RC25638 (WAT1611-044) November 10, 2016
Mathematics

IBM Research Report

Network Disconnection Games

Mourad Baïou
CNRS and Université Clermont II
Campus des Cézeaux BP 125
63173 Aubière Cedex, France

Francisco Barahona
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598 USA

NETWORK DISCONNECTION GAMES

MOURAD BAÏOU AND FRANCISCO BARAHONA

Abstract. We study network security games arising from placing checkpoints on a set of arcs to intercept every path between two distinguished nodes s and t. First we study a two-person zero-sum game where one player, the attacker, chooses a set of arcs that intercepts every path from s to t, then a second player, the inspector, chooses an arc to inspect trying to find the attacker. We give combinatorial polynomial algorithms to find optimal strategies for both players. Here the Nash-equilibrium payoff gives a measure of the resilience of the network to this type of attack.

Then we study a study a cooperative game where every player controls an arc and is able to place a checkpoint on it. We also assume that there is an adversary trying to travel from s to t. The value of a coalition S is the maximum number of disjoint st-cuts included in S. This is the number of independent ways that a coalition can use to intercept every path between s and t. We assume that if the adversary crosses a checkpoint it will be detected with some probability, thus a coalition having different independent ways to intercept every st-path has a higher probability of success than a coalition that can intercept in only one way. We give polynomial combinatorial algorithms for testing membership to the core and for computing the nucleolus. This analysis shows which are the most important locations to put checkpoints.

1. Introduction

For a directed graph $G = (V, A)$ with two distinguished nodes s and t, we study network security games arising from placing checkpoints on a set of arcs to intercept every path from s to t. This type of games relates to the security of communication and transportation networks. Here we study first a two-person zero-sum game. Two-person zero-sum games in a network have been studied in [26], [1] and [15]. The first paper presents a game where a player picks a path from s from t, and a second player picks an arc trying to intercept the first player. The other two papers deal with a game where a player picks a spanning tree in a network, and a second player picks an edge trying to intercept the first player. The interception probability gives a measure of the network's security. In our case we assume that one attacker chooses a set of arcs that intercept every path from s from t, then a second player, the inspector, picks an arc trying to find the attacker. Here the probability of success gives a measure of the resilience of the network to this type of attack. Also the structure of both players strategies gives insight on their expected behavior. We give polynomial combinatorial algorithms to find optimal strategies for both the attacker and the inspector. We call this the *Network Interception Game.*

Next we study a cooperative game where each agent controls an arc of the network and is able to place a checkpoint at it, also we assume that there is an adversary trying

Date: November 4, 2016.

Key words and phrases. network disconnection, two-person games, shortest path, minimum cut, cooperative games.

to travel from s to t. If the adversary crosses a checkpoint there is no certitude that she will be detected, but there is a probability of detection, thus she might be able to cross several checkpoints before being detected. The value of a coalition $S \subseteq A$ is the maximum number of disjoint st-cuts included in S. This is the number of independent ways that the coalition can use to intercept every st-path. Thus a coalition having different independent ways to intercept every st-path has a higher probability of success than a coalition that can intercept in only one way. We call this the *Network Disconnection Game*. We give polynomial combinatorial algorithms for testing membership to the core and to compute the nucleolus of this game. This analysis shows which are the most important locations where to place checkpoints. This game is in some sense dual to a flow game studied in [16], where the value of a coalition $S \subseteq A$ is the maximum number of disjoint st-paths included in S. The core of this flow game has been studied in [16], and the nucleolus has been studied in [7] and [21]. This game is also related to *Path Disruption Games* (PDG) that have been introduced in [3] and [2]. In this case the value of a coalition is one if it controls a set of arcs whose removal disconnects s and t, and the value is zero otherwise. For PDG a polynomial algorithm to compute the ϵ-core was given in [2].

Other references for the study of combinatorial games are [19] and [8]. Other work on the nucleolus of combinatorial games appears in [25], [13], [17], [10]. See also the surveys [5] and [6].

This paper is organized as follows. In Section 2 we give some notation and review some network flow techniques. In Section 3 we study the Network Interception Game. In section 4 we study a cooperative Network Disconnection Game. In Section 5 we discuss some extensions of this work.

2. Preliminaries

Here we define some notation and recall some results from Network Flows. All this material will be used in the following sections. A reference for Network Flows theory is [23].

