Iranian Journal of Numerical Analysis and Optimization
Vol. 10, No. 1, (2020), pp 1–18
DOI:10.22067/ijnao.v10i1.75392

Research Article

Approximation algorithm for maximum flow network interdiction problem

M. Afsharirad*

Abstract

We consider the maximum flow network interdiction problem. We provide a new interpretation of the problem and define a concept called "optimal cut". We propose a heuristic algorithm to obtain an approximated cut, and we also obtain its error bound. Finally, we show that our heuristic is an α-approximation algorithm for a class of networks. By implementing it on three network types, we show the advantage of it over solving the model by CPLEX.

AMS(2010): 68W25.

Keywords: Interdiction; Approximation algorithm; Network flow; Minimum capacity cut.

1 Introduction

In the maximum flow network interdiction problem (MFNIP), an *interdictor* selects a subset of arcs in a given capacitated network to fully interdict them subject to a given budget limitation in order to minimize the *evader's* maximum flow in the remaining network. Two positive numbers are assigned to all arcs as their capacities and interdiction costs. The problem is known as cardinality maximum flow network interdiction problem (CMFNIP), if all interdiction costs are equal to one.

The original form of MFNIP roots back to more than five decades ago, (see [25, 10]), and has followed by researchers with several variations and application fields. Wood [26] provided the first 0-1 mathematical programming with two types of valid inequalities for MFNIP and showed that the

*Corresponding author
Received 15 September 2018; revised 11 July 2019; accepted 14 July 2019
Maria Afsharirad
Department of Mathematics, University of Science and Technology of Mazandaran, P.O. Box: 48518-78195, Behshahr, Iran. e-mail: m.afsharirad@mazust.ac.ir

problem is NP-hard (Nondeterministic polynimial time), even in cardinality case. He also added two types of valid inequalities to the relaxation of the problem both in general and cardinality cases. Altner, Ergun, and Uhan [3] suggested two types of polynomially separable inequalities for CMFNIP and showed that even if the known IP model of the problem defined in [26] is strengthened, with these valid inequalities, there is still large integrality gap. Two extended formulations are provided for CMFNIP defined in [2], one based on the valid inequalities of [3], and the other based on some new valid inequalities.

In a stochastic interdiction problem, some or all parameters are considered as stochastic variables (see [9, 19, 24]). The maximum reliability path problem on bipartite networks in the stochastic case is the subject of [23]. They provided a convex hull for the polytope of interdictors decision variables and strengthened the relaxation model up to 25% by adding separable inequalities. If there exist information asymmetries between interdictor and evader, then a more computationally difficult version of the problem will arise (see [14, 21]).

Interdicting arcs on dynamic network, in which a positive number is also assigned to all arcs as their traversal time, was studied in [1]. They minimized the dynamic maximum flow in a given period of time. There is also another dynamic version in [13], where interdiction strategies may vary in discrete periods of time.

Network interdiction problems have several applications such as the infection control in hospital (see [4, 17]), identifying critical infrastructure (see [8, 5]), nuclear smuggling (see [16]), vulnerability analysis (see [5]), supply chain networks (see [12]), and border control (see [19, 22]). Minimizing the connectivity of a given network by interdicting a path is another NP-hard interdiction problem, called the critical disruption path; see [11].

There exists only a $(1+\epsilon, 1+\frac{1}{\epsilon})$-pseudo polynomial algorithm for MFNIP; see [6]. For planar network, there is a fully polynomial time approximation scheme (FPATAS) for it; see [20]. Recently Chestnut and Zenklusen [7] provided a $2(n-1)$-approximation algorithm for general graphs with n vertices.

In this paper, we apply the LP (Linaer programming) relaxation of the model defined in [26] and provide a heuristic algorithm for MFNIP. We also calculate the error bound of the method in general networks, while for some special networks with proven large integrality gap, we will show that the proposed algorithm is an α-approximation method.

This paper is organized as follows. In Section 2, we consider the mathematical model for MFNIP and provide a new interpretation for MFNIP. Also we define the concept of "optimal cut". An st-cut is recommended in Section 3 as an approximation for the optimal cut, which leads to a heuristic method. The error bound of the heuristic algorithm is calculated in Section 4. Moreover, we show that the proposed heuristic is an α-approximation algorithm for a class of networks, which overcomes the algorithm defined in [7] for this special class. Section 5 reports the numerical results for our approach

on three types of networks, and we compare the performance with the $0-1$ model in literature. Finally, Section 6 concludes the paper.

2 Interpreting MFNIP

Let $G = (N, A)$ be a digraph with set of nodes N and set of arcs A. Let $\mathcal{N} = (G, \mathbf{u}, \mathbf{r}, s, t)$ be a given network, in which $\mathbf{u}$ and $\mathbf{r}$ are the vectors of arc capacities and interdiction costs, respectively, and $\{s, t\}$ is the set of terminal nodes. Our goal in MFNIP is to find the subset of arcs with interdiction costs lower than or equal to a given limited budget R, whose removal leads to the least maximum st-flow in the remaining network.

