Budgeted Adversarial Network Resource Utilization Games

Yi Zhang and Sanjiv Kapoor(✉)

Illinois Institute of Technology, Chicago, IL 60616, USA
yzhan257@hawk.iit.edu, kapoor@iit.edu

Abstract. This paper studies budgeted adversarial resource utilization game, where one of the player's (designer) strategy is the utilization of resources while the other player's (adversary) role is to police the resources for misuse. In this context, we consider routing games where a designer plans routes on a computer network and the adversary intercepts the routes on the network. Another example is in determining adversarial strategies to block access to travel or resources that may be considered to pose a risk to society, e.g. during a pandemic where the population (designer) goal may not be coincide with the societal goal of minimizing accessing a banned resource. We model this as a zero-sum game with constraints on the adversary or designer budgets. While zero-sum games can be solved using linear programs, we illustrate faster combinatorial methods to solve the problem. We first consider the resource access problem game on a bipartite graph where both the designer and the adversary have independent budget constraints and distinct costs and show a fast algorithm to determine a Nash equilibrium. We also consider the situation where the designer would strategize on paths in a general graph. In this application of determining network paths, where the adversary would attack edges in order to block the paths, we also discuss the case of multiple designers and, in particular show faster algorithms when there are 2 designers. These results utilize properties of minimum cuts in 2-commodity flow routing.

Keywords: Security · Network · Game theory · Nash equilibrium

1 Introduction

In this paper we consider a budgeted adversarial resource utilization game, where a designer plans the utilization of resources when facing an adversary. The adversary is a user who attempts to prevent the designer from using resources. The adversary and/or the designer have budgets. Typical examples would include public health restrictions on movements during pandemics (where analysis of behavior and policies using game theory has been investigated [4,11] or ensuring network security. While no moral subjective is assigned to the players, a rational approach of a traveller would be to adopt a route which has the least stringent

F. Fang and F. Shu (Eds.): GameNets 2022, LNICST 457, pp. 339–365, 2022.
https://doi.org/10.1007/978-3-031-23141-4_25

governmental restrictions when planning airline trips across boundaries, while inadvertently carrying the virus. The governing public health policy takes on the role of an adversary with tests and mandates within the travel network, placing restrictions on edges to prevent virus transmission through human or even goods transportation.

Security issues in resource allocations games have been considered extensively. Applications in communication and transportation networks have been studied via game theoretic analysis in [3,12,15,16]. An example problem is securing the connectivity in communication systems on a graph where a set of nodes is to be connected using a spanning tree. The resources are edges and a subset of edges provides utility to the designer. The adversary may attack one or more edges, each edge having a cost of attack. The adversary has a budget which cannot be exceeded.

This paper considers network security games in the larger context of resource utilization adversarial games. Given a set of resource elements, a collection of resource subsets defines designer strategies. Each resource element has a cost of attack and the adversary has a budget. This is a zero-sum game and it is required to determine a Nash equilibrium in this model with strategies over a continuous range. Pure Nash equilibrium with integral feasible strategies, where attacker pays for entire cost of edges if present in strategy set, need not always exist. Mixed equilibrium over the integral strategy sets exists but additional budget constraints impact the optimality. Budget utilization leading to fractional solutions provides for challenging problems, while mixed solutions over finite integral strategies is not considered and left for further work.

In this paper consider an important application of this framework, a network routing game where a designer plans routes (paths) on a network and the adversary intercepts the routes on the network. The designer has to route paths between a source-destination pair and can choose multiple paths to satisfy her unit requirement. The fractional solution can also be interpreted as the probability of the designer choosing the particular path. The adversary chooses edges to attack simultaneously. Each attack on an edge incurs a cost. The attack on edges is fractional and may contain multiple edges as long as the budget allows. As indicated above, the role of budget is particularly interesting and there exists an example where paying for the entire edge does not result in a Nash equilibrium. The role of the designer and adversary may be reversed as in the case of drug-trafficking [19].

Computing Nash equilibrium in general games is a hard problem, even for mixed strategies. The problem is PPAD-hard [5,14] and polynomial solutions are not known. An important sub-class is the class of zero-sum games. Zero-sum games are easier to compute using linear programming formulation and require time polynomial in the size of strategy space. However, in most cases the strategy space is exponential in the description of the problem, as in the case of the routing or connectivity games. To obtain polynomial solutions, one approach is to utilize separation oracles and the ellipsoid method [6] for solving linear programs, which unfortunately has substantial complexity, albeit polynomial. Furthermore, additional budget constraints on the strategies adds additional

complication and may require enumeration of the feasible sets that form the support set. An approach to tackle the exponential space is to employ duality theory to identify problem characteristics.