Let $G = (V, A)$ be a directed graph, for $S \subseteq V$, we denote by $\delta^+(S)$ the set $\delta^+(S) = \{(u, v) \in A \,|\, u \in S, v \notin S\}$, also $\delta^-(S)$=$\delta^+(V \setminus S)$. If there are two distinguished vertices s and t, for a set $S \subset V$, with $s \in S$, $t \notin S$, the set of arcs $\delta^+(S)$ is called an *st-cut*. We use $\delta^+(u)$ (resp. $\delta^-(u)$) instead of $\delta^+(\{u\})$ (resp. $\delta^-(\{u\})$).

An st-path is a sequence of arcs $(u_1, u_2), (u_2, v_3), \ldots, (u_{k-1}, u_k)$, where $s = u_1$, $u_k = t$, and all nodes $\{u_i\}$ are distinct.

For a set $S \subseteq A$, its incidence vector $x^S \in \mathbb{R}^A$ is defined by $x^S(a) = 1$ if the arc $a \in S$, and $x^S(a) = 0$ otherwise.

For a vector $x \in \mathbb{R}^A$ and for $S \subseteq A$, we use $x(S)$ to denote $x(S) = \sum_{a \in S} x(a)$. For a graph $G = (V, A)$ we use n to denote $n = |V|$, and m to denote $|A|$.

2.1. **Shortest paths and cut packings.**

We need a linear programming formulation of the shortest path problem, as follows. Let B be a matrix whose rows are the incidence vectors of all st-cuts. For a weight function $w : A \to \mathbb{R}_+$ consider the linear program below

$$\min wx \tag{1}$$
$$Bx \geq 1 \tag{2}$$
$$x \geq 0, \tag{3}$$

its dual is

$$\max y1 \tag{4}$$
$$yB \leq w \tag{5}$$
$$y \geq 0. \tag{6}$$

For the primal problem there is a variable $x(u, v)$ for each arc (u, v). For the dual problem there is a variable y_C for each st-cut C. This can be solved with the following primal-dual version of Dijkstra's algorithm [9].

Dijkstra's Algorithm

Step 0. Set $d = w$, $S = \{s\}$, $x(u, v) = 0$ for all $(u, v) \in A$, $y_C = 0$ for each st-cut C.
Step 1. Let $(k, l) = \text{argmin}\{d(u, v) \mid (u, v) \in \delta^+(S)\}$.
Set
$predecesor(l) = k$,
$y_C = d(k, l)$, with $C = \delta^+(S)$.
$d(u, v) \leftarrow d(u, v) - y_C$, for all $(u, v) \in C = \delta^+(S)$,
$S \leftarrow S \cup \{l\}$.
Step 2. If $l \neq t$ go to Step 1.
If $l = t$, use the predecessors to retrace a path from s to t, set $x(u, v) = 1$, for each arc (u, v) in this path, and stop.

At the end, the set of arcs (u, v) with $x(u, v) = 1$, form a directed path from s to t, so x satisfies (2)-(3). At every iteration the algorithm produces a vector y that satisfies (5)-(6). It is easy to see that the vectors x and y satisfy the complementary slackness conditions of Linear Programming, this proves the optimality of x and y.

The dual problem (4)-(6) gives a *maximum packing of st-cuts* with arc-capacities w. Notice that if the weights w are integer, then the dual vector y produced by this algorithm is also integral. So in particular if w is the vector of all ones, the primal solution gives an st-path of minimum cardinality, and the dual solution gives the maximum number of disjoint st-cuts contained in A. This will be used in Sections 3 and 4.

This algorithm also shows that the extreme points of the polyhedron defined by (2)-(3) are the incidence vectors of all st-paths.

2.2. **A network flows formulation.** Assume that $w(a) > 0$ for all $a \in A$. Then the set of optimal solutions of (1)-(3) corresponds to the set of optimal solutions of the network flows problem below.

$$\min \sum w(u,v)x(u,v) \tag{7}$$

$$x(\delta^-(v)) - x(\delta^+(v)) = \begin{cases} -1 & \text{if } v = s, \\ 0 & \text{if } v \neq s, t, \\ 1 & \text{if } v = t, \end{cases} \tag{8}$$

$$x(u,v) \geq 0 \quad \text{for all } (u,v) \in A \ . \tag{9}$$

Let A_0 be the set of arcs (u,v) with $x(u,v) = 0$, for every optimal solution of (7)-(9). Then the set of optimal solutions is defined by the system below. This will be used in Section 4.