The first mathematical model for MFNIP was proposed in [26], with the following decision variables:

$$\pi_i = \begin{cases} 1 & \text{if node } i \text{ is in the sink side of the cut,} \\ 0 & \text{otherwise,} \end{cases}$$

$$\theta_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is the forward arc of the cut,} \\ 0 & \text{otherwise,} \end{cases}$$

$$\beta_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is the forward arc of the cut and not interdicted,} \\ 0 & \text{otherwise,} \end{cases}$$

$$\gamma_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is interdicted,} \\ 0 & \text{otherwise.} \end{cases}$$

Consider the following integer programming from [26]:

$$\min \quad \sum_{(i,j)\in A} u_{ij}\beta_{ij}, \tag{1}$$

$$s.t. \quad \pi_i - \pi_j + \theta_{ij} \geq 0, \quad (i,j) \in A, \tag{1a}$$

$$\pi_t - \pi_s \geq 1, \tag{1b}$$

$$\sum_{(i,j)\in A} r_{ij}\gamma_{ij} \leq R, \tag{1c}$$

$$\gamma_{ij} + \beta_{ij} - \theta_{ij} \geq 0, \quad (i,j) \in A, \tag{1d}$$

$$\pi_i \in \{0,1\}, \quad i \in N, \tag{1e}$$

$$\theta_{ij} \in \{0,1\}, \quad (i,j) \in A, \tag{1f}$$

$$\gamma_{ij} \in \{0,1\}, \quad (i,j) \in A, \tag{1g}$$

$$\beta_{ij} \in \{0,1\}, \quad (i,j) \in A. \tag{1h}$$

In optimality, inequality (1d) changes to the equality (see [26]). Hence, letting $\theta_{ij} = \beta_{ij} + \gamma_{ij}$ leads to the main integer programming defined in [26]. Here we substitute β_{ij} by $\theta_{ij} - \gamma_{ij}$ to obtain

$$\min \quad \sum_{(i,j)\in A} u_{ij}(\theta_{ij} - \gamma_{ij}) \tag{2}$$

$$\begin{aligned} s.t. \quad & \gamma_{ij} \leq \theta_{ij},\ (i,j) \in A, && \text{(2a)}\\ & \text{(1a)} - \text{(1c)},\\ & \pi_i \in \{0,1\},\ i \in N,\\ & \gamma_{ij}, \theta_{ij} \in \{0,1\},\ (i,j) \in A. \end{aligned}$$

Note that $\beta_{ij} \in \{0,1\}$, for all $(i,j) \in A$, so inequalities (2a) are imposed to model (2), since no arc out of the cut will be interdicted in optimal case. The objective function (2) can be rewritten as follows:

$$\min_{\pi,\theta} \sum_{(i,j)\in A} u_{ij}\theta_{ij} - \max_{\gamma} \sum_{(i,j)\in A} u_{ij}\gamma_{ij},$$

which means we are solving the minimum capacity cut problem with constraints (1a), (1b), (1e), and (1f) and the knapsack problem with constraints (1c) and (1g), and constraints (2a) relate these two problems.

Remark 1. Solving the $0-1$ knapsack problem on a cut, means to suppose forward arcs of the cut as items, arc capacity as the item's value, and arc interdiction cost as the item's weight. The limited budget R is also the limited capacity of the knapsack. Interdicting arc (i,j) corresponds to select item (i,j) for knapsack.

Remark 2. The MFNIP is to solve the $0-1$ knapsack problem on all feasible st-cuts of the problem and choose the cut, in which the remaining capacity is minimum after interdiction.

Therefore we define the following concept.

Definition 1. For a given network $\mathcal{N} = (G, \mathbf{u}, \mathbf{r}, s, t)$ and budget R, an st-cut $\mathcal{C}$ is called *optimal cut*, if solving $0-1$ knapsack problem on it (based on Remark 1), results in minimum remaining capacity on it. The optimal cut is shown by $\mathcal{C}^*$.

Now it is obvious that MFNIP is to find the optimal cut according to Definition 1, which is not necessarily the minimum capacity cut.

3 Approximated cut

In order to approximate the optimal cut, consider the dual of the LP relaxation of model (1) as follows:

$$\max \quad f - Rw \tag{3}$$

$$s.t. \quad \sum_{j:(s,j)\in A} f_{sj} - \sum_{j:(j,s)\in A} f_{js} = f, \tag{3a}$$

$$\sum_{j:(i,j)\in A} f_{ij} - \sum_{j:(j,i)\in A} f_{ji} = 0,\ i \in N, \tag{3b}$$

$$\sum_{j:(t,j)\in A} f_{tj} - \sum_{j:(j,t)\in A} f_{jt} = -f, \tag{3c}$$

$$\lambda_{ij} \le u_{ij},\ (i,j) \in A, \tag{3d}$$

$$\lambda_{ij} - wr_{ij} \le 0,\ (i,j) \in A, \tag{3e}$$

$$f_{ij} - \lambda_{ij} \le 0,\ (i,j) \in A, \tag{3f}$$

$$\lambda_{ij}, f_{ij} \ge 0,\ (i,j) \in A, \tag{3g}$$

$$f, w \ge 0. \tag{3h}$$

The variables f_{ij} and f are the dual variables of constraints (1a) and (1b), respectively, and the variables λ_{ij} and w are also the dual variables of constraints (1d) and (1c), respectively.