In this paper we consider budgeted versions of zero-sum games and describe polynomial method to solve these games. While the complexity of zero-sum games with succinct representation has been considered, e.g. for Colonel Blotto games [1] and the ellipsoid method is used, the budgeted case is not addressed in that research. Costs on elements have been addressed in the case of matroidal zero-sum games where the defender or designer chooses a basis B of the matroid and the adversary incurs a cost of attack on elements of the subsets [2]. That research considers a utility function that incorporates the costs without any explicit budget constraint. Additionally, research [8] has also considered the problem of designing a spanning tree and attacks on the tree using a single edge attack. There is a cost of attack on edges which, again, is incorporated into the objective and the paper describes how to identify a critical set of edges over which a mixed equilibrium can be determined. Addition of budgets has direct impact on the feasible polyhedron describing the strategy space and thus complicates the problem. Related security games have been studied by [20], where the adversary only picks one target to attack. Additionally, research in [17] [18] extend the model to incorporate multiple attack targets and to target network security; however targets or elements are not distinguished by differing costs or weights. Arguably, adversaries typically have budgets within which they attempt to inflict as much damage on resources, say a communication network, and our current approach models real life scenarios effectively. Budgeted attacks have also been considered in [7] where constrained optimization methods are used to determine the expected benefits of the adversary and defender. However, no specific form of the underlying problem are used and the algorithms are not guaranteed to be polynomial.

The key challenge in the analysis of designer-adversary strategies is the exponential number of strategies, even when the game can be described succinctly. As we illustrate, it is possible to solve classes of these problems using combinatorial interpretations based on primal-dual formulations. The optimization programs for the designer and attacker are constrained and interdependent. The key technique we employ is to combine both the defender's and adversary's primal and dual constraints in a complementary fashion to formulate a composite linear program and its dual. We then apply a primal-dual algorithm that is shown to be efficient. We first illustrate this for the case of resource access bipartite networks with independent designer and adversary costs and budgets. Additionally for our general network application problem, the routing problem, we utilize primal-dual programs to characterize the Nash equilibrium solutions. The solutions based on these characterizations can be obtained by simple combinatorial max-flow algorithms that also identify the minimum cut in a graph. In particular we also consider the case when there are multiple designers and provide combinatorial solutions for the case of 2-designers. This is a particularly interesting case as 2-commodity network flows have interesting cut theorems and combinatorial algorithms exist for this problem [9,10].

2 Preliminaries and Results

2.1 Problem Definitions and Notations

We first define the games that we will consider in this paper.

Bipartite Routing Adversarial Game: In the *Bipartite Routing Adversarial Game* we consider a bipartite network $N(E, d, c)$. $E = \{e_1, e_2, \cdots, e_m\}$ is a bipartition of m edges, where each edge e_i is associated with a positive design cost $d[i]$ and attack cost $c[i]$. $d, c \in \mathbb{R}_+^m$ are the cost vectors. We denote the designer's and the adversary's budget by B_D and B_A, respectively.

The flow design is represented by a vector $f \in [0,1]^m$ with the requirement $\sum_{i=1}^m f[i] = 1$, where $f[i]$ is the flow amount on edge e_i. For each edge e_i with flow amount $f[i]$, the designer's cost is $d[i]f[i]$. We say a design f is within budget if $\sum_{i=1}^m d[i]f[i] \leq B_D$.

An attack is a vector $X \in [0,1]^m$ indicating the attack on edges, where $X[i]$ denotes the attack, or attack level, on edge e_i, the fraction of flow captured on e_i. We say an attack X is within budget if $\sum_{i=1}^m c[i]X[i] \leq B_A$.

The adversary's benefit is the sum of flow she captures over the bipartition and defined as $\sum_{j=1}^m f[j] \cdot X[j]$. This is a zero-sum game where the adversary maximizes her benefit by choosing an attack strategy and the designer minimizes it by choosing a flow design.

Network Routing Adversarial Game: In the *Network Routing Adversarial Game*, we consider a network $N(G, s, t, c)$ where $G = (V, E)$ is an undirected graph, V is a set of vertices with $|V| = n$ and $E = \{e_1, e_2, \cdots, e_m\}$ is a set of m edges, where each edge e_i is associated with a positive adversary cost $c[i]$. The adversary has a budget $B > 0$. The source destination pair is $(s, t) \in V$. The flow designer is required to determine paths to route a normalized unit of traffic/flow over $\mathcal{P} = \{p_1, p_2, \cdots, p_k\}$, the collection of all k possible $s-t$ paths. In the case of r-commodity network, each designer i has her source and sink pair s_i, t_i and her set of path $\mathcal{P}_i$, where $|\mathcal{P}_i| = k_i$. The overall path set is denoted by $\mathcal{P}$, with $|\mathcal{P}| = k$ where $k = \sum_{i=1}^r k_i$.

The flow design is represented by a vector $f \in [0,1]^k$ with the requirement $\sum_{j=1}^k f[j] = 1$, where $f[j]$ is the flow amount on path p_j. We denote by $f_e[i] = \sum_{j|p_j \ni e_i} f[j]$ the flow on edge e_i, which is the summation of flow on every path containing e_i. In the case of multiple designers, $f_e[i]$ includes the flow of all designers.