$$x(\delta^-(v)) - x(\delta^+(v)) = \begin{cases} -1 & \text{if } v = s, \\ 0 & \text{if } v \neq s, t, \\ 1 & \text{if } v = t, \end{cases} \tag{10}$$

$$x(u,v) \geq 0 \quad \text{for all } (u,v) \in A, \tag{11}$$

$$x(u,v) = 0 \quad \text{for all } (u,v) \in A_0 \ . \tag{12}$$

2.3. **Network flows feasibility.** In the next sub-section we need a way to decide if the system below has a solution.

$$x(\delta^-(v)) - x(\delta^+(v)) = b(v), \text{ for all } v \in V,$$
$$x(u,v) \geq 0 \quad \text{for all } (u,v) \in A.$$

Here we have a directed graph $G = (V, A)$, $b : V \to \mathbb{R}$, and we assume that $\sum_v b(v) = 0$. Let $b^+(v) = \max\{b(v), 0\}$, and $b^-(v) = \max\{-b(v), 0\}$, for each $v \in V$. To test if the system above has a solution, we add two nodes s and t. Then for each node v with $b(v) < 0$, we add an arc (s, v) with capacity $b^-(v)$, and for each node v with $b(v) > 0$ we add an arc (v, t) with capacity $b^+(v)$. We give infinite capacity to all original arcs. Then we look for a maximum flow from s to t.

Let $\alpha = \sum_v b^+(v) = \sum_v b^-(v)$. If the flow value is α, then there is a solution, otherwise there is a minimum cut whose capacity is less than α. Now we have to discuss the structure of this cut. The cut is $\delta^+(S \cup \{s\})$, where $S \subset V$. Its capacity is

$$b^+(S) + b^-(T) = b^+(S) + \alpha - b^-(S) = \alpha + b(S) < \alpha,$$

where $T = V \setminus S$. Thus $b(S) < 0$, and since we obtained a minimum cut, we have a set $S \subset V$ with $\delta^+(S) = \emptyset$ and such that $b(S)$ is minimum. This will be used in the next sub-section. A maximum flow and a minimum cut can be found in $O(nm \log(n^2/m))$ time, see [12] and [23].

2.4. **Parametric flows.** In Section 4 we will need to find the maximum value of λ so that the system below has a solution.

$$x(\delta^-(v)) - x(\delta^+(v)) = b(v) + \lambda d(v), \text{ for all } v \in V,$$
$$x(u,v) \geq 0 \quad \text{for all } (u,v) \in A.$$

Here we have a directed graph $G = (V, A)$, $b : V \to \mathbb{R}$, $d : V \to \mathbb{R}$. We assume that $\sum_v b(v) = 0$, $\sum_v d(v) = 0$, and that for $\lambda = 0$ the system is feasible. We also assume that we know a large value λ_M such that the system is infeasible.

In general if the system is infeasible for some value $\bar{\lambda} > 0$, we have seen in the previous sub-section that there is a set $S \subset V$ with $\delta^+(S) = \emptyset$, $b(S) + \bar{\lambda} d(S) < 0$. Since the system is feasible for $\lambda = 0$, we should have $b(S) \geq 0$, and $d(S) < 0$. To have feasibility, we should impose $b(S) + \lambda d(S) \geq 0$, or $\lambda \leq b(S)/(-d(S))$. Thus

$$\lambda = \min \frac{b(S)}{-d(S)}, \tag{13}$$

where the minimum is taken among all sets $S \subset V$, with $\delta^+(S) = \emptyset$ and $d(S) < 0$. This minimum can be found with Newton's algorithm below, [22].

Newton's method

Step 0. Set $\lambda = \lambda_M$.
Step 1. Find $\bar{S} = \text{argmin}\{b(S) + \lambda d(S)\}$, with $S \subset V$, $\delta^+(S) = \emptyset$ and $d(S) < 0$.
Step 2. If $b(\bar{S}) + \lambda d(\bar{S}) < 0$, then update λ as

$$\lambda = \frac{b(\bar{S})}{-d(\bar{S})}$$

and go to Step 1.
Otherwise $b(\bar{S}) + \lambda d(\bar{S}) = 0$, and we stop.

