingly, by setting $r_a = q_a$ for all arcs $a \in A$, we can state that the FB is also strongly $\mathcal{NP}$-complete in its decision version and strongly $\mathcal{NP}$-hard in its optimization version, even if all arcs have the same blocker cost. Moreover, by Proposition 5, an optimal solution for the FB instance where all arcs have the same capacity can be found in polynomial time. This is achieved by solving a MC in the non-interdicted network associated to an optimal solution for an FI instance where all arcs have the same interdiction cost. Therefore, we can deduce that the FB is also strongly $\mathcal{NP}$-hard, even if all arcs have the same capacity. Finally, as the FI and the FB share the same decision problem, the FI is also strongly $\mathcal{NP}$-hard, even if all arcs have the same capacity.

4. Computational results

In this section, we present the results of our computational campaign. The aim is to assess the performance of the ILP formulations for the FB presented in this article, i.e., the natural formulations n-ILP$_{\text{B}}$ (see Model (8)), n-ILP$_{\text{TF}}$ (see Model (13)), n-ILP$_{\text{B+TF}}$ (see Model (17)) and the compact formulation c-ILP (see Model (20)). The first three models feature an exponential number of constraints. Accordingly, they are solved via a Branch-and-Cut algorithm. The last model has a polynomial number of variables and constraints. It is solved directly using an ILP solver. Moreover, since there are no exact algorithms existing in the literature for addressing the blocker variant of the MF, the current state-of-the-art approach involves employing an online-accessible solver designed for bilevel programs. Therefore, we compare this technique to the ILP formulations proposed in the paper.

The specific goals of our computational campaign are threefold. The first is to assess the best configuration of the Branch-and-Cut algorithms for the natural formulations n-ILP$_{\text{B}}$ and n-ILP$_{\text{TF}}$ and to determine the potential effectiveness of enhancing n-ILP$_{\text{B}}$ with a second set of constraints, namely the target-flow inequalities (12) (see Model n-ILP$_{\text{B+TF}}$ (17)). The second involves conducting performance comparisons between the best Branch-and-Cut algorithm for the natural formulations and the direct use of an ILP solver on the compact formulation c-ILP. In addition, we computationally analyze the quality of the FB lower bounds provided by the Linear Programming (LP) relaxations of n-ILP$_{\text{B}}$, n-ILP$_{\text{TF}}$, n-ILP$_{\text{B+TF}}$ and c-ILP. The third and final goal is to determine the maximum size of FB instances that can be solved to proven optimality using the exact methods presented in this paper. Specifically, we investigate the practical computational difficulty of the main features of FB instances.

The experiments are conducted on a processor Intel Core i5-3340M CPU of 2.70 GHz × 4. To tackle the ILP formulations, we use `CPLEX 22.1.1` (called `CPLEX` for brevity) in `C++`, one of the state-of-the-art commercial solvers for ILP problems. The Branch-and-Cut for n-ILP$_{\text{B}}$, n-ILP$_{\text{TF}}$ and n-ILP$_{\text{B+TF}}$ is implemented using the `CONCERT TECHNOLOGY` of `CPLEX` and c-ILP is directly solved using this solver. In our experiments, all computations are performed in a single-thread mode with default values for all `CPLEX` parameters. We employ the open-source `C++` library, called `LEMON`, for implementing network data structures. For a detailed description of the library, we refer the interested readers to Dezső et al. (2011). `LEMON` provides a comprehensive set of efficient components specifically designed for networks and network algorithms. For example, we use `LEMON` to solve maximum flow problems.

4.1. Benchmark set of FB instances

To the best of our knowledge, this is the first article that presents exact methods to solve the FB and no instances can be found in the literature. We have created synthetic FB instances and produced some starting from instances of the FI that have been proposed in Royset and Wood (2007). The new FB instances are characterized by two

main components: (*i*) the *network-structural* features and (*ii*) the *flow-blocker* features. Creating a well-diversified and representative set of FB instances is a challenging task. We explain in what follows the specific choices we adopted in creating our benchmark set of FB instances. The network-structural features of FB instances concern the characteristics of the input network $N = (V, A)$ on which the FB is addressed. The two principal ones are the number n of vertices in V and the number m of arcs in A. The arc set A is also characterized by an arc-density $d(N) = |A|/(n^2 - n)$. In addition, it is necessary to determine a source node $s \in V$ and a destination node $t \in V$. FB instances, as well as FI instances, have capacities on the arcs. The distribution of the arc capacities determines the value $f_m(N, \mathbf{c})$ of the maximum flow between the source s and the destination t. The flow-blocker features of FB instances consist in determining the blocker cost $r_a \in \mathbb{Z}_+$ for each arc $a \in A$ and the target flow $\Phi \in \mathbb{Z}_+$. We have chosen to determine the target flow by setting it as a percentage $\lambda \in [0, 1]$ of the maximum flow value $f_m(N, \mathbf{c})$ in the given instances. By opting for this approach, we can conduct an analysis of the computational complexity of FB instances based on the maximum flow value allowed in the non-blocked network. It is worth noticing that if $\lambda = 0$, the value of the target flow Φ is equal to 0, which implies that no flow between s and t should remain in the non-blocked network. In this case, the FB can be reduced to a MC. On the other hand, if $\lambda = 1$, the value of the target flow Φ is equal to the maximum flow value in N, i.e., $\Phi = f_m(N, \mathbf{c})$, implying that there is no need to remove any arcs from N. The distribution of the blocker costs on the arcs determines the value of the optimal solution for the FB.

As far as the network-structural features of FB instances are concerned, we consider three classes of instances: (*i*) `GRID`, (*ii*) `REAL-NETWORKS`, (*iii*) `SYNTHETIC`.

The first two classes are the instances used in the computational study conducted in Royset and Wood (2007). The third class was generated from a random network generator that we implemented in `Python 3.8.10`.