An attack is a vector $X \in [0,1]^m$ indicating the attack on edges, where $X[i]$ denotes the attack, or attack level, on edge e_i. We say an attack X is within budget if $\sum_{i=1}^m c[i]X[i] \leq B$.

Given an attack $X \in [0,1]^m$, the impact vector of the attack over all paths is denoted by $X_{\mathcal{P}}$ where $X_{\mathcal{P}}[j] = \min\{\sum_{i|e_i \in p_j} X[i], 1\}$. $X_{\mathcal{P}}[j]$ is the attack on a path p_j, the fraction of flow on that path captured by the adversary, which is the summation of attacks on edges on that path, up to 1.

The adversary's benefit is the sum of fractions of flow on the paths she attacks: $\sum_{j=1}^k f[j] \cdot X_{\mathcal{P}}[j]$. This is a zero-sum game where the adversary maximizes her benefit by choosing an attack strategy and the designer minimizes it by choosing a flow design.

An $s-t$ cut C is a set of edges whose removal disconnects all paths from s to t and partitions V into 2 component S, T, where $S \cap T = \emptyset$, $s \in S$, $t \in T$. We define the cost of C to be $|C| = \sum_{e_i \in C} c[i]$. Given an input graph, let C_{min} be the min-cost $s-t$ cut.

2.2 Results

In this paper we obtain the following results:

1. Our first result is a combinatorial algorithm to solve the Bipartite Routing Adversarial Game with both designer and adversary budget, modeled via a bipartite graph, in $O(m \log m)$ steps. Note that the costs of edges incurred by the designer and the adversary (attacker) are different.
2. We design a combinatorial algorithm to solve the Network Routing Adversarial game with adversary budget constraint in $O(nm)$ steps.
3. We illustrate combinatorial methods for the 2-commodity game in the context of r-commodity Network Routing Adversary game with budgets. The time complexity is $O(n^3)$.

These methods have complexities that improve the standard linear programming approach. Some of the proofs are provided in the appendix.

3 Bipartite Routing Adversarial Game

In this section we look at the **Bipartite Routing Adversarial Game**. Given a bipartite network $N(E, d, c)$ where each edge $e_i \in E, \forall i = 1 \ldots m$ has a cost for designer $d[i]$ and cost for adversary $c[i]$. The designer and adversary has budget B_D, B_A respectively. We determine the designer's strategy f and the adversary's strategy X. We assume each adversary cost $c[i]$ and each designer cost $d[i]$ are unique and $d[i] \geq 1, \forall i$. We sort E in increasing order of $d[i]$ in $O(m \log m)$ steps if they are not sorted. In our method the designer will consider this sorted order of edges whenever her strategy requires it to add additional edges. We assume $B_D \geq d[1]$, the smallest designer cost, to ensure feasible flow design. We also assume $B_A \leq \sum_{i=1}^{m} c[i]$ for otherwise the adversary always captures the entire flow by fully attacking every edge.

Given the adversary's strategy X, the designer solves the following linear program with its dual:

$$\begin{aligned} \textbf{LP1}: \min & \sum_{i=1}^{m} X[i]f[i] \\ s.t. & \sum_{i=1}^{m} f[i] \geq 1 \\ & \sum_{i=1}^{m} d[i]f[i] \leq B_D \end{aligned} \qquad (1)$$

$$\begin{aligned} \textbf{LP2:} \max & \; X_D - B_D\Omega \\ s.t. & \; X_D - X[i] \leq d[i]\Omega, \forall i \end{aligned} \qquad (2)$$

Given the designer's strategy f, the adversary solves the following linear program with its dual:

$$\begin{aligned} \textbf{LP3:} \max & \sum_{i=1}^{m} f[i]X[i] \\ s.t. & \sum_{i=1}^{m} c[i]X[i] \le B_A \\ & X[i] \le 1, \forall i \end{aligned} \qquad (3) \qquad \begin{aligned} \textbf{LP4:} \min & \; B_A\gamma + \sum_{i=1}^{m} \delta[i] \\ s.t. & \; c[i]\gamma + \delta[i] \ge f[i], \forall i \end{aligned} \qquad (4)$$

Solving **LP**1 and **LP**3 jointly requires non-linear optimization. We avoid this by presenting an equivalent primal linear program **LP**5 and its dual **LP**6 to compute the Nash equilibrium with **LP**5 optimizing the adversary's strategy and **LP**6 computing the optimal designer's strategy:

$$\begin{aligned} \textbf{LP5:} \min & \; B_A\gamma + \sum_{i=1}^{m} \delta[i] \\ s.t. & \; c[i]\gamma + \delta[i] \ge f[i], \forall i \\ & \sum_{i=1}^{m} f[i] \ge 1 \\ & \sum_{i=1}^{m} d[i]f[i] \le B_D \end{aligned} \qquad (5) \qquad \begin{aligned} \textbf{LP6:} \max & \; X_D - B_D\Omega \\ s.t. & \; X_D - X[i] \le d[i]\Omega, \forall i \\ & \sum_{i=1}^{m} c[i]X[i] \le B_A \\ & X[i] \le 1, \forall i \end{aligned} \qquad (6)$$