The set $\bar{S}$ in Step 1 can be found using a minimum cut algorithm, as seen in Subsection 2.3. Newton's method has been analysed in [22]. It is easy to show that if $S_1, \ldots, S_k$ is the sequence of sets produced, then $|d(S_i)| > |d(S_{i+1})|$, for $i = 1, \ldots, k-1$, see [4] for instance. Thus if $|d(S)| \leq m$, for each set $S \subset V$, and d is integer valued, then Newton's method takes at most m iterations, recall that $m = |A|$.

3. A zero-sum game

Consider an attacker that tries to intercept all paths from s to t. We assume that he concentrates on intercepting the arcs in an st-cut. So for each st-cut C, the attacker will choose it with probability y_C. The network owner (the inspector), has to develop an inspection strategy. He inspects an arc a with probability x_a. If the inspector is at arc a, there is a probability p_a of detecting the attacker if the latter is at arc a. The inspector has to find a probabilistic arc-inspection strategy which maximizes the average probability of detecting the attacker. The attacker has to find a cut-selection strategy that minimizes the average probability of being detected. This probability gives a measure of the resilience of the network to this type of attack. Von Neumann's classic Minimax Theorem [20] shows the existence of a Nash-equilibrium for this type of games. This can be computed by solving a linear program. One difficulty here is that this linear program has an exponential number of variables, that could be treated with the ellipsoid method [14]. Here we show that there is no need for such an impractical algorithm like the ellipsoid method, namely Dijkstra's algorithm gives a way to compute both strategies. We call this the *Network Interception Game*.

Let P a diagonal matrix that contains the probabilities $\{p_a\}$, and x a row vector, then xP is a row with components $\{x_a p_a\}$. Let D be a matrix whose columns are the incidence vectors of all st-cuts, and y a column vector whose components are $\{y_C\}$. Then Dy is a column whose component associated with an arc a is the probability that the attacker will be at arc a. Thus $xPDy$ is the probability that the attacker will be detected. Thus we concentrate on the following two-person game:

$$\max_x \; \min_y \; xPDy \tag{14}$$
$$\sum x_a = 1, \tag{15}$$
$$\sum y_C = 1, \tag{16}$$
$$x \geq 0, \tag{17}$$
$$y \geq 0. \tag{18}$$

If we fix y we have

$$\max_x \; xPDy \tag{19}$$
$$\sum x_a = 1, \tag{20}$$
$$x \geq 0. \tag{21}$$

And its dual is

$$\min_\mu \mu \tag{22}$$
$$\mu \geq p_a \sum \{y_C \mid a \in C\} \quad \forall a. \tag{23}$$

Then (14)-(18) is equivalent to

$$\min_{\mu,y} \; \mu \tag{24}$$
$$\mu - p_a \sum \{y_C \mid a \in C\} \geq 0 \quad \forall a, \tag{25}$$
$$\sum y_C = 1, \tag{26}$$
$$y \geq 0. \tag{27}$$

This can be written as

$$\min_{\mu,y} \; \mu \tag{28}$$
$$\sum \{y_C \mid a \in C\} \leq \frac{\mu}{p_a} \quad \forall a, \tag{29}$$
$$\sum y_C = 1, \tag{30}$$
$$y \geq 0. \tag{31}$$

Here we are looking for the minimum value of μ such that there is a packing of *st*-cuts of value one, with arc-capacities $\{\mu/p_a\}$. We assume that the numbers $\{1/p_a\}$ are rational, and that we have an integer number ρ such that the numbers $\{\rho/p_a\}$ are integers. Consider the linear program below.

$$\max \sum_C y_C \tag{32}$$

$$\sum \{y_C \mid a \in C\} \leq \frac{\rho}{p_a} \quad \forall a \tag{33}$$

$$y \geq 0. \tag{34}$$

Here we are looking for a maximum packing of st-cuts with arc-capacities $\{\rho/p_a\}$. This is the dual problem studied in Sub-section 2.1. Thus let λ be the value of a shortest st-path with weights $\{\rho/p_a\}$, and let $\bar{y}$ be an optimal solution of (32)-(34). Then if we set $\mu = \rho/\lambda$ and $\hat{y} = (1/\lambda)\bar{y}$, we have that $(\mu, \hat{y})$ is an optimal solution of (28)-(31).