1. The `GRID` class consists in rectangular grid networks where the vertices are connected through a grid structure of n_1 rows and n_2 columns such that $n = n_1 n_2 + 2$ and $m = 2n_1 + 2((n_2 - 1)n_1 + (n_1 - 1)n_2)$. On opposite sides of the grid, we have the source s and the destination t. The arcs connecting either the source or the destination have infinite capacities. For the remaining arcs, their capacities are randomly drawn from a discrete uniform distribution on $[1, 49]$. In addition, the arcs outgoing the source or entering the destination cannot be blocked and have infinite blocker costs. All other arcs have a blocker cost generated by selecting a number from the discrete uniform distribution on $[1, 3]$. This class contains a total of 17 networks.
2. The `REAL-NETWORKS` class consists of networks representing highways and roads of two cities in the northeastern United States. In these instances, there are multiple source vertices and destination vertices, connected respectively to a super-source s and a super-destination t with infinite blocker costs and infinite capacities. This class is categorized into two types, denoted as `Type A` and `Type B`, featuring the same network structure of 3670 vertices and 9876 arcs but different source and destination vertices. Each type consists of seven instances, with distinct values assigned to the blocker costs and capacities on the arcs. For every arc that is not connected to the super-source or to the super-destination, the blocker cost can take three possible values; 1, 2, or 3 and the capacity is drawn randomly from the discrete uniform distribution $[1, 49]$. This class contains a total of 14 networks.
3. The `SYNTHETIC` class consists of Erdős–Rényi random networks denoted as $G(n, p)$, where the parameter p determines the probability of including each arc. In this case, the value of p is set to 0.5, meaning that each arc has a 50% chance of being included in the network. We consider a large set of instance sizes based on the number of vertices n in the network ranging from 20 to 1500. Specifically, n takes the following values: 20, 30, 40, 50, 60, 100, 300, 500, 700, 900, 1000, 1100, 1200, 1300, 1400 and 1500. The size of the instances also varies depending on the arc densities $d(N)$, which range from 0.1 to 0.9 with a step size of 0.1. For each combination of n and $d(N)$, we create a total of 5 distinct Erdős–Rényi networks using different random seed generators. The source s and the destination t are randomly selected between the vertices of the networks such that s and t are linked by at least three consecutive arcs. As for the two previous classes, the arcs outgoing s or entering t have infinite capacities and blocker costs. All the other arcs have a capacity value (resp. blocker cost value) generated by drawing a number from the range [1, 49] (resp. [1, 3]) uniformly at random and rounding it to the nearest integer value. This class contains a total of 420 networks.

It is worth noticing that for the GRID and REAL-NETWORKS instances, we followed the distributions used in Royset and Wood (2007). For the SYNTHETIC instances, we implemented a different distribution approach to explore alternative configurations. We believe this approach aligns more closely with the construction method of synthetic networks, where arcs are added probabilistically. Moreover, as stated at the beginning of the paper, all instances have an arc (t, s) connecting the destination t to the source s. This arc is given an infinite capacity. Accordingly, this arc does not impact the value of the maximum flow in the network. Moreover, it has an infinite blocker cost, ensuring that it cannot be blocked. Consequently, the arc (t, s) does not affect the solution of the FB.

In the remainder of this paper, all computing times are expressed in seconds. We impose a CPU time limit of 600 s for every instance. If this time limit is exceeded, it is reported as *t.l.* in the computational results. Detailed results as well as the instances can be found on GitHub.[1]

4.2. Computational performance of the natural formulations

In this section, we study and discuss the computational performance of the natural formulations n-ILP$_{\text{B}}$ (8), n-ILP$_{\text{TF}}$ (13) and n-ILP$_{\text{B+TF}}$ (17). The aim of this section is to determine the best Branch-and-Cut algorithm to solve a natural formulation for the FB.

The natural formulation n-ILP$_{\text{B}}$ is solved using a Branch-and-Benders-Cut algorithm where Benders cuts are separated at every node of the branching tree and added to the RMP. As explained in Section 2.3, this can be done by solving a MCCP, see Model (11), using `CPLEX` LP solver. It is worth noticing that the RMP is initialized with one Benders cut that is associated with the maximum flow in N. This flow is, by definition, a flow of value greater than Φ. We recall that we separate Benders cuts both for integer and fractional solutions.

In the case of n-ILP$_{\text{TF}}$, the separation problem for target flow inequalities is $\mathcal{NP}$-hard for fractional solutions (see Proposition 3). Consequently, an efficient separation procedure is crucial. In light of this, we have explored several strategies to separate target flow inequalities (12). The first approach involves separating fractional solutions by solving the exact separation problem (see Model (16)). Additionally, alternative methods have been considered, wherein a feasible solution of the separation problem (16) is obtained instead of the optimal solution. To this end, we can use an approximation algorithm with a performance guarantee of $\Phi + 1$ designed in Krumke et al. (1998) for the MECF. The general idea of this approximation algorithm relies on solving a MCCP in the network. Lastly, we investigate an algorithm to heuristically separate fractional solutions. This algorithm is a heuristic based on successive resolutions of shortest path problems (SPP). We refer the interested reader to Ahuja et al. (1993) for further details on the SPP. After conducting experiments on these separation strategies,

[1] https://github.com/ismabentoumi/MaximumFlowBlockerProblem

Table 1
Performance comparison of Branch-and-Cut for n-ILP$_B$, n-ILP$_{B+TF^*}$ and the bilevel solver (BISOLVER) on SYNTHETIC instances.