The optimum solution of **LP**5, **LP**6 must satisfy the following complementary slackness conditions:

$$\begin{cases} (c[i]\gamma + \delta[i] - f[i])X[i] = 0, \forall i & \cdots(i) \\ (\sum_{i=1}^{m} f[i] - 1)X_D = 0 & \cdots(ii) \\ (\sum_{i=1}^{m} d[i]f[i] - B_D)\Omega = 0 & \cdots(iii) \\ (X_D - X[i] - d[i]\Omega)f[i] = 0, \forall i & \cdots(iv) \\ (\sum_{i=1}^{m} c[i]X[i] - B_A)\gamma = 0 & \cdots(v) \\ (X[i] - 1)\delta[i] = 0, \forall i & \cdots(vi) \end{cases}$$

We present an algorithm to calculate a feasible solution that satisfies all the complementary slackness conditions. We observe that the structure of the Nash equilibrium evolves step by step as the designer budget B_D and opponents budget B_A increases. First assume that the attack budget B_A is very small in

which case the attack on no edge will reach 1. Denote $C(k) = \sum_{i=1}^{k} c[i]$ and $CD(k) = \sum_{i=1}^{k} c[i]d[i], \forall k = 1 \dots m$ for convenience. They can be computed in $O(m)$ steps. While the designer's goal is to even out the bang-per-buck of every edge, note that with the limited B_D she may not be able to do so on the entire cut. If the designer limits attention to only the k cheapest edges, we can calculate the flow on those edges with the same bang-per-buck γ along with the corresponding designer budget $B_D[k]$ by solving the following system of equations.

$$\begin{cases} \sum_{i=1}^{k} f[i] = 1 \\ \sum_{i=1}^{k} d[i]f[i] = B_D[k] \\ f[i] = c[i]\gamma, \forall i = 1 \dots k \end{cases} \implies \begin{cases} \gamma = \dfrac{1}{C(k)} \\ f[i] = \dfrac{c[i]}{C(k)}, \forall i = 1 \dots k \\ B_D[k] = \dfrac{CD(k)}{C(k)} \end{cases} \tag{7}$$

One can verify that $0 \leq f[i] \leq 1, \forall i = 1 \dots k$. Each $B_D[k]$ can be calculated in $O(1)$ steps. If $B_D[k] < B_D < B_D[k+1]$, the designer's flow pattern is a mix of flow patterns corresponding to $B_D[k]$ and $B_D[k+1]$ respectively.

Lemma 1. $B_D[k-1] < B_D[k]$

Since $B_D[k]$ increases monotonically w.r.t. k we get:

Corollary 1. *Let $\hat{k}$ be the largest integer with $B_D[\hat{k}-1] < B_D$. The index $\hat{k}$ can be found in $O(\log m)$ steps.*

The index $\hat{k}$ is very important in our methodology. With a small B_A, the adversary's attack will focus only on the first $\hat{k}-1$ edges. As B_A increases, the designer will have incentive to route flow on the edge with the next cheapest design cost, and the adversary's response also includes one more edge. For $t = \hat{k} \dots m$, we define $B_A[t]$ to be the highest adversary budget where e_t is not attacked. When B_A goes above $B_A[t]$ the designer will start routing on e_{t+1} and the adversary will start attacking e_t. Calculate $B_A[t]$ by solving the following system of equations:

$$\begin{cases} \sum_{i=1}^{t} c[i]X[i] = B_A[t] \\ X[1] = 1 \\ X[t] = 0 \\ X[i] = X_D - d[i]\Omega, \forall i = 1 \dots t \end{cases} \implies \begin{cases} X_D = \dfrac{d[t]}{d[t] - d[1]} \\ \Omega = \dfrac{1}{d[t] - d[1]} \\ X[i] = \dfrac{d[t] - d[i]}{d[t] - d[1]}, \forall i = 2 \dots t \\ B_A[t] = \dfrac{d[t]C(t) - CD(t)}{d[t] - d[1]} \end{cases} \tag{8}$$

One can verify that $0 \leq X[i] \leq 1$, $\forall i = 1 \dots t$. Each $B_A[t]$ can be calculated in $O(1)$ steps.

Lemma 2. $B_A[t-1] < B_A[t]$.