We have the values for the variables y, now we have to find the values for the variables x. Let $\mathcal{P}$ be a shortest st-path with arc weights $\{\rho/p_a\}$. We set $\hat{x}(a) = \rho/(\lambda p_a)$ if $a \in \mathcal{P}$, and $\hat{x}(a) = 0$ otherwise. Thus

$$\sum_{a \in A} \hat{x}(a) = \sum_{a \in \mathcal{P}} \hat{x}(a) = \frac{1}{\lambda} \sum_{a \in \mathcal{P}} \frac{\rho}{p_a} = \frac{1}{\lambda}\lambda = 1.$$

The complementary slackness conditions imply

$$\sum \{\hat{y}_C \mid a \in C\} = \frac{\mu}{p_a} \quad \forall a \in \mathcal{P}.$$

Thus

$$\hat{x}PD\hat{y} = \sum_{a \in \mathcal{P}} \frac{\rho}{\lambda p_a} p_a \sum \{\hat{y}_C \mid a \in C\} = \sum_{a \in \mathcal{P}} \frac{\rho}{\lambda p_a} p_a \frac{\rho}{\lambda p_a} = \frac{\rho}{\lambda} = \mu.$$

This shows that the pair $(\hat{x}, \hat{y})$ is an optimal solution of (14)-(18). Thus we have the following.

Theorem 1. *Optimal strategies for both players can be computed in polynomial time. The inspector strategy can be obtained from a shortest st-path with arc weights* $\{\rho/p_a\}$*. The attacker strategy can be obtained from a maximum packing of st-cuts with arc capacities* $\{\rho/p_a\}$*.*

This theorem not only gives a way to compute the strategies, but also shows the structure of the strategies of both players, this gives insight on their expected behavior.

4. A Cooperative Game

Given a directed graph $G = (V, A)$ with two distinguished nodes s and t, we assume that each player controls an arc and is able to place a checkpoint on it. We also assume that there is an adversary trying to travel from s to t. If the adversary crosses a checkpoint there is no certitude of detecting her, but there is a probability of detection. Thus she might be able to cross several checkpoints before being detected. We define the *Network Disconnection Game* $(A, \mathbf{v})$, where the characteristic function $\mathbf{v} : 2^A \to \mathbb{R}_+$, gives for each coalition S, the maximum number of disjoint st-cuts included in S. This is the number of independent ways that the coalition S can use to intercept every path from s to t. To motivate this definition, notice that a coalition that can intercept every st-path in several independent ways has a higher probability of success than a coalition that can intercept in only one way. The core [11] and the nucleolus [24] of a game are two notions introduced seeking stability, we study their algorithmic aspects below.

4.1. **The core.** Its definition is based on the following stability condition: No subgroup of players does better if they break away from the joint decision of all players to form their own coalition. Thus the core of this game is the polytope defined below.

$$x(A) = \mathbf{v}(A)$$
$$x(S) \geq \mathbf{v}(S),\ \forall S \subseteq A.$$

Here $x(a)$ represents the amount paid to player a. First we need a simpler description of the core as follows.

Lemma 2. *Let k be the length of an st-path of minimum cardinality. Then the core is determined by*

$$x(A) = k, \tag{35}$$
$$x(C) \geq 1,\ \textit{for each } st\textit{-cut } C, \tag{36}$$
$$x \geq 0. \tag{37}$$

Proof. It follows from Dijkstra's algorithm in Sub-section 2.1, that the value of a minimum cardinality st-path is equal to the maximum number of disjoint st-cuts included in A, therefore $\mathbf{v}(A) = k$.

Consider now an inequality

$$x(S) \geq \mathbf{v}(S), \tag{38}$$

for $S \subset A$. Let $\mathbf{v}(S) = q > 0$, then S contains q disjoint st-cuts $C_1, \ldots, C_q$. Thus (38) can be obtained by adding $x(C_i) \geq 1$, for $i = 1, \ldots, q$, and $x(a) \geq 0$ for $a \in S \setminus (\cup_i C_i)$.

If $\mathbf{v}(S) = 0$, then $x(S) \geq 0$ can be obtained by adding $x(a) \geq 0$, for $a \in S$. □

Inequalities (36)-(37) correspond to (2)-(3), thus the extreme points of the core are the incidence vectors of all minimum cardinality st-paths. Also any vector x in the core corresponds to a minimum cost flow from s to t. The flow costs are all equal to one. We summarise this below.