For a fixed w, model (3) is a maximum flow problem reduced by a factor Rw, in which the capacity of arc (i,j) is

$$\lambda_{ij} = \min\{u_{ij}, wr_{ij}\}. \tag{4}$$

Let w^* be the optimal value of w in model (3). Then for $w = w^*$, the maximum flow value is equal to $f^* - Rw^*$ and is obtained by capacities $\lambda_{ij} = \min\{u_{ij}, w^* r_{ij}\}$. The dual problem of maximum flow problem (3) is the following minimum capacity cut problem:

$$U_{\mathcal{C}_{w^*}} = \min \quad \sum_{(i,j)\in A} \lambda_{ij}\bar{\theta}_{ij}, \tag{5}$$

$$\bar{\pi}_i - \bar{\pi}_j + \bar{\theta}_{ij} \ge 0,\ (i,j) \in A \tag{5a}$$

$$\bar{\pi}_t - \bar{\pi}_s \ge 1, \tag{5b}$$

$$\bar{\theta}_{ij} \ge 0,\ (i,j) \in A. \tag{5c}$$

Arc capacities in model (5) are calculated by (4) with $w = w^*$. Also $\mathcal{C}_{w^*}$ is the cut obtained by model (5) and $U_{\mathcal{C}_{w^*}}$ is its capacity.

Remark 3. Since w is the dual variable of the LP relaxation problem, the equality $\mathcal{C}_{w^*} = \mathcal{C}^*$ does not necessarily hold.

3.1 Parametric procedure

In this section, we provide a parametric procedure to calculate w^*. For a fixed w, let $f(w)$ be the optimal value of the maximum flow problem in (3), according to arc capacities in (4). It is clear that $f(0) = 0$ and that increasing w increases $f(w)$ up to $w = w_{\max} = \max_{(i,j)\in A} \frac{u_{ij}}{r_{ij}}$, so

$$\max_{w} f(w) = f(w_{\max}) = f_{\max}.$$

Let $\hat{E}(w)$ be the set of arcs for which their capacities are decreased by the parameter w:

$$\hat{E}(w) = \{(i,j) \in A | \lambda_{ij} = w r_{ij}\}.$$

In this paper, $\hat{E}(w)$ is called *set of active arcs*.

If $U_{\mathcal{C}}$ shows the capacity of $s-t$ cut $\mathcal{C}$, $f(w)$ is the maximum flow, and $Z(w)$ is the optimal objective value of model (3) by arc capacities obtained by (4), then by the maximum flow-minimum cut theorem, we have

$$f(w) = U_{\mathcal{C}_w} = \sum_{(i,j)\in \mathcal{C}_w \setminus \hat{E}(w)} u_{ij} + w \sum_{(i,j)\in \mathcal{C}_w \cap \hat{E}(w)} r_{ij}.$$

It is not difficult to verify that $f(w)$ is a concave piecewise linear function, since increasing w leads to decrease number of arcs in $\hat{E}(w)$ and consequently in $\mathcal{C}_w \cap \hat{E}(w)$, so increasing w decreases the slope of linear pieces. Let

$$Z(w) = f(w) - Rw = \sum_{(i,j)\in \mathcal{C}_w \setminus \hat{E}(w)} u_{ij} + w \Big(\sum_{(i,j)\in \mathcal{C}_w \cap \hat{E}(w)} r_{ij} - R \Big). \tag{6}$$

Then for $w \in [0, \frac{f_{\max}}{R}]$, $Z(w)$ is a concave piecewise linear function as shown in Figure 1. The slope of any linear piece of $Z(w)$ is

$$s(w) = \sum_{(i,j)\in \mathcal{C}_w \cap \hat{E}(w)} r_{ij} - R. \tag{7}$$

Note that similar to $f(w)$, increasing w limits the set $\hat{E}(w)$ and the slope $s(w)$ decreases, so $Z(w)$ is a concave function.

Now in order to calculate $Z(w)$, it suffices to start with any w_l and w_r with $0 < w_l < w_r$ and obtain the cross point of the line passing through point $(w_l, Z(w_l))$ with slope $s(w_l)$ and the line passing through point $(w_r, Z(w_r))$

with slope $s(w_r)$. Call this point $(\bar{w}, \bar{Z})$. Whenever $\bar{Z} = Z(\bar{w})$, we stop, otherwise we update w_l and w_r. Algorithm 1 formally explains the procedure.

Algorithm 1 Parametric procedure

Input: $G = (N, A)$, $\mathcal{N} = (G, \mathbf{u}, \mathbf{r}, s, t)$; Total budget: R.
Output: Optimal value w^*, for problem (3).

1: Let $w_\ell = 0$ and $w_r = \max_{(i,j)\in A} \frac{u_{ij}}{r_{ij}}$;
2: For $w = w_l$ and $w = w_r$, calculate $Z(w)$ and $s(w)$ from (6) and (7), respectively;
3: Let $\bar{w} = \frac{Z(w_r) - Z(w_\ell) + w_\ell\, s(w_\ell) - w_r\, s(w_r)}{s(w_\ell) - s(w_r)}$ and $\bar{Z} = Z(w_\ell) + (\bar{w} - w_\ell)s(w_\ell)$;
4: Calculate $Z(\bar{w})$ and $s(\bar{w})$;
5: **if** $Z(\bar{w}) = \bar{Z}$ **then**
6: $w^* = \bar{w}$;
7: **if** $s(w^*) > 0$ **then**
8: Let $(w_\ell^*, w_r^*) = (w^*, w_r)$;
9: **else**
10: Let $(w_\ell^*, w_r^*) = (w_\ell, w^*)$;
11: **end if**
12: Stop;
13: **else**
14: **if** $s(\bar{w}) > 0$ **then**
15: Let $w_l = \bar{w}$, $Z(w_\ell) = Z(\bar{w})$ and $s(w_\ell) = s(\bar{w})$;
16: **end if**
17: **if** $s(\bar{w}) < 0$ **then**
18: Let $w_r = \bar{w}$, $Z(w_r) = Z(\bar{w})$ and $s(w_r) = s(\bar{w})$;
19: **end if**
20: **end if**
21: Go to step 3