We now construct best responses for both parties. For convenience all the unmentioned variables $X[t+1] \dots X[m]$ take value of 0. Depending on the value of B_A, there are three cases. In each case the design and attack follow a specific pattern. In the first case where $B_A \leq B_A[\hat{k}]$, the designer routes flow on the first $\hat{k}$ edges, the adversary budget is enough to cover only the first $\hat{k}-1$ edges and no edge is attacked fully. In the second case where $B_A[t] < B_A \leq B_A[t+1], \hat{k} \leq t \leq m-1$, the design includes the first t edges, the attack includes the first $t-1$ edges and e_1 is always attacked fully. In the third case where $B_A > B_A[m]$, the flow design and the attack both uses all m edges. The Nash equilibrium in each of the cases can be computed in $O(m)$ steps.

Case(I): $B_A \leq B_A[\hat{k}]$

In this case the design uses the first $\hat{k}$ edges and the attack uses the first $\hat{k}-1$ edges. The attack on e_1 has not reached 1 yet. We first calculate the designer's best response f^* along with the dual variables by solving the following system of equations in $O(m)$ steps.

$$\begin{cases} \sum_{i=1}^{\hat{k}} f^*[i] = 1 \\ \sum_{i=1}^{\hat{k}} f^*[i]d[i] = B_D \\ c[i]\gamma = f^*[i], \forall i = 1 \dots \hat{k}-1 \end{cases} \implies \begin{cases} \gamma = \dfrac{d[\hat{k}] - B_D}{d[\hat{k}] \cdot C(\hat{k}-1) - CD(\hat{k}-1)} \\ f^*[\hat{k}] = \dfrac{B_D \cdot C(\hat{k}-1) - CD(\hat{k}-1)}{d[\hat{k}] \cdot C(\hat{k}-1) - CD(\hat{k}-1)} \\ f^*[i] = \dfrac{c[i](d[\hat{k}] - B_D)}{d[\hat{k}] \cdot C(\hat{k}-1) - CD(\hat{k}-1)}, \\ \quad \forall i = 1 \dots \hat{k}-1 \end{cases}$$

Lemma 3. *f^*, γ are feasible solutions to the linear program.*

Now we calculate the adversary's best response X^* with dual variables X_D, Ω by solving the following system of equations in $O(m)$ steps.

$$\begin{cases} \sum_{i=1}^{\hat{k}} c[i]X^*[i] = B_A \\ X^*[\hat{k}] = 0 \\ X_D - X^*[i] = d[i]\Omega, \forall i = 1 \dots \hat{k} \end{cases} \implies \begin{cases} X_D = \dfrac{d[\hat{k}]B_A}{d[\hat{k}]C(\hat{k}) - CD(\hat{k})} \\ \Omega = \dfrac{B_A}{d[\hat{k}]C(\hat{k}) - CD(\hat{k})} \\ X^*[i] = \dfrac{(d[\hat{k}] - d[i])B_A}{d[\hat{k}]C(\hat{k}) - CD(\hat{k})}, \forall i = 1 \dots \hat{k} \end{cases}$$

Lemma 4. *X^*, X_D, Ω are feasible solutions to the linear program.*

Lemma 5. *The best responses calculated above form a Nash equilibrium.*

Case(II): $B_A[t] < B_A \leq B_A[t+1], \hat{k} \leq t \leq m-1$

In this case the design uses the first t edges and the attack uses the first $t-1$ edges. The attack on e_1 is always 1. Recall that $B_A[t]$ increases monotonically w.r.t. t. Therefore we can search for the value of t in $O(\log m)$ steps. We now

calculate the designer's best response f^* alone with the dual variables by solving the following system of equations in $O(m)$ steps.

$$\begin{cases} \sum_{i=1}^{t} f^*[i] = 1 \\ \sum_{i=1}^{t} f^*[i]d[i] = B_D \\ c[1]\gamma + \delta[1] = f^*[1] \\ c[i]\gamma = f^*[i], \forall i = 2 \dots t \end{cases} \implies \begin{cases} \gamma = \dfrac{B_D - d[1]}{CD(t) - d[1]C(t)} \\ \delta[1] = \dfrac{CD(t) - B_D C(t)}{CD(t) - d[1]C(t)} \\ f^*[1] = \dfrac{c[1](B_D - d[1]) + CD(t) - B_D C(t)}{CD(t) - d[1]C(t)} \\ f^*[i] = \dfrac{c[i](B_D - d[1])}{CD(t) - d[1]C(t)}, \forall i = 2 \dots t \end{cases}$$

Lemma 6. *f^*, δ, γ are feasible solutions to the linear program.*

Now we calculate the adversary's best response X^* with dual variables X_D, Ω by solving the following system of equations in $O(m)$ steps.