Theorem 3. *The core is also defined by the system*

$$x(\delta^-(v)) - x(\delta^+(v)) = \begin{cases} -1 & \textit{if } v = s, \\ 0 & \textit{if } v \neq s, t, \\ 1 & \textit{if } v = t, \end{cases} \tag{39}$$
$$x(u, v) \geq 0 \quad \textit{for all } (u, v) \in A, \tag{40}$$
$$x(u, v) = 0 \quad \textit{for all } (u, v) \in A_0\ . \tag{41}$$

Here A_0 is the set of arcs that do not belong to any shortest st-path.

Theorem 4. *For the Network Disconnection Game, the core is non-empty if and only if there is a path from s to t.*

Theorem 5. *Given a vector $\bar{x}$, we can test whether $\bar{x}$ belongs to the core in polynomial time.*

Proof. To test if $\bar{x}$ satisfies (35)-(37), we have to solve a shortest path problem, and a minimum cut problem.

Alternatively we can test if $\bar{x}$ satisfies (39)-(41). For this we need to identify the set A_0, this reduces to a sequence of shortest path problems. □

4.2. **The nucleolus.** For a coalition S and a vector $x \in \mathbb{R}^A$, their *excess* is $e(x, S) = x(S) - \mathbf{v}(S)$. A vector $x \in \mathbb{R}^A$ with $x(A) = \mathbf{v}(A)$ is called an *allocation.* The nucleolus is the allocation that lexicographically maximizes the vector of non-decreasingly ordered excesses, cf. [24], thus in this sense, it is the fairest allocation. The nucleolus can be computed with a sequence of linear programs as follows, cf. [18]. First solve

$$\begin{aligned} &\max \epsilon \\ &x(S) \geq \mathbf{v}(S) + \epsilon, \quad \forall S \subset A \\ &x(A) = \mathbf{v}(A). \end{aligned}$$

Let ϵ_1 be the optimal value of this, and $P_1(\epsilon_1)$ be the polytope defined above, with $\epsilon = \epsilon_1$, i.e., $P_1(\epsilon_1)$ is the set of optimal solutions of the linear program above. For a polytope $P \subset \mathbb{R}^A$ let

$$\mathcal{F}(P) = \{S \subseteq A \,|\, x(S) = y(S), \forall x, y \in P\}$$

denote the set of coalitions fixed for P. In general given ϵ_{r-1} we solve

$$\max \epsilon \tag{42}$$

$$x(S) \geq \mathbf{v}(S) + \epsilon, \forall S \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1})) \tag{43}$$

$$x \in P_{r-1}(\epsilon_{r-1}). \tag{44}$$

We denote by ϵ_r the optimal value of this, and $P_r(\epsilon_r)$ the polytope above with $\epsilon = \epsilon_r$. We continue for $r = 2, ..., |E|$, or until $P_r(\epsilon_r)$ is a singleton. Notice that each time the dimension of $P_r(\epsilon_r)$ decreases by at least one, so it takes at most $|E|$ steps for $P_r(\epsilon_r)$ to be a singleton.

The following lemma gives a simpler formulation of (42)-(44).

Lemma 6. *Instead of solving* (42)-(44)*, we can solve*

$$\max \epsilon \tag{45}$$

$$x(C) \geq 1 + \epsilon, \text{ for each } st\text{-cut } C \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1})), \tag{46}$$

$$x(a) \geq \epsilon \text{ for each arc } a \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1})), \tag{47}$$

$$x \in P_{r-1}(\epsilon_{r-1}). \tag{48}$$

Proof. Consider $S \subset A$, with $S \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$. First assume that $\mathbf{v}(S) = q > 0$, and let $C_1, \ldots, C_q$ be a set of disjoint st-cuts included in S.

If there is at least one of them, C_1 say, with $C_1 \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$, then $x(S) \geq \mathbf{v}(S)+\epsilon$ can be obtained as the sum of $x(C_1) \geq 1 + \epsilon$, $x(C_i) \geq 1$ for $i = 2, \ldots, q$, and $x(a) \geq 0$ for $a \in S \setminus (\cup_j C_j)$.

If $C_i \in \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$, for all i, then there is an arc $\bar{a} \in S \setminus (\cup_j C_j)$ with $\bar{a} \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$. Then $x(S) \geq \mathbf{v}(S) + \epsilon$ can be obtained as the sum of $x(C_i) \geq 1$ for $i = 1, \ldots, q$, $x(a) \geq 0$ for $a \in S \setminus (\cup_j C_j)$, $a \neq \bar{a}$, and $x(\bar{a}) \geq \epsilon$.