In step (1), lower and upper bounds of w are determined. In step (2), values of $Z(w)$ and $s(w)$ are obtained. In step (3), the crosspoint of two lines passing from $(w_\ell, Z(w_\ell))$ and $(w_r, Z(w_r))$ with slopes $s(w_\ell)$ and $s(w_r)$, respectively, is achieved. In step (4), values of $Z(\bar{w})$ and $s(\bar{w})$ are obtained by using equations (6) and (7), respectively. In step (5), we check the stop condition, if it is not satisfied, and we update values of w_ℓ and w_r, in step (13) and repeat the procedure.

Note that calculating $Z(w)$ and $s(w)$ for a given w in steps (2) and (4), means to solve a maximum flow problem for which there exist several algorithms, for example, the algorithm of [18], which has a running time of $o(|N||A|)$. Therefore, Algorithm 1 is running in polynomial time.

Table 1: Notations

Z_{IP}	Optimal objective value of model (1).
Z_{RIP}	Optimal objective value of model (3).
Z_H	Objective value obtained by applying Algorithm 2.
$f(w)$	Maximum flow with capacities (4).
$Z(w)$	$= f(w) - RW$.
E_0	$= \{(i,j) \in \mathcal{C}_{w^*} \mid u_{ij} \leq w^* r_{ij}\}$.
E_1	$= \{(i,j) \in \mathcal{C}_{w^*} \mid u_{ij} > w^* r_{ij}\}$.
I_{w^*}	$= \{(i,j) \in \mathcal{C}_{w^*} \mid (i,j)$ is interdicted by Algorithm 2, (or is selected in knapsack problem)$\}$.
I_0	$= \{(i,j) \in I_{w^*} \mid u_{ij} \leq w^* r_{ij}\}$.
I_1	$= \{(i,j) \in I_{w^*} \mid u_{ij} > w^* r_{ij}\}$.

3.2 Heuristic method

Now we suggest a heuristic method, and in section 4, we will show this heuristic turns out to be an α-approximation method for a class of networks.
The proposed heuristic is based on the cut $\mathcal{C}_{w^*}$. In this method, we consider cut $\mathcal{C}_{w^*}$ as an approximation for the optimal cut $\mathcal{C}^*$, and we call it an *approximated cut*, then we solve knapsack problem on it.

Algorithm 2 The heuristic algorithm

Input: $\mathcal{N} = (G, \mathbf{u}, \mathbf{r}, s, t)$, $G = (N, A)$; Total budget: R.
Output: Subset of arcs of cost lower than or equal to R
1: Calculate w^* by algorithm 1;
2: Obtain st-cut $\mathcal{C}_{w^*}$, from model (5);
3: Solve $0-1$ knapsack problem on $\mathcal{C}_{w^*}$, according to Remark 2.

It is clear that Algorithm 2 is not polynomial because of step (3), however one can apply some known heuristics, for example, greedy algorithm to approximately solve the respective knapsack problem.

4 Error term

In this section, we calculate the error term of Algorithm 2. Table 1 shows notations, applied in this section.

First, note that $f(w)$ is a linear function in of w and $Z(w)$ is a concave piece-wise linear function of w, starting from original and meets the vertical axis in $w = \frac{f_{\max}}{R}$, where $f_{\max}$ is the maximum value of $f(w)$ among all values

of w. For a fixed w, the slow of any piece of $Z(w)$ is calculated as follows:

$$s(w) = \sum_{(i,j)\in C^* \cap \{(i,j)|wr_{ij}<u_{ij}\}} r_{ij} - R.$$

Figure 1 shows the function $Z(w)$, where w_ℓ and w_r are two arbitrary points lower and greater than w^*, respectively.

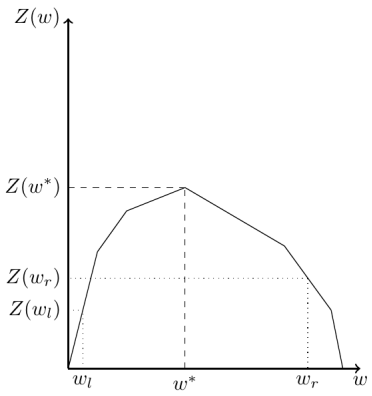

Figure 1: Diagram of $Z(w)$

According to Table 1, we have $\mathcal{C}_{w^*} = E_0 \cup E_1$ and $I_{w^*} = I_0 \cup I_1$. It is clear that

$$Z_{RIP} = f^* - w^*R = U_{\mathcal{C}_{w^*}} - w^*R = \sum_{(i,j)\in E_0} u_{ij} + w^*(\sum_{(i,j)\in E_1} r_{ij} - R). \quad (8)$$

Also

$$\begin{aligned} Z_H &= \sum_{(i,j)\in\mathcal{C}_{w^*}} u_{ij} - \sum_{(i,j)\in I_{w^*}} u_{ij} \\ &\leq \sum_{(i,j)\in\mathcal{C}_{w^*}} u_{ij} - \sum_{(i,j)\in I_1} u_{ij} \quad (9) \\ &\leq \sum_{(i,j)\in\mathcal{C}_{w^*}} u_{ij} - w^* \sum_{(i,j)\in I_1} r_{ij}. \quad (10) \end{aligned}$$