			n-ILP$_B$				n-ILP$_{B+TF^*}$					BISOLVER		
				time		#B		time		#B	#TF		time	
n	$d(N)$	#	#opt	avg.	max	avg.	#opt	avg.	max	avg.	avg.	#opt	avg.	max
40	0.2	15	15	0.13	0.32	20	15	0.00	0.09	17	1	15	4.53	19.31
	0.4	15	15	0.07	0.21	64	15	0.25	0.56	48	7	4	495.17	t.l.
	0.6	15	15	0.99	10.98	123	15	0.77	4.62	149	20	1	592.99	t.l.
	0.8	15	15	3.04	10.12	287	15	2.69	8.73	357	41	0	t.l.	t.l.
50	0.2	15	15	1.7	54.17	81	15	0.12	2.87	22	2	12	110.54	t.l.
	0.4	15	15	1.97	45.32	124	15	3.53	30.27	79	17	8	t.l.	t.l.
	0.6	15	15	21.98	100.51	183	15	22.86	76.43	204	24	3	t.l.	t.l.
	0.8	15	15	30.41	331.98	337	15	66.21	210.09	410	54	0	t.l.	t.l.
60	0.2	15	15	12.06	88.65	102	15	0.36	4.40	58	16	8	t.l.	t.l.
	0.4	15	15	21.43	95.67	305	15	5.26	41.14	299	58	0	t.l.	t.l.
	0.6	15	14	45.78	t.l.	853	15	43.42	145.32	937	82	0	t.l.	t.l.
	0.8	15	14	87.98	t.l.	1151	14	114.09	t.l.	1042	105	0	t.l.	t.l.
100	0.2	15	13	86.45	t.l.	1011	14	67.42	t.l.	974	307	0	t.l.	t.l.
	0.4	15	11	102.12	t.l.	1321	11	151.69	t.l.	1111	85	0	t.l.	t.l.
	0.6	15	8	209.98	t.l.	1431	8	326.38	t.l.	1616	93	0	t.l.	t.l.
	0.8	15	6	299.09	t.l.	1549	7	416.16	t.l.	1698	133	0	t.l.	t.l.
300	0.2	15	0	t.l.	t.l.	2011	6	336.94	t.l.	1298	210	0	t.l.	t.l.
	0.4	15	0	t.l.	t.l.	2351	0	t.l.	t.l.	1176	176	0	t.l.	t.l.
	0.6	15	0	t.l.	t.l.	2431	0	t.l.	t.l.	1876	308	0	t.l.	t.l.
	0.8	15	0	t.l.	t.l.	2549	0	t.l.	t.l.	2187	237	0	t.l.	t.l.
Total		300	216				225					28		

we observed that the overall computing time of the Branch-and-Cut algorithm for n-ILP$_{TF}$ significantly exceeds the time required for solving n-ILP$_B$. This can be explained by two main reasons. First, solving the separation problem (16) for the target-flow inequalities (12) can be time-consuming. Second, the approximation algorithm or the heuristic proposed may provide poor-quality solutions, thereby compromising the overall solution of the RMP in the Branch-and-Cut algorithm.

In the same line, we observed for n-ILP$_{B+TF}$ that the performance of the Branch-and-Cut algorithm is negatively impacted by the separation of the target-flow inequalities. However, further use of target-flow inequalities can be derived from the separation of the Benders cuts. Indeed, it is worth noticing that the separation of target-flow inequalities (12) as well as the separation of Benders cuts (8b) require finding a flow of value greater than Φ. Consequently, for every Benders cut generated, one may check if the associated target-flow inequality is violated. If so, then this target-flow inequality is added to the RMP. We denote by n-ILP$_{B+TF^*}$ the natural formulation featuring Benders Cuts (8b) and target flow inequalities (12), solved by separating Benders Cuts and incorporating target-flow inequalities within the Branch-and-Cut algorithm. As the results obtained with n-ILP$_{TF}$ and n-ILP$_{B+TF}$ when separating the target-flow inequalities do not demonstrate effective performance, we do not report them in this section.

As mentioned in the beginning of the section, we also compare the performance of the natural formulations against the direct use of a state-of-the-art exact solver for mixed-integer bilevel linear programs (MIBLPs), available online.[2] This solver is associated with the work of Fischetti et al. (2017), where the authors propose a proof-of-concept exact MIBLP solver based on a B&C MILP approach, strengthened with new families of cuts, namely intersection cuts (see Fischetti et al. (2018)) and hypercube intersection cuts. Additionally, the solver incorporates a preprocessing procedure and many other features including heuristics, propagations, multi-threading support, and LP parametrization. For our experiments, we use the setting MIX++, which represents the robust default setting for the B&C algorithm employed by the solver. MIX++ separates intersection cuts at each separation call using the separation procedure described in Fischetti et al. (2017) and the extended bilevel-free set of Xu (2012). It also applies preprocessing to the follower problem. By the tests made with the natural formulations, the solver is used in a single-thread mode. This resolution method is denoted by BISOLVER.

Table 1 compares performance of the bilevel solver, i.e, BISOLVER, and the two Branch-and-Cut algorithms for solving n-ILP$_B$ and n-ILP$_{B+TF^*}$. The instances used for this analysis consist of networks with $40, 50, 60, 100$ and 300 vertices from the SYNTHETIC class with density values of $0.2, 0.4, 0.6$ and 0.8. We consider three values of the target flow defined by $\lambda = 0.2$, $\lambda = 0.6$ and $\lambda = 0.9$. Each row in the table reports the results of a group of tests conducted on five similar networks (same vertices and arcs) with distinct blocker costs and capacities on the arcs. The column # reports the total number of instances per row. For each formulation, we report the following values computed for every group of tests: the number of instances solved to proven optimality (#opt), the average (avg.) and maximum (max) computing time (time) in seconds and the average number of Benders cuts generated to separate integer and fractional infeasible points (#B). For n-ILP$_{B+TF^*}$, we also report the average number of target-flow inequalities generated (#TF).

Table 1 illustrates that the two Branch-and-Cut algorithms proposed for n-ILP$_B$ and n-ILP$_{B+TF^*}$ outperform the direct use of the bilevel solver. Indeed, with BISOLVER, only 28 instances out of 300 have been solved to proven optimality, which constitutes less than 1% of the instances. The time limit is reached even for instances with 40 vertices and a density of 0.4. In the meanwhile, the two natural formulations begin to encounter challenges for networks with 60 vertices and a density of 0.6. Remark that, at this stage, the bilevel solver fails to solve any instance. In addition, n-ILP$_{B+TF^*}$ achieves better computational performance than n-ILP$_B$. Indeed, n-ILP$_{B+TF^*}$ manages to solve 225 instances out of 300 against 216 for n-ILP$_B$. This result is also supported by the consistently lower computing time of n-ILP$_{B+TF^*}$ compared to n-ILP$_B$. Furthermore, we observe that incorporating target flow inequalities into the Branch-and-Cut algorithm reduces the number of Benders cuts generated and allows us to reach the optimal solution faster. Indeed, even if the average computing time spent on the separation of Benders cuts is remarkably short (approximately 0.01 s), for some instances, a significant number of Benders cuts need to be generated to reach the optimal solution, which leads to an increase in the total computing time. Finally, it is worth noticing that all Benders cuts are not generated in the root node of the branching tree. Moreover, the separation of Benders cuts is always performed with a violation tolerance of 10^{-6}.