If $\mathbf{v}(S) = 0$, then there is an arc $\bar{a} \in S$ with $\bar{a} \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$, then $x(S) \geq \epsilon$ can be obtained adding $x(\bar{a}) \geq \epsilon$, and $x(a) \geq 0$ for $a \in S$, $a \neq \bar{a}$. ☐

Now we plan to show that to find ϵ_r it is enough to work with constraints $x(a) \geq \epsilon$, for $a \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$, then the constraints $x(\delta^+(S)) \geq 1 + \epsilon$, for $\delta^+(S) \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$, will be automatically satisfied. We treat this in the lemma below. Notice that for $a \in \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$ we denote by $l(a)$ the fixed value that $x(a)$ should take.

Lemma 7. *Assume that x is in the core, $x(a) \geq \epsilon$ for each $a \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$, $x(a) = l(a)$ for $a \in \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$. Then $x(\delta^+(S)) \geq 1+\epsilon$ for $\delta^+(S) \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$.*

Proof. Consider $S \subset V$, $s \in S$, $t \notin S$. The system (39)-(41) implies $x(\delta^+(S)) - x(\delta^-(S)) = 1$, or $x(\delta^+(S)) = 1 + x(\delta^-(S))$. We have two cases:

- If $a \in \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$ for all $a \in \delta^-(S)$ then $\delta^+(S) \in \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$.
- If there is an arc $\bar{a} \in \delta^-(S)$ with $\bar{a} \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$, then $x(\bar{a}) \geq \epsilon$. Therefore $x(\delta^+(S)) = 1 + x(\delta^-(S)) \geq 1+\epsilon$. □

The lemma above shows that to compute the nucleolus, at each step we have to look for the maximum value of λ such that the system below has a solution.

$$x(\delta^-(v)) - x(\delta^+(v)) = \begin{cases} -1 & \text{if } v = s, \\ 0 & \text{if } v \neq s,t, \\ 1 & \text{if } v = t, \end{cases} \tag{49}$$

$$x(u,v) = l(u,v),\ \forall (u,v) \in \mathcal{F}(P_{r-1}(\epsilon_{r-1})), \tag{50}$$

$$x(u,v) \geq l(u,v) + \lambda,\ \forall (u,v) \in A_r = A \setminus \mathcal{F}(P_{r-1}(\epsilon_{r-1})), \tag{51}$$

$$\lambda \geq 0. \tag{52}$$

Here we assume that for $(u,v) \in \mathcal{F}(P_{r-1}(\epsilon_{r-1}))$, $x(u,v) = l(u,v)$. And for $(u,v) \in A_r$, $l(u,v) = \epsilon_{r-1}$.

This could be done with linear programming techniques, however now we show that it can be done in a combinatorial way, namely it reduces to a sequence of minimum cut problems. We define

$$x'(u,v) = x(u,v) - l(u,v) - \lambda, \text{ for } (u,v) \in A_r.$$

Then we have to find the largest value of λ such that the system below has a solution.

$$x'(\delta^-(v)) - x'(\delta^+(v)) = b(v) + \lambda d(v), \text{ for each } v \in V,$$
$$x' \geq 0.$$

Here the arc-set is A_r. We define $b'(s) = -1$, $b'(t) = 1$, and $b'(v) = 0$ if $v \neq s,t$. Then for each $v \in V$,

$$b(v) = b'(v) - \sum_{(u,v)\in A} l(u,v) + \sum_{(v,u)\in A} l(v,u),$$

and

$$d(v) = |\{(v,u) \in A_r\}| - |\{(u,v) \in A_r\}|.$$

Then the maximum value of λ can be found with Newton's method as in Sub-section 2.4. The value λ_M required in Sub-section 2.4 can be $\lambda_M = 2$, for instance. Each time that we apply the algorithm of Sub-section 2.4, we obtain a new set of arcs such that their associated variables should remain fixed. Thus it takes at most m times until all variables are fixed. This leads to the following.

Theorem 8. *Computing the nucleolus of the Network Disconnection Game requires $O(m^3 n \log(n^2/m))$ time.*

Proof. Since Newton's has to be applied at most m times, and each time requires at most m min-cut problems, we have $O(m^2)$ minimum cut problems. Since each of them requires $O(nm \log(n^2/m))$ time we obtain the bound. □

Once the nucleolus has been computed, the values of its components show which are the most important locations to put checkpoints.