Now it is easy to conclude

$$
\begin{aligned}
&Z_{RIP} \le Z_{IP} \le Z_H \\
&\Rightarrow Z_H - Z_{IP} \le Z_H - Z_{RIP} \\
&\qquad \le \sum_{(i,j)\in \mathcal{C}_{w^*}} u_{ij} - w^* \sum_{(i,j)\in I_1} r_{ij} - \sum_{i\in E_0} u_{ij} - w^* \sum_{(i,j)\in E_1} r_{ij} + w^* R \\
&\qquad = \sum_{(i,j)\in E_1} u_{ij} - w^* \sum_{(i,j)\in I_1} r_{ij} + w^*(R - \sum_{(i,j)\in E_1} r_{ij}). \qquad (11)
\end{aligned}
$$

Therefore the relative error is

$$
\begin{aligned}
Err &= \frac{Z_H - Z_{IP}}{Z_{IP}} \le \frac{Z_H - Z_{IP}}{Z_{RIP}} \qquad (12) \\
&\le \frac{\sum_{(i,j)\in E_1} u_{ij} - w^* \sum_{(i,j)\in I_1} r_{ij} + w^*(R - \sum_{(i,j)\in E_1} r_{ij})}{\sum_{(i,j)\in E_0} u_{ij} + w^*(\sum_{(i,j)\in E_1} r_{ij} - R)} \\
&\le \begin{cases} \frac{\sum_{(i,j)\in E_1} u_{ij} - w^* \sum_{(i,j)\in I_1} r_{ij}}{\sum_{(i,j)\in E_0} u_{ij}} & if s(w^*) > 0, \\ \frac{\sum_{(i,j)\in E_1} u_{ij} - w^* \sum_{(i,j)\in I_1} r_{ij} + w^*(R - \sum_{(i,j)\in E_1} r_{ij})}{\sum_{(i,j)\in E_0} u_{ij} + w^*(\sum_{(i,j)\in E_1} r_{ij} - R)} & if s(w^*) < 0. \end{cases} \qquad (13)
\end{aligned}
$$

4.1 Approximation scheme for $I_{\kappa,\mu}$ networks

The $I_{\kappa,\mu}$ class of networks (see [3]) is defined as follows:

For two positive integers μ and κ, with $\kappa \ge 2$ and $\mu >> \kappa$, $I_{\kappa,\mu} = (N, A)$, where $N = \{s, t\} \cup X \cup Y \cup Z$, $A = A^s \cup A^t \cup A^d \cup A^b$, with $|X| = \kappa$, $|Y| = |Z| = \mu$, and

$$
\begin{array}{llll}
A^s = \{(s,i),\ i \in X\}, & u_{ij} = \mu, & (i,j) \in A^s, \\
A^t = \{(i,t),\ i \in X \cup Y\}, & u_{ij} = 1, & (i,j) \in A^t, \\
A^b = \{(i,j),\ i \in X, j \in Y\}, & u_{ij} = \mu^2, & (i,j) \in A^b, \\
A^d = \{(s,i),(i,t),\ i \in Z\}, & u_{ij} = \mu^2, & (i,j) \in A^d.
\end{array}
$$

All interdiction costs are equal to one and $R = \mu + \kappa - 1$. Altner, Ergun, and Uhan [3] improved valid inequalities of [26] and showed that adding their valid inequalities does not improve the value of the objective function. They also proved that for $I_{\kappa,\mu}$ networks, the integrality gap of model (1) is not bounded by a constant. According to Definition 1, $\{(s,i),\ i \in X \cup Z\}$ is the optimal cut (see [3]). Solving the knapsack problem on the optimal cut results in $Z_{IP} = \mu$ amount of remaining capacity.

In this section, we show that Algorithm 2 finds a bound for integrality gap of $I_{\kappa,\mu}$ networks.

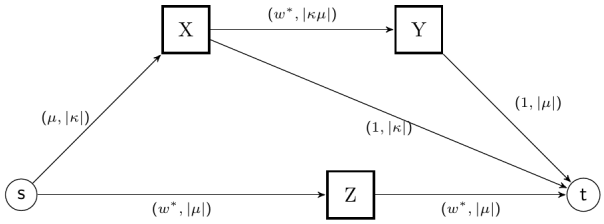

Figure 2: Network instance of $I_{\kappa,\mu}$, with capacities $\lambda_{ij} = \min\{u_{ij}, w^*\}$ and $\mu < w^* \leq \mu^2$

Lemma 1. Let w^* be optimal value for model (3) for an instance of $I_{\kappa,\mu}$; then $w^* \leq \mu$.