[2] https://msinnl.github.io/pages/bilevel.html

Table 2
Performance comparison between c-ILP and n-ILP$_{B+TF^*}$ on SYNTHETIC instances.

			n-ILP$_{B+TF^*}$				c-ILP			
				time		nodes		time		nodes
n	$d(N)$	#	# opt	avg.	max	avg.	# opt	avg.	max	avg.
40	0.2	15	15	0.02	0.09	3.01	15	0.00	0.01	0.00
	0.4	15	15	0.25	0.56	8.53	15	0.00	0.03	0.00
	0.6	15	15	0.77	4.62	10.2	15	0.00	0.01	0.00
	0.8	15	15	2.69	8.73	21.02	15	0.01	0.05	0.00
50	0.2	15	15	0.12	2.87	22.32	15	0.02	0.1	0.00
	0.4	15	15	3.53	30.27	42.39	15	0.04	0.05	0.00
	0.6	15	15	22.86	76.43	98.21	15	0.06	0.09	0.00
	0.8	15	15	66.21	210.09	88.21	15	0.09	0.13	0.00
60	0.2	15	15	0.36	4.4	65.21	15	0.06	0.08	0.00
	0.4	15	15	5.26	41.14	59.32	15	0.13	0.22	0.00
	0.6	15	15	43.42	145.32	102.32	15	0.19	0.34	0.00
	0.8	15	14	114.09	t.l.	110.35	15	0.28	0.53	0.00
100	0.2	15	14	67.42	t.l.	202.25	15	0.16	0.28	0.00
	0.4	15	11	151.69	t.l.	251.36	15	0.33	1.02	0.00
	0.6	15	8	326.38	t.l.	365.87	15	0.54	1.14	0.00
	0.8	15	7	416.16	t.l.	401.58	15	0.98	1.73	0.00
300	0.2	15	6	336.94	t.l.	587.23	15	0.25	0.53	0.00
	0.4	15	0	t.l.	t.l.	58.69	15	0.36	1.02	0.00
	0.6	15	0	t.l.	t.l.	171.10	15	0.65	1.79	0.00
	0.8	15	0	t.l.	t.l.	0.00	15	1.1	1.85	0.00
Total		300	225				300			

Table 3
Performance comparison between c-ILP and n-ILP$_{B+TF^*}$ on GRID and REAL-NETWORKS instances.

			n-ILP$_{B+TF^*}$				c-ILP			
				time		nodes		time		nodes
Class	Size\Type	#	# opt	avg.	max	avg.	# opt.	avg.	max	avg.
GRID	10 × 20	6	6	5.14	14.50	16	6	0.05	0.08	0.00
	20 × 160	3	1	426.08	t.l.	30	3	8.87	11.23	0.00
	20 × 40	15	8	275.38	t.l.	0	15	0.33	0.76	0.00
	30 × 60	6	1	283.01	t.l.	15	6	0.99	1.22	0.00
	40 × 80	18	0	390.31	t.l.	63	18	2.22	6.55	0.00
	80 × 40	3	1	402.97	t.l.	0	3	2.94	1.98	0.00
REAL-NETWORKS	Type A	21	13	315.41	t.l.	0	21	1.95	10.62	0.00
	Type B	21	11	401.69	t.l.	0.85	21	3.32	10.34	0.20
Total		93	41				93			

To conclude, Table 1 shows that the most efficient natural formulation is n-ILP$_{B+TF^*}$ solved using a Branch-and-Cut algorithm that separates Benders cuts (8b) at every node of the branching tree and incorporates target-flow inequalities.

4.3. Comparison of the effectiveness of the natural and the compact formulation

In this section, we compare the computational performance of our best Branch-and-Cut algorithm for the natural formulations, i.e., n-ILP$_{B+TF^*}$, against the compact formulation c-ILP solved by CPLEX MIP solver. This study is performed on instances from the SYNTHETIC class, including networks with 40, 50, 60, 100, and 300 vertices, with the same density values as mentioned earlier, i.e., $d(N) \in \{0.2, 0.4, 0.6, 0.8\}$. Additionally, instances from the GRID class and the REAL-NETWORKS class are also examined. We consider three values of the target flow defined by $\lambda \in \{0.2, 0.6, 0.9\}$. Results are reported in Tables 2 and 3. For each formulation, we report the number of instances solved to proven optimality, the average and maximum computing time, and the average number of nodes explored in the branching tree (nodes).

Table 2 directly shows that c-ILP outperforms n-ILP$_{B+TF^*}$. For the smallest networks ($n = 40$), c-ILP solves all instances in record time; less than 0.1 s for all densities. For the same group of instances, n-ILP$_{B+TF^*}$ manages to solve all instances in at most 8.73 s. However, for larger networks, the performance spread between the two formulations is exacerbated. More precisely, we observe that for networks of 100 vertices with a density of 0.4, n-ILP$_{B+TF^*}$ fails to solve four instances to proven optimality. This number increases to 8 instances for a density $d(N)$ equal to 0.8. Subsequently, for networks of 300 vertices and a density larger than 0.2, none of the instances were solved to proven optimality within the time limit. In addition, for the natural formulation n-ILP$_{B+TF^*}$, the number of nodes explored in the branching tree tends to increase as the size of the network grows. In particular, for networks with 50 vertices and a density of 0.2, the maximum number of explored nodes reaches an approximate value of 22 while for networks with 100 vertices and a density of 0.8, this number averages around 400. For larger instances, such as networks of 300 vertices with a density of 0.8, no nodes are explored within the time limit. This observation highlights that the LP relaxation of n-ILP$_{B+TF^*}$ requires a substantial amount of time. Furthermore, based on the results from previous instances, it can be inferred that the time invested in the linear relaxation does not contribute to an overall reduction in the computing time of the Branch-and-Cut algorithm for n-ILP$_{B+TF^*}$. In contrast to n-ILP$_{B+TF^*}$, c-ILP successfully solves all instances with 100 and 300 vertices at the root node, achieving a maximum computing time of 1.85 s. This demonstrates the high efficiency of the compact formulation, which does not require branching to solve the instances in most of the cases. It is worth noticing that the high performance of c-ILP comes essentially from enhancements made by ILP solvers. Recent versions of CPLEX incorporate specific operations executed before the branching phase of the Branch-and-Bound tree, e.g., the preprocessing phase and the cut generation phase. Our experiments have shown that