5. Extensions

We have studied games defined on the arcs of a directed graph. A similar development can be done for undirected graphs, and for games defined on the nodes of a graph. We skip the details.

References

[1] H. Aziz, O. Lachish, M. Paterson, and R. Savani, *Wiretapping a hidden network*, in Internet and Network Economics, Springer, 2009, pp. 438–446.
[2] H. Aziz and T. B. Sørensen, *Path coalitional games*, arXiv preprint arXiv:1103.3310, (2011).
[3] Y. Bachrach and E. Porat, *Path disruption games*, in Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 1123–1130.
[4] W. H. Cunningham, *Optimal attack and reinforcement of a network*, Journal of the ACM (JACM), 32 (1985), pp. 549–561.
[5] I. Curiel, *Cooperative combinatorial games*, in Pareto optimality, game theory and equilibria, Springer, 2008, pp. 131–157.
[6] X. Deng and Q. Fang, *Algorithmic cooperative game theory*, in Pareto Optimality, Game Theory And Equilibria, Springer, 2008, pp. 159–185.
[7] X. Deng, Q. Fang, and X. Sun, *Finding nucleolus of flow game*, Journal of combinatorial optimization, 18 (2009), pp. 64–86.
[8] X. Deng and C. H. Papadimitriou, *On the complexity of cooperative solution concepts*, Mathematics of Operations Research, 19 (1994), pp. 257–266.
[9] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische mathematik, 1 (1959), pp. 269–271.
[10] U. Faigle, W. Kern, and J. Kuipers, *Note computing the nucleolus of min-cost spanning tree games is np-hard*, International Journal of Game Theory, 27 (1998), pp. 443–450.
[11] D. B. Gillies, *Solutions to general non-zero-sum games*, Contributions to the Theory of Games, 4 (1959), pp. 47–85.
[12] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum-flow problem*, Journal of the ACM (JACM), 35 (1988), pp. 921–940.
[13] D. Granot, M. Maschler, G. Owen, and W. R. Zhu, *The kernel/nucleolus of a standard tree game*, International Journal of Game Theory, 25 (1996), pp. 219–244.
[14] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.
[15] A. Gueye, J. C. Walrand, and V. Anantharam, *Design of network topology in an adversarial environment*, in Decision and Game Theory for Security, Springer, 2010, pp. 1–20.
[16] E. Kalai and E. Zemel, *Totally balanced games and games of flow*, Mathematics of Operations Research, 7 (1982), pp. 476–478.
[17] W. Kern and D. Paulusma, *Matching games: the least core and the nucleolus*, Mathematics of Operations Research, 28 (2003), pp. 294–308.
[18] A. Kopelowitz, *Computation of the kernels of simple games and the nucleolus of n-person games.*, tech. rep., DTIC Document, 1967.
[19] N. Megiddo, *Computational complexity of the game theory approach to cost allocation for a tree*, Mathematics of Operations Research, 3 (1978), pp. 189–196.
[20] J. v. Neumann, *Zur theorie der gesellschaftsspiele*, Mathematische Annalen, 100 (1928), pp. 295–320.
[21] J. Potters, H. Reijnierse, and A. Biswas, *The nucleolus of balanced simple flow networks*, Games and Economic Behavior, 54 (2006), pp. 205–225.
[22] T. Radzik, *Fractional combinatorial optimization*, in Handbook of combinatorial optimization, Springer, 1999, pp. 429–478.

[23] K. A. Ravindra, T. L. Magnanti, and J. B. Orlin, *Network flows: Theory, algorithms, and applications*, 1993.
[24] D. Schmeidler, *The nucleolus of a characteristic function game*, SIAM Journal on applied mathematics, 17 (1969), pp. 1163–1170.
[25] T. Solymosi and T. E. Raghavan, *An algorithm for finding the nucleolus of assignment games*, International Journal of Game Theory, 23 (1994), pp. 119–143.
[26] A. Washburn and K. Wood, *Two-person zero-sum games for network interdiction*, Operations Research, 43 (1995), pp. 243–251.

(M. Baïou) CNRS, and Université Clermont II, Campus des cézeaux BP 125, 63173 Aubière cedex, France.

E-mail address, M. Baïou: `baiou@isima.fr`

(F. Barahona) IBM T. J. Watson research Center, Yorktown Heights, NY 10589, USA.

E-mail address, F. Barahona: `barahon@us.ibm.com`