$$\begin{cases} \sum_{i=1}^{t} c[i]X^*[i] = B_A \\ X^*[1] = 1 \\ X_D - X^*[i] = d[i]\Omega, \forall i = 1 \dots t \end{cases} \implies \begin{cases} X_D = \dfrac{CD(t) - d[1]B_A}{CD(t) - d[1]C(t)} \\ \Omega = \dfrac{C(t) - B_A}{CD(t) - d[1]C(t)} \\ X^*[i] = \dfrac{CD(t) - d[1]B_A - d[i](C(t) - B_A)}{CD(t) - d[1]C(t)}, \\ \forall i = 1 \dots t \end{cases}$$

Lemma 7. *X^*, X_D, Ω are feasible solutions to the linear program.*

Lemma 8. *The best responses calculated above form a Nash equilibrium.*

Case(III): $B_A > B_A[m]$

In this case both the design and the attack uses all m edges and the attack on e_1 is 1. As B_A increases the attack become closer and closer to a full attack on every edge. We first calculate the designer's best response f^* alone with the dual variables by solving the following system of equations in $O(m)$ steps.

$$\begin{cases} \sum_{i=1}^{m} f^*[i] = 1 \\ \sum_{i=1}^{m} f^*[i]d[i] = B_D \\ c[1]\gamma + \delta[1] = f^*[1] \\ c[i]\gamma = f^*[i], \forall i = 2 \dots m \end{cases} \implies \begin{cases} \gamma = \dfrac{B_D - d[1]}{CD(m) - d[1]C(m)} \\ \delta[1] = \dfrac{CD(m) - B_D \cdot C(m)}{CD(m) - d[1]C(m)} \\ f^*[1] = \dfrac{c[1](B_D - d[1]) + CD(m) - B_D \cdot C(m)}{CD(m) - d[1]C(m)} \\ f^*[i] = \dfrac{c[i](B_D - d[1])}{CD(m) - d[1]C(m)}, \forall i = 2 \dots m \end{cases}$$

Lemma 9. *f^*, δ, γ are feasible solutions to the linear program.*

Now we calculate the adversary's best response X^* with dual variables X_D, Ω by solving the following system of equations in $O(m)$ steps.

$$\begin{cases} \sum_{i=1}^{m} c[i]X^*[i] = B_A \\ X^*[1] = 1 \\ X_D - X^*[i] = d[i]\Omega, \forall i = 1 \dots m \end{cases} \implies \begin{cases} X_D = \dfrac{CD(m) - d[1]B_A}{CD(m) - d[1]C(m)} \\ \Omega = \dfrac{C(m) - B_A}{CD(m) - d[1]C(m)} \\ X^*[i] = \dfrac{CD(m) - d[1]B_A - d[i](C(m) - B_A)}{CD(m) - d[1]C(m)}, \\ \forall i = 1 \dots m \end{cases}$$

Lemma 10. *X^*, X_D, Ω are feasible solutions to the linear program.*

Lemma 11. *The best responses calculated above form a Nash equilibrium.*

To summarize, here is the algorithm to compute the Nash equilibrium of **Bipartite Routing Adversarial Game**.

Algorithm 1. Bipartite Routing Adversarial Game

Input: $N(E, d, c), B_D, B_A$
Output: f^*, X^*

1: Sort d in increasing order if not already sorted
2: Calculate $C(i), CD(i), \forall i = 1 \dots m$
3: Calculate $B_D[i], \forall i = 1 \dots m$ using **Equation(7)**
4: Determine $\hat{k}$ using **Corollary 1**
5: Calculate $B_A[i], \forall i = \hat{k} \dots m$ using **Equation(8)**
6: Determine the case from one of **Case(I),(II),(III)** using B_A and $B_A[i]$ and calculate the best responses f^*, X^*

Theorem 1. *A Nash equilibrium in a **Bipartite Routing Adversarial Game** can be computed in $O(m \log m)$ steps. With pre-sorted designer costs $d[i]$, the Nash equilibrium can be computed in $O(m)$ steps.*

4 Network Routing Adversarial Game

In this section we look at the **Network Routing Adversarial Game**. Given a network $N(G, s, t, c)$, let E be the edge set and $\mathcal{P}$ be the set of all $s-t$ paths. Each edge e_i is associated with an adversary cost $c[i]$ and the adversary has a budget B. Note that in this version of game there is no budget constraint for the designer.

In order to solve this zero-sum game we establish linear programs whose solution provides the routing and attack strategy. We note that this is a non-atomic routing game, not a mixed equilibrium game, though it may be interpreted as one.