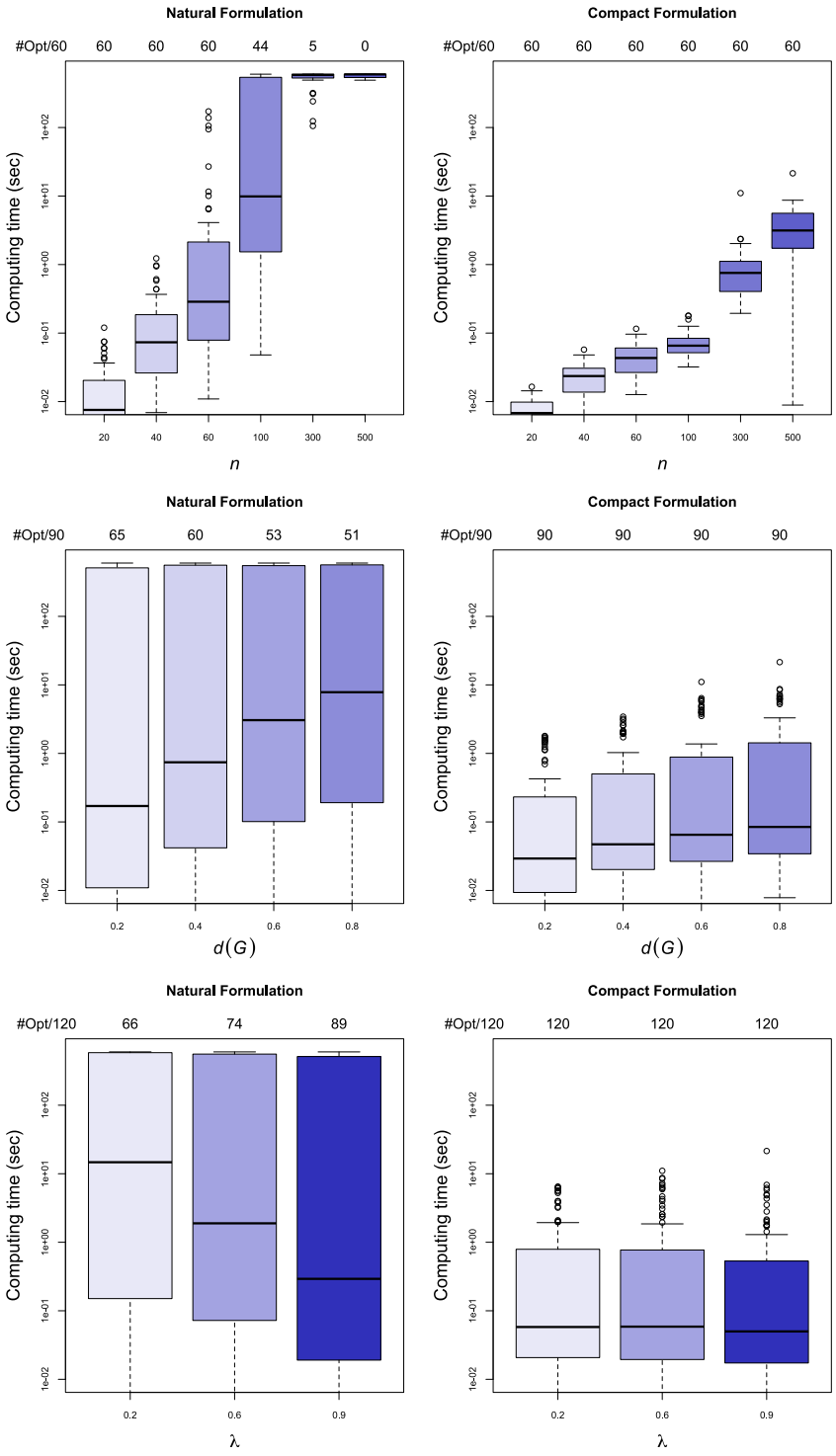

Fig. 5. Computing time boxplots of n-ILP$_{B+TF^*}$ and c-ILP on SYNTHETIC instances.

the aforementioned operations help drastically enhance the efficiency of the solver, especially when dealing with large networks.

Table 3 presents a comparison of c-ILP and n-ILP$_{B+TF^*}$ applied to GRID and REAL-NETWORKS instances. The first six rows correspond to GRID instances grouped by the size, written as $n_1 \times n_2$. We count 17 grid networks with distinct blocker costs and capacities including two grid networks of size 10×20 and 30×60, one grid network of size 20×160 and 80×40, five grid networks of size 20×40 and six grid networks of size 40×80. The last two rows correspond to instances of the REAL-NETWORKS class grouped by the type, i.e., Type A or Type B. As for the SYNTHETIC instances, Table 3 demonstrates that c-ILP outperforms n-ILP$_{B+TF^*}$ by several orders of magnitude. For small GRID instances (10×20, 20×40 and 30×60), c-ILP achieves an optimal solution in less than 2 s, while n-ILP$_{B+TF^*}$ requires an average of approximately 275 and 284 s for grids of size 20×40 and 30×60, respectively. The same performance disparities can be observed for the remaining instances. For Type A and Type B instances from the REAL-NETWORKS class, n-ILP$_{B+TF^*}$ requires an average computing time of 315 s and 401 s, respectively, while c-ILP solves all REAL-NETWORKS instances in approximately 10 s. Moreover, c-ILP manages

to solve all the 93 instances considered for this study, while n-ILP$_{B+TF^*}$ is only able to solve 41 instances to proven optimality (slightly more than 44%). Finally, we observe that on average, c-ILP finds an optimal solution at the root node, while n-ILP$_{B+TF^*}$ tends to explore more nodes. Therefore, as for the SYNTHETIC instances (see Table 2), c-ILP finds an optimal solution of the FB very quickly compared to n-ILP$_{B+TF^*}$ and in most cases at the root node, resulting in a better global computational performance. Table 3 reveals that our best Branch-and-Cut algorithm for the natural formulation n-ILP$_{B+TF^*}$ does not solve all instances from Royset and Wood (2007).