Proof. Note that

$$w^* \leq w_{\max} = \max_{(i,j)\in A} \frac{u_{ij}}{r_{ij}} = \mu^2.$$

Now by contradiction suppose that $\mu < w^* \leq \mu^2$; then we have the network in Figure 2. All arcs in Figure 2 are labeled with two numbers $(a, |b|)$, where a is the arc capacity according to (4), and b is the number of such arcs in the network. It is clear from Figure 2 that $\{(i,t),\ i \in X \cup Y \cup Z\}$ is the minimum capacity cut with capacity $\kappa + \mu + w^*\mu$. Consequently the optimal value of the objective function of model (3) is equal to

$$Z(w^*) = \mu + \kappa + w^*\mu - Rw^*.$$

Comparing two cuts with arc capacities in (4), one with $w = w^*$ and one with $w = \mu$, leads to

$$\begin{aligned} Z(\mu) - Z(w^*) &= \mu^2 + \mu + \kappa - R\mu - \mu - \kappa - w^*\mu + Rw^* \\ &= \mu^2 - w^*\mu + R(w^* - \mu) \\ &= \mu^2 - w^*\mu + (\mu + \kappa - 1)(w^* - \mu) \\ &= \kappa w^* - w^* - \mu\kappa + \mu = (w^* - \mu)(\kappa - 1) > 0 \\ &\Longrightarrow Z(w^*) < Z(\mu), \end{aligned}$$

which contradicts with optimality of w^*. Therefore $w^* \leq \mu$. □

Theorem 1. Algorithm 2 on an instance of $I_{\kappa,\mu}$ is $\frac{1}{\mu}$-approximation algorithm.

Proof. Consider an instance $I_{\kappa,\mu}$ and assume that w^* has been obtained by model (3). Now we have to solve the minimum capacity cut problem on

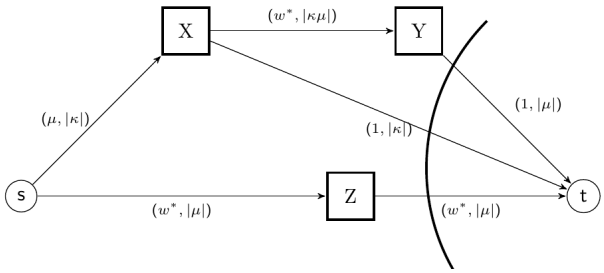

Figure 3: Network instance of $I_{\kappa,\mu}$, with capacities $\lambda_{ij} = \min\{u_{ij}, w^*\}$

the network with arc capacities $\lambda_{ij} = \min\{u_{ij}, w^*\}$ for all $(i, j) \in A$. This network is shown in Figure 3.

Note that in arc capacities of Figure 3, we have applied Lemma 1, that is, $\min\{\mu, w^*\} = w^*$. First we have to find the minimum capacity cut. Among all st-cuts in Figure 3, there are two st-cuts with the minimum capacity depending on the value of w^*: The capacity of dotted cut is equal to $\kappa w^* + w^*\mu$, and the capacity of the solid cut is equal to $\mu + \kappa + w^*\mu$. Therefore we have the following two cases:

Case 1: If $\kappa(w^* - 1) < \mu$, then the dotted cut is the min cut obtained by model 5, since we mentioned that this cut is the optimal cut and solving knapsack on it will give us the optimal value equal to μ.

Case 2: If $\kappa(w^* - 1) > \mu$, then the solid cut is the result and knapsack problem should be solved on it. We have to remove $\kappa + \mu - 1$ arcs to minimize the remaining capacity. By removing all arcs from set Z to node t and all arcs but one from set X to t, the remaining capacity is equal to $\mu+1$.

Note that if $\kappa(w^* - 1) = \mu$ either first case or second case may happen. Thus

$$\frac{Z_H - Z_{IP}}{Z_{IP}} \leq \frac{\mu + 1 - \mu}{\mu} = \frac{1}{\mu}.$$

□

5 Numerical results

In this section, we solve model (1) for three types of networks and also apply Algorithm 2 on them to investigate the efficiency of the proposed algorithm. We carry out all computational tests on an Intel(R) Core(TM)2 Duo Proces-

Table 2: Results for $I_{\kappa,\mu}$ networks

Network parameters			Optimal value		CPU time (min:sec)	
μ	κ	$(\|N\|,\|A\|)$	Z_{RIP}	Z_H	model (1)	Algorithm 2
10	2	(23,54)	6.00	11	00:00.509	00:00.692
20	5	(46,170)	5.00	20	00:00.405	00:00.666
40	5	(86,330)	9.00	40	00:00.494	00:00.739
50	5	(106,410)	11.00	51	00:00.415	00:00.818
100	10	(211,1320)	11.00	100	00:00.991	00:01.241
150	20	(321,3490)	8.50	150	00:04.707	00:01.541
150	50	(360,8050)	4.00	150	00:14.781	00:02.303
200	50	(451,10700)	5.00	200	00:24.617	00:03.083
200	70	(471,14740)	3.86	200	00:45.182	00:04.316
200	100	(501,20800)	3.00	200	01:25.428	00:06.858
500	100	(1101,51700)	6.00	500	12:59.496	00:21.992

sor 2.20 GHz, 4 GB of RAM, GAMS 24.2.2 generates models, and CPLEX 12.6.0.0. solves them.

5.1 $I_{\kappa,\mu}$ networks

$I_{\kappa,\mu}$ networks were introduced in section 4.1. Table 2 shows results for this class of networks. Eleven $I_{\kappa,\mu}$ networks with different parameter values were tested. The column "*Optimal value*" shows the value of objective function. Also Z_{RIP} is the optimal value of relaxation of model (1), which shows the large integrality gap, while Z_H is the value obtained by Algorithm 2. Note that there is no column for optimal value Z_{IP}, since we know that $Z_{IP} = \mu$, so this value is not reported in this column.

It is clear in Table 2 that specially for large amount of κ and μ, Algorithm 2 is extremely faster than solving the original model (1) by CPLEX. However results in section 4.1 also confirms the advantage of Algorithm 2.