Since the number of possible paths in the network is exponential, we would attempt to utilize combinatorial methods. We first consider the adversary's strategy. Let the designer's flow pattern be f; then the adversary solves the following program with its dual:

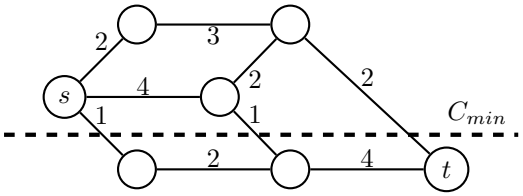
$$\begin{aligned}
\textbf{LP9:}\ \max\ & \sum_{j=1}^{k} f[j] \cdot X_{\mathcal{P}}[j] \\
s.t.\ & \sum_{i=1}^{m} c[i] \cdot X[i] \le B \\
& X_{\mathcal{P}}[j] \le \sum_{i \mid e_i \in p_j} X[i], \forall j \\
& 0 \le X[i] \le 1, \forall i \\
& 0 \le X_{\mathcal{P}}[j] \le 1, \forall j
\end{aligned} \qquad (9)$$

$$\begin{aligned}
\textbf{LP10:}\ \min\ & \sum_{j=1}^{k} \alpha[j] + B \cdot \gamma + \sum_{i=1}^{m} \delta[i] \\
s.t.\ & \alpha[j] + \beta[j] \ge f[j], \forall j \\
& \delta[i] + c[i] \cdot \gamma \ge \sum_{j \mid p_j \ni e_i} \beta[j], \forall i
\end{aligned} \qquad (10)$$

The optimum solution of **LP**9, **LP**10 must satisfy the following complementary slackness conditions:

$$\begin{cases} (X_{\mathcal{P}}[j]-1)\alpha[j]=0, \forall j & \cdots(i) \\ (X_{\mathcal{P}}[j]-\sum_{i|e_i\in p_j} X[i])\beta[j]=0, \forall j & \cdots(ii) \\ (\sum_{i=1}^{m} c[i]\cdot X[i]-B)\gamma=0 & \cdots(iii) \\ (X[i]-1)\delta[i]=0, \forall i & \cdots(iv) \\ (\alpha[j]+\beta[j]-f[j])X_{\mathcal{P}}[j]=0, \forall j & \cdots(v) \\ (\delta[i]+c[i]\cdot\gamma-\sum_{j|p_j\ni e_i}\beta[j])X[i]=0, \forall i & \cdots(vi) \end{cases}$$

We interpret γ as $f_e[i]/c[i]$, i.e. the bang-per-buck of all the edges attacked. Condition (vi) implies that the adversary will only attack edges with the highest γ. Let C_{min} be the min-cost $s-t$ cut and c^* be its cost. It's natural to attack C_{min}, as every possible $s-t$ path goes through at least 1 edge in C_{min}. Since the flow amount on C_{min} is at least 1, the highest γ is at least $1/c^*$. In case there are multiple min-cuts, the adversary may choose any one of them to achieve the same attack level on any possible paths, and different choices of min-cut will result in different optimal adversary strategies and different Nash equilibria, though the final utility will be the same. In the following example where the numbers denote the cost of edges, attacking edges in C_{min} yields the highest $\gamma = 1/4$.



Without loss of generality, we assume $B < c^*$, for otherwise the attack may simply cover every edge in C_{min}. It is possible to attack edges in C_{min} to obtain the attack on paths $X_{\mathcal{P}}$ that forces a rational designer to design the flow satisfying the following **Property A1**:

$$\textbf{Property A1}: \begin{cases} X_{\mathcal{P}}[j]=\chi, \forall j \text{ s.t. } f[j]>0 \\ X_{\mathcal{P}}[j]\geq\chi, \forall j \end{cases}, \text{ where } \chi \text{ is a constant } \in [0,1]$$

With this attack pattern, the designer does not need to deviate from her current routing.

Lemma 12. *Let f be the designer's flow strategy. Given an attack X, that defines $X_{\mathcal{P}}$, and which satisfies* **Property A1** *w.r.t f, the designer's flow strategy f is stable w.r.t. X, i.e. the designer has no incentive to change the flow design f.*

Proof. The designer's current loss is χ since every path that carries flow is attacked by χ. Given any other flow pattern f', the loss is $\sum_j X_{\mathcal{P}}[j]f'[j] \geq \sum_j \chi \cdot f'[j] = \chi$.

In other words, the current flow design uses only paths from the designer's support set based on the attack, no other flow pattern will reduce the loss.

The following lemma describes a flow design that satisfies specific properties on C_{min}.

Lemma 13. *Given any graph $G(V, E)$, the designer is able to route a flow pattern f^* in $O(nm)$ steps, satisfying the following* **Property A2**, *where c^* is the cost of C_{min}.*

$$\textbf{\textit{Property A2}}: \begin{cases} \forall e_i \in E, & f_e^*[i] \leq c[i]/c^* \\ \forall e_i \in C_{min}, & f_e^*[i] = c[i]/c^* \end{cases}$$

Proof. We create a new flow network as follows. Use the same vertex set V and for each undirected edge $e = (u, v) \in E$, create directed edges (u, v) and (v, u), both with capacity c_e. Now we compute a maximum $s - t$ flow denoted by f^* in $O(nm)$ steps [13]. By the max-flow min-cut theorem, the flow amount is c^* since C_{min} is the min-cost $s - t$ cut. This flow satisfies the following:

$$\begin{cases} \forall e_i \in E, & f_e^*[i] \leq c_e \\ \forall e_i \in C_{min}, & f_e^*[i] = c_e \end{cases}$$

Now after scaling the flow amount on each edge by $1/c^*$, we get an $s - t$ flow amounting to 1, satisfying **Property A2**. Note that such flow design is not unique, so technically the optimal designer strategy and the Nash Equilibrium is not unique, but they all contribute to the same final utility. In case there are mulitple min-cuts, the way of flow construction also makes sure of the optimality no matter which min-cut the adversary chooses to attack.