In Fig. 5, we discuss more precisely the impact of network-structural features and flow-blocker features on the performance of c-ILP and n-ILP$_{B+TF^*}$. Fig. 5 gives the computing time boxplots of the two formulations grouping the instances from the SYNTHETIC class by the number of vertices in the network n, the density $d(N)$ and value λ for the target flow. We graphically show the time spent by each formulation through their quartiles. The lines extending vertically from the boxes indicate the variability outside the upper and lower quartiles. Above the upper quartile, the outliers are plotted as individual points. The y-axis is the computing time (in logarithmic scale) and the x-axis represents the group of instances. On the top part of the figure, we report, for each group, the total number of instances solved to proven optimality (#Opt). In the top-left corner, we report #Opt/k, where k is replaced by the total number of instances in each group of tests for a given parameter (n, $d(N)$, and λ). These boxplots provide evidence that c-ILP is the best-performing model to solve the FB, independently of the size of the network and the target flow. However, for both formulations, we observe that the computing time increases as the size of the network grows. Indeed, the number of vertices and the density of the network are directly correlated to the number of variables and constraints of the two models (see (20) for the compact formulation c-ILP and (17) for the natural formulation n-ILP$_{B+TF^*}$). As an illustrative example, for networks of 500 vertices, the natural formulation n-ILP$_{B+TF^*}$ does not reach an optimal solution within the time limit. Regarding network densities, we observe that n-ILP$_{B+TF^*}$ successfully solves 65 instances with a density of 0.2, which decreases to 60, 53 and 51 for greater densities. On the other hand, for networks of up to 500 vertices, c-ILP reaches an optimal solution with a very reasonable computing time, i.e., almost half of the time limit. Finally, Fig. 5 illustrates that n-ILP$_{B+TF^*}$ is more efficient for large values of the target flow. For example for $\lambda = 0.9$, 89 instances out of 120 have been solved to proven optimality within the time limit, while for $\lambda = 0.2$, n-ILP$_{B+TF^*}$ manages to solve only 66 instances. This is consistent with the constraints of the natural formulation n-ILP$_{B+TF^*}$. Indeed, due to the structure of Benders cuts (8b), when increasing the value of the target flow, the left member decreases, which makes the problem easier to solve. In conclusion, Table 2, Table 3 and Fig. 5 demonstrate that c-ILP outperforms n-ILP$_{B+TF^*}$, regardless of the network-structural features or the flow-blocker features.

Since n-ILP$_{B+TF^*}$ (17) and c-ILP (20) share a family of variables, i.e., $\boldsymbol{x}$ variables, which correspond to the blocker decision variables, the target-flow inequalities (12) and the Benders cuts (8) are valid inequalities for the compact formulation. These inequalities can be employed to further enhance the performance of c-ILP. Hence, we performed some experiments on instances from the SYNTHETIC class to evaluate the impact of these constraints in the compact formulation. However, the addition of these cuts does not yield any noticeable computational enhancements, even in larger instances. This can be explained by the optimizations performed by CPLEX ILP solver that offers high computing performance.

A further comparison of the proposed formulations can be found in Appendix, where the quality of the LP relaxation of n-ILP$_B$, n-ILP$_{TF}$, n-ILP$_{B+TF}$, and c-ILP is evaluated to computationally show the results of Proposition 7 and Proposition 8.

4.4. Testing the limits of the compact ILP formulation

In this section, we test the efficiency of c-ILP in solving large-scale networks. More precisely, we are seeking to achieve the limits of the compact formulation (20). By doing so, we can determine the maximum size of instances that can be solved to proven optimality within a fixed amount of CPU time limit of 600 s, similar to the previous experiments. To this end, we include a more extensive set of instances derived from the SYNTHETIC class, including networks with a wide range of values for the number of vertices n. As in our previous experiments, we focus on four specific arc densities, 0.2, 0.4, 0.6, and 0.8, and we consider 9 values of λ, from 0.1 to 0.9. In Fig. 6, we present the computing time boxplot of c-ILP for 10 groups of 180 instances, where each group has the same number of vertices n. As in Fig. 5, we graphically represent the computing time using quartiles and we report the number of instances solved to proven optimality (#opt) for each group out of the total number of instances in the group (180). As expected, we observe a direct correlation between the computing time of c-ILP and the number of vertices in the network. Specifically, for networks with up to 700 vertices, c-ILP reaches an optimal solution in a reasonable time (less than 300 s for the worst case). The compact formulation encounters difficulties when dealing with networks of 1000 vertices, where eight instances remain unsolved. Subsequently, the number of instances solved to proven optimality decreases to 142 out of 180 for $n = 1100$. The most significant disparity is observed for networks of 1200 vertices, with only 34 instances (slightly less than a quarter of the total) achieving proven optimality. Lastly, for networks of 1400 and 1500 vertices, this number decreases to 18 and 1, respectively. Finally, for larger networks (1800 vertices), no instances have been solved to proven optimality within the time limit. It is worth noticing that in cases where instances have not been solved to proven optimality, the solver encounters difficulties in establishing good lower and upper bounds.

5. Conclusion

In this paper, we studied the maximum flow blocker problem. Utilizing a bilevel formulation for the problem, we derived four integer linear programming formulations designed for the first time to address the FB. The first three formulations, involving only the natural variables associated with the arcs and exponential families of constraints, are solved using tailored exact algorithms. The fourth and last formulation is a compact model solved using a state-of-the-art ILP solver. The natural (non-compact) formulations feature two exponential families of constraints. The first one, with separation being polynomial for fractional and integer solutions, is referred to as Benders cuts. The second one, with separation being $\mathcal{NP}$-hard for fractional solutions, is referred to as target-flow inequalities. Using these results, we developed tailored Branch-and-Cut algorithms enhanced with Benders cuts and target-flow inequalities. Additionally, using the compact formulation, we established the first connection between the maximum flow blocker problem and the maximum flow interdiction problem. In particular, we have shown that solutions to one problem can be found using solutions to the other. Furthermore, we conducted a study to compare the strength of the LP relaxations of the proposed ILP formulations. An extensive computational analysis was performed to evaluate the performance of the proposed formulations. The experiments have shown that the models proposed in this paper significantly outperform the direct use of a bilevel solver, representing the current state-of-the-art technique for addressing FB. Moreover, the compact formulation proves to be the most efficient, surpassing the natural formulations and demonstrating its ability to handle large-sized instances efficiently within a reasonable time.