5.2 $G_{h,g}$ networks

In this section, we test the proposed algorithm on some layered networks called $G_{h,g}$, with g columns of nodes and h nodes in each column. Nodes of the first column are considered as source nodes and nodes of the last are sink nodes. Any node in the ith column is incident to all nodes in the $i+1$th

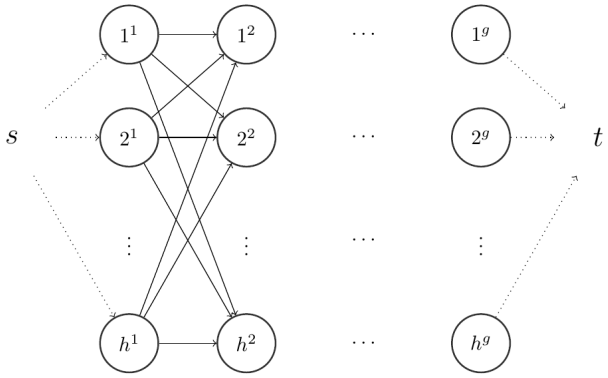

Figure 4: Network instance of $G_{h,g}$

column, for $i = 1, \ldots, g-1$. Arc capacities and interdiction costs are random integer numbers, uniformly distributed in $[10, 30]$. Figure 4 shows a schematic of $G_{h,g}$ networks.

Table 3 shows the results obtained by solving model (1) and applying Algorithm 2 on $G_{h,g}$ networks. Despite $I_{\kappa,\mu}$ networks, Algorithm 2 does not perform well on $G_{h,g}$ networks in compare to model (1). Moreover there are several cases in Table 3, where Algorithm 2 does not find the optimal value.

5.3 Star mesh networks

This section describes tests on star mesh networks; see [15], for a detailed description of this network type. Figure 5 shows a star mesh network. Also $S_{\rho,\nu}$ is a star network with ρ rays and ν rings. The node in the center of the star is the source node s, and all nodes in the outer ray are sink nodes. Arc capacities and interdiction costs are drawn form a discrete uniform distribution on $[1, 10]$ and $[10, 30]$, respectively.

Table 3: Results for $G_{h,g}$ networks

Network parameters			Optimal value			CPU time (min:sec)	
(h,g)	$(\|V\|,\|A\|)$	R	Z_{RIP}	Z_{IP}	Z_H	model (1)	Algorithm 2
(5,5)	(27,110)	43	99	103	105	00:01.974	00:01.741
(5,10)	(52,235)	21	207.2	209	209	00:00.550	00:01.721
(6,10)	(62,336)	235	230.23	242	253	00:01.250	00:01.140
(8,10)	(81,592)	235	670	673	680	00:03.160	00:01.156
(7,9)	(65,406)	75	439.8	448	469	00:01.043	00:01.085
(9,9)	(83,666)	82	455.105	461	482	00:01.121	00:01.083
(9,15)	(137,1152)	251	228.28	239	239	00:02.930	00:01.488
(10,8)	(82,720)	257	571.14	574	577	00:00.701	00:01.103
(10,15)	(152,1420)	300	290.60	293	293	00:01.144	00:01.582
(25,30)	(751,8175)	700	5750.75	5751	5760	01:00.500	02:15.684
(25,30)	(751,8175)	2000	1581.25	1582	1585	03:02.430	03:36.746

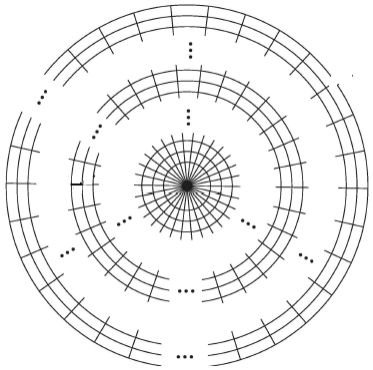

Figure 5: Star mesh network instance of $S_{\rho,\nu}$

Table 4 provides detailed results on star mesh networks. For fairly small networks, model (1) is much more faster than Algorithm 2, however as network parameters increase, Algorithm 2 exceeds model (1).

6 Conclusion

In this paper, we applied the $0-1$ mathematical model for MFNIP in literature and provided an interpretation and consequently a heuristic algorithm for it. We also showed that the provided algorithm is a $\frac{1}{\mu}$-approximation al-

Table 4: Results for star mesh networks

Network parameters			Optimal value			CPU time (min:sec)	
(ρ,ν)	$(\|V\|,\|A\|)$	R	Z_{RIP}	Z_{IP}	Z_H	model (1)	Algorithm 2
(5,50)	(251,505)	30	3.42	4	4	00:00.609	00:00.942
(10,10)	(101,210)	25	15.42	17	18	00:00.339	00:00.932
(15,10)	(151,315)	30	29.3	31	32	00:00.397	00:00.840
(20,15)	(301,620)	71	34.27	35	35	00:00.490	00:00.990
(20,20)	(401,820)	100	29.66	31	31	00:01.176	00:01.033
(25,20)	(501,1025)	130	30.96	32	33	00:00.881	00:01.227
(30,30)	(901,1830)	200	29	30	31	00:02.882	00:01.947
(40,30)	(1201,2440)	200	49.3	50	50	00:04.719	00:03.209
(40,50)	(2001,4040)	300	33.54	34	36	00:20.667	00:09.278
(70,50)	(3501,7070)	500	54.85	55	55	01:36.782	00:36.621
(70,80)	(5601,11270)	1000	6.26	7	7	06:30.792	02:14.322

gorithm for $I_{\kappa,\mu}$ class of networks. However the error of the proposed method was also calculated for general networks.