We observe that there is no edge in C_{min} carrying any flow from T to S, since f^* is a scaled max flow. We decompose the flow pattern into a set of paths $\mathcal{P}'$, $|\mathcal{P}'| \leq m$, such that $\forall p_j \in \mathcal{P}'$, $f^*[j] > 0$. The decomposition can be computed in $O(nm)$ steps. We ignore any path $p_j \notin \mathcal{P}'$ as it carries no flow, therefore won't be considered by the adversary. This allows the adversary to focus on a reduced set of paths $\mathcal{P}'$ that doesn't have exponential size.

Given the cut C_{min}, define an attack X_{min} of full coverage on C_{min} to be

$$\begin{cases} X_{min}[i] = 1, \ \forall i | e_i \in C_{min} \\ X_{min}[i] = 0, \ \forall i | e_i \notin C_{min} \end{cases}$$

That is, every edge in the cut is fully attacked. Note that since $B < c^*$ this attack is not within the budget.

The following lemma describes the attack at Nash equilibrium.

Lemma 14. *Let f^* be a flow design satisfying* ***Property A2****. Then the best response of the adversary is as follows*

$$\begin{cases} X^* = (B/c^*) \cdot X_{min} \\ X^*_{\mathcal{P}}[j] = B/c^*, \ \forall j | p_j \in \mathcal{P}' \end{cases}$$

Proof. The response specified in the lemma only attacks edges from C_{min} with the same level. $X^*_{\mathcal{P}}$ is calculated correctly because every path in $\mathcal{P}'$ goes through only one edge in C_{min}. We show the optimality of the attack via complementary slackness.

We provide the following solution:

$$\begin{cases} \delta[i] = 0, \forall e_i \\ \alpha[j] = 0, \forall p_j \\ \beta[j] = f^*[j], \forall p_j \\ \gamma = 1/c^* \end{cases}$$

Complementary slackness conditions $(i)(ii)(iv)(v)$ are clearly satisfied.

Condition (iii) is satisfied since $\sum_i X^*[i]c[i] = \sum_{i|e_i \in C_{min}} X^*[i]c[i] = \frac{B}{c^*} \cdot \sum_{i|e_i \in C_{min}} c[i] = \frac{B}{c^*} \cdot c^* = B$.

Condition (vi) is satisfied since

1. $\forall i | e_i \notin C_{min}$, $X^*[i] = 0$.
2. $\forall i | e_i \in C_{min}$, $\delta[i] + c[i] \cdot \gamma - \sum_{j|p_j \ni e_i} \beta[j] = c[i] \cdot \gamma - \sum_{j|p_j \ni e_i} f[j] = c[i] \cdot \gamma - f_{e_i} = \frac{c[i]}{c^*} - \frac{c[i]}{c^*} = 0$.

Therefore given the flow pattern, the proposed attack is optimum for the adversary.

Theorem 2. *f^*, the design from* ***Lemma 13****, and X^*, the attack from* ***Lemma 14****, form a Nash equilibrium for the* ***Network Routing Adversarial Game****. The Nash equilibrium can be computed in $O(nm)$ steps.*

5 r-Commodity Network Routing Adversarial Game

In this section we consider the game where there are r designers and 1 adversary. Recall the problem definition from the section on notations. The adversary attacks edges to capture part of the flow from all designers in the similar fashion.

Given the flow pattern f from the designers, the adversary solves the following program with its dual:

$$\textbf{LP11:}\ \max \sum_{j=1}^{k} f[j] \cdot X_{\mathcal{P}}[j] \quad s.t.\ \sum_{i=1}^{m} c[i] \cdot X[i] \le B,\ X_{\mathcal{P}}[j] \le \sum_{i|e_i \in p_j} X[i], \forall j,\ 0 \le X \le 1, \forall i,\ 0 \le X_{\mathcal{P}} \le 1, \forall j \tag{11}$$

$$\textbf{LP12:}\ \min \sum_{j=1}^{k} \alpha[j] + B \cdot \gamma + \sum_{i=1}^{m} \delta[i] \quad s.t.\ \alpha[j] + \beta[j] \ge f[j], \forall j,\ \delta[i] + c[i] \cdot \gamma \ge \sum_{j|p_j \ni e_i} \beta[j