Future works will go in the direction of extending our research to other optimization problems and their variants. An important generalization of the maximum flow problem arises when considering

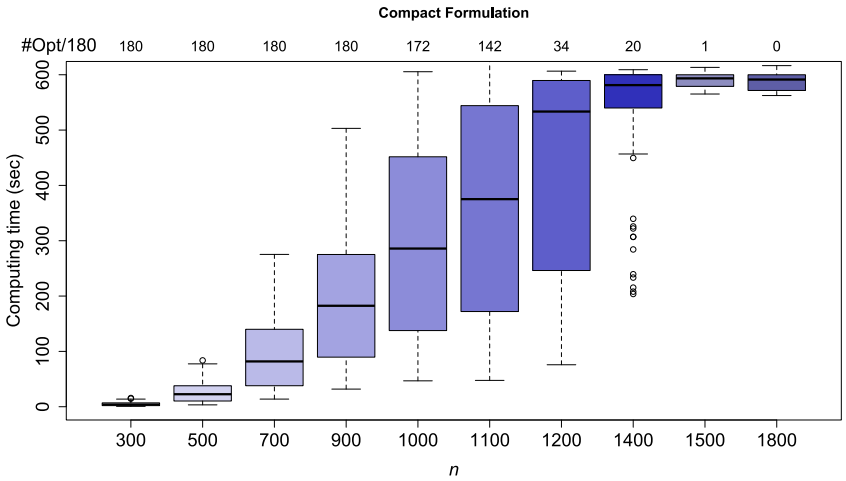

Fig. 6. Computing time boxplot of c-ILP on SYNTHETIC instances.

Table A.4
Performance comparison between LP relaxations of n-ILP$_B$, n-ILP$_{TF}$, n-ILP$_{B+TF}$ and c-ILP on SYNTHETIC instances.

			n-ILP$_B$			n-ILP$_{TF}$				n-ILP$_{B+TF}$					c-ILP	
			#B	lp gap			#TF	lp gap			#B	#TF	lp gap			
n	$d(N)$	#	avg.	avg.	max.	#optLP	avg.	avg.	max.	#optLP	avg.	avg.	avg.	max.	avg.	max.
20	0.2	45	1.0	50.0	90.0	45	21.6	76.2	84.4	45	1.0	51.3	13.3	90.0	41.9	88.9
	0.4	45	7.0	29.1	84.7	45	542.0	53.1	90.0	45	53.2	77.5	4.7	71.6	25.4	80.0
	0.6	45	9.8	20.0	70.1	42	1530.5	63.8	87.2	45	116.4	127.0	6.3	70.1	18.3	58.6
	0.8	45	26.6	22.8	63.9	35	2641.1	59.4	80.4	45	166.9	178.6	6.0	46.3	19.8	61.5
40	0.2	45	22.3	26.7	80.5	41	2072.3	56.8	89.3	45	226.3	249.8	8.8	80.5	23.8	76.2
	0.4	45	75.8	20.6	65.7	13	997.2	25.6	80.7	44	610.4	621.1	4.1	48.7	18.0	64.3
	0.6	45	72.7	10.1	30.0	12	818.2	14.0	90.0	44	355.0	305.5	3.5	27.3	9.1	26.5
	0.8	45	177.5	11.0	42.7	6	684.8	2.8	16.7	38	357.5	353.9	4.8	37.6	9.9	40.0
50	0.2	45	50.3	26.2	80.8	23	1301.5	36.2	84.9	45	520.3	529.0	4.8	80.8	24.4	76.4
	0.4	45	153.5	22.1	70.5	10	1462.6	2.8	14.3	36	593.5	545.1	8.7	70.5	19.6	66.7
	0.6	45	127.6	10.3	57.5	4	608.5	0.0	0.0	37	390.5	392.4	5.9	57.5	9.2	55.0
	0.8	45	355.1	9.2	37.4	2	793.5	0.0	0.0	40	785.7	780.6	3.5	19.5	8.1	37.4
Total		540				278				509						

multiple sources and destinations. This problem, also known as the multi-commodity flow problem, aims to maximize the flow between a set of source–destination pairs. Exploring the blocker variant of the multi-commodity flow problem presents an interesting prospect for further study. Moreover, it would be also interesting to study the adaptation to the FB of the optimality cuts proposed in Laporte and Louveaux (1993) for stochastic integer programs and the ones proposed in Türkoğulları et al. (2016) for an application of Benders Decomposition in seaside operations of container terminals.

CRediT authorship contribution statement

Isma Bentoumi: Investigation, Software, Writing – original draft, Writing – review & editing. **Fabio Furini:** Investigation, Methodology, Writing – original draft, Writing – review & editing, Supervision. **A. Ridha Mahjoub:** Formal analysis, Investigation, Methodology, Supervision. **Sébastien Martin:** Formal analysis, Investigation, Methodology, Project administration, Supervision.

Acknowledgments

The authors are grateful to Ivana Ljubic for illuminating discussions on the subject which also helped us in the presentation of the article. Additionally, we would like to thank the anonymous referees for their valuable comments and suggestions, which significantly improved the quality of this paper. We also wish to acknowledge Huawei Technologies France for their support and collaboration, as this work is part of a PhD thesis conducted in partnership with the company.

Appendix. LP relaxation gaps

In this appendix, we provide a computational comparison of the LP relaxation of n-ILP$_B$, n-ILP$_{TF}$, n-ILP$_{B+TF}$ and c-ILP. The LP relaxation of c-ILP is solved using the LP solver of CPLEX. Solution of the LP relaxation of n-ILP$_B$ and n-ILP$_{B+TF}$ involves separating the Benders cuts by solving the separation problem, i.e., Model (11) (see Section 2.3). In the LP relaxation of n-ILP$_{TF}$ and n-ILP$_{B+TF}$, the target flow inequalities are separated by solving Model (16). In the case of n-ILP$_{B+TF}$, a solution of the LP relaxation is found when all Benders cuts and all target-flow inequalities are separated. To broaden our analysis, we consider 9 values of λ ranging from 0.1 to 0.9 with a step size of 0.1. For each formulation, we report the average and maximum value of *lp gap*, the optimality gap of the LP relaxation. For each instance, lp gap is computed with regards to the optimal solution value *opt* as $100 \times \frac{opt - lp_{val}}{opt}$, where lp_{val} is the value of the linear programming relaxation of the corresponding formulation. For n-ILP$_B$, n-ILP$_{TF}$ and n-ILP$_{B+TF}$, we also report the average and maximum number of cuts generated to solve the LP relaxation (#B for Benders cuts and #TF for target flow inequalities). Finally, for n-ILP$_{TF}$ and n-ILP$_{B+TF}$, we report the number of instances solved for which the lp relaxation solution has been found within the time limit set to 1800 s (#optLP). It is worth noticing that instances for which the LP relaxation has not been found are not considered in the computation of lp gap, #B and #TF. Section 2.6 shows that n-ILP$_B$ and n-ILP$_{TF}$ do not dominate each other. This is highlighted by Table A.4, where the average and maximum lp gap values computed with n-ILP$_B$ can either be smaller or greater than those computed with n-ILP$_{TF}$ for a specific group of instances.