Numerical results showed that the proposed algorithm is generally faster than the $0-1$ model by CPLEX, specially on large size grids and star mesh networks. Although the proposed algorithm is not polynomial due to the knapsack problem in it, but it is still faster than $0-1$ model on large networks. The reason is that according to the provided interpretation of the problem, solving MFNIP is equivalent to solving knapsack problem as many as st-cuts in the network, while our proposed heuristic algorithm reduced it to solving knapsack problem just once. This fact indeed, explains its preponderant performance.

Acknowledgements

I would like to thank Paolo Serafini (University of Udine) for his useful insights provided in the early stages of this paper, and also I am grateful to there anonymous referees and editor for their constructive comments.

References

1. Afsharirad, M. and Taghizadeh, H. *Maximum dynamic network flow interdiction problem: New formulation and solution procedures*, Comput. Ind. Eng. 65(4) (2013), 531–536.

2. Afsharirad, M. and Taghizadeh, H. *Two extended formulations for cardinality maximum flow network interdiction problem*, Networks, 69(4) (2017), 367–377.

3. Altner, D.S., Ergun, O. and Uhan, N.A., *The maximum flow network interdiction problem: valid inequalities, integrality gaps and approximability,* Oper. Res. Lett., 38 (2010), 33–38.

4. Assimakopoulos, N. *A network interdiction model for hospital infection control,* Computers in Biology and Medicine, 17 (1987), 413–422.

5. Brown, G., Carlyle, M., Salmeron, J. and Wood, R.K. *Defending critical infrastructure,* Interfaces, 36(6) (2005), 530–544.

6. Burch, C., Carr, R., Krumke, S., Marathe, M., Phillips, C. and Sundberg, E. *A decomposition-based pseudoapproximation algorithm for network flow inhibition*, Network interdiction and stochastic integer programming (Davis, CA, 2002), 51–68, Oper. Res./Comput. Sci. Interfaces Ser., 22, Kluwer Acad. Publ., Boston, MA, 2003.

7. Chestnut, S.R. and Zenklusen, R. *Hardness and approximation for network flow interdiction,* Networks, 69(4) (2017), 378-387.

8. Church, R.L., Scaparra, M.P. and Middleton, R.S. *Identifying critical infrastructure: The median and covering facility interdiction problems,* Annals of the Association of American Geographers, 94 (2004), 491–502.

9. Cormican, K.J., Morton, D.P., and Wood, R.K. , *Stochastic network interdiction*, Oper. Res., 46 (1998), 184–197.

10. Durbin, E.P. *An interdiction model of highway transportation,* Santa Monica, CA: RAND Corporation, 1966.

11. Granata, D., Steeger, G. and Rebennack, S. *Network interdiction via a critical disruption path: Branch-and-price algorithms,* Comput. Oper. Res., 40(11) (2013), 2689–2702.

12. Lim, C. and Smith, J.C. *Algorithms for discrete and continuous multi commodity flow network interdiction problems,* IIE Transactions, 39 (2007),15–26.

13. Lunday, B.J. and Sherali, H.D. *A dynamic network interdiction problem,* Informatica, 21(4) (2010), 553–574.

14. Lunday, B.J. and Sherali, H.D. *Network interdiction to minimize the maximum probability of evasion with synergy between applied resources,* Ann. Oper. Res. 196 (2012), 411–42.

15. Miller, L.E. (2011), *Catalog of network connectivity models,* http://w3.antd.nist.gov/wctg/netanal/netanal-netmodels.html. Accessed 14 July 2011.

16. Morton, D.P., Pan, F. and Saeger, K.J. *Models for nuclear smuggling interdiction,* IIE Transactions 38 (2007), 3–14.

17. Nandi, A.K. and Medal H.R. *Methods for removing links in a network to minimize the spread of infections,* Comput. Oper. Res., 69 (2016), 10–24.

18. Orlin, J. *Max flows in O(nm) time, or better.* STOC'13—Proceedings of the 2013 ACM Symposium on Theory of Computing, 765–774, ACM, New York, 2013.

19. Pan, F. *Stochastic network interdiction: Models and methods,* Ph.D. thesis, University of Texas at Austin, 2005.

20. Philips, C.A. *The network inhibition problem*, Proceeding of 25th ACM Symposium of Theory of Computing, San Diego, CA. (1993), 776–785.

21. Salmeron, J. (2012), *Deception tactics for network interdiction: A multi objective approach,* Networks, 60(1), 45–58.

22. Sullivan, K.M., Morton, D.P., Pan, F. and Smith, J.C. (2011), *Securing a border under asymmetric information,* Naval Res. Logist. 61(2) (2014), 91–100.

23. Sullivan, K.M., Smith J.C. and Morton, D.P. *Convex hull representation of the interdiction problem,* Math. Program. 145(1-2) (2014), 349–376.

24. Towle, E. and Luedtke, J. *New solution approaches for the maximum-reliability stochastic network interdiction problem,* Comput. Manag. Sci. 15(3-4) (2018), 455–477.

25. Wollmer, R.D. *Removing arcs from a Network,* Oper. Res. , 12 (1964), 934–940.

26. Wood, R.K. *Deterministic network interdiction problem,* Math. Comp. Model., 17 (1993), 1–18.