It is worth noticing that the small values of lp gap computed by n-ILP$_{TF}$ for large instances $(2.78, 2.76, 0.00)$ are due to the exclusion of unsolved instances from the computations. Table A.4 shows that the LP relaxation of c-ILP is at least as strong as the LP relaxation of n-ILP$_{B}$, for the instances considered. Specifically, the average value and the maximum value of lp gap for c-ILP are always smaller than the one obtained with n-ILP$_{B}$. When considering networks with 40 vertices and a density of 0.2, the quality of the LP relaxation of c-ILP surpasses that of n-ILP$_{B}$ by approximately 11%. Regarding the maximum computed values of lp gap, this difference becomes more noticeable. Finally, Table A.4 illustrates that c-ILP and n-ILP$_{B+TF}$ are not directly comparable in terms of LP relaxation, as shown in Proposition 8. Indeed, on average, the lp gap of n-ILP$_{B+TF}$ consistently outperforms the average lp gap value computed by c-ILP. However, in specific instances, the maximum lp gap is smaller for c-ILP. For example, when considering networks with 20 vertices and a density of 0.6, the maximum lp gap value for n-ILP$_{B+TF}$ is 70.14, while the corresponding value for c-ILP is 58.57.

References

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*. N.Y: Prentice hall.

Altner, D. S., Ergun, Ö., & Uhan, N. A. (2010). The maximum flow network interdiction problem: Valid inequalities, integrality gaps, and approximability. *Operations Research Letters*, *38*(1), 33–38.

Cormican, K. J., Morton, D. P., & Wood, R. K. (1998). Stochastic network interdiction. *Operations Research*, *46*(2), 184–197.

Dempe, S. (2020). Bilevel optimization: Theory, algorithms, applications and a bibliography. In S. Dempe, & A. Zemkoho (Eds.), *Springer optimization and its applications*, *Bilevel optimization* (pp. 581–672).

Dezső, B., Jüttner, A., & Kovács, P. (2011). LEMON – an open source C++ network template library. *Electronic Notes in Theoretical Computer Science*, *264*(5), 23–45.

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

Fischetti, M., Ljubić, I., Monaci, M., & Sinnl, M. (2018). On the use of intersection cuts for bilevel optimization. *Mathematical Programming*.

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

Furini, F., Ljubić, I., San Segundo, P., & Zhao, Y. (2021). A branch-and-cut algorithm for the edge interdiction clique problem. *European Journal of Operational Research*, *294*(1), 54–69.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco: W.H. Freeman.

Ghafir, I., Prenosil, V., & Hammoudeh, M. (2016). A survey on network security monitoring systems. In *IEEE 4th international conference on future internet of things and cloud workshops* (pp. 77–82). IEEE Xplore Digital Library.

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

Junior, G., Rodrigues, J., Carvalho, L., Al-Muhtadi, J., & Proença, M. (2019). A comprehensive survey on network anomaly detection. *Telecommunication Systems*, *70*(3), 447–489.

Krumke, S. O., Noltemeier, H., Schwarz, S., Wirth, H. C., & Ravi, R. (1998). Flow improvement and network flows with fixed costs. In P. Kall, & H. J. Lüthi (Eds.), *Operations research proceedings* (pp. 158–167). Berlin, Heidelberg: Springer.

Laporte, G., & Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, *13*(3), 133–142.

Lei, X., Shen, S., & Song, Y. (2017). Stochastic maximum flow interdiction problems under heterogeneous risk preferences. *Computers & Operations Research*, *90*, 97–109.

Magnouche, Y., & Martin, S. (2020). Most vital vertices for the shortest $s-t$ path problem: complexity and branch-and-cut algorithm. *Optimization Letters*, *14*, 2039–2053.

McMasters, A. W., & Mustin, T. M. (1970). Optimal interdiction of a supply network. *Naval Research Logistics*, *17*(3), 261–268.

Olver, N., & Végh, L. (2016). A simpler and faster strongly polynomial algorithm for generalized flow maximization. *Journal of the ACM*, *67*(2), 1–26.

Pajouh, F. M., Boginski, V., & Pasiliao, E. (2014). Minimum vertex blocker clique problem. *Networks*, *64*(1), 48–64.

Royset, J., & Wood, R. K. (2007). Solving the bi-objective maximum-flow network-interdiction problem. *INFORMS Journal on Computing*, *19*(2), 175–184.

Schrijver, A. (2002). On the history of the transportation and maximum flow problems. *Mathematical Programming*, *91*(3), 437–445.

Schrijver, A. (2003). *Combinatorial optimization polyhedra and efficiency*. Springer-Verlag Berlin Heidelberg.

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

Tardos, Eva (1985). A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, *5*(3), 247–255.

Türkoğulları, Y., Taşkın, Z., Aras, N., & Altinel, K. (2016). Optimal berth allocation, time-variant quay crane assignment and scheduling with crane setups in container terminals. *European Journal of Operational Research*, *254*(3), 985–1001.

Wei, N., & Walteros, J. (2022). Integer programming methods for solving binary interdiction games. *European Journal of Operational Research*, *302*(2), 456–469.

Wollmer, R. (1964). Removing arcs from a network. *Operations Research*, *12*(6), 934–940.

Wood, R. K. (1993). Deterministic network interdiction. *Mathematical and Computer Modelling*, *17*(2), 1–18.

Wood, R. K. (2011). Bilevel network interdiction models: Formulations and solutions. *Wiley Encyclopedia of Operations Research and Management Science*, *174*, 1–11.

Xu, P. (2012). *Three essays on bilevel optimization algorithms and applications*. (Ph.D. thesis), Iowa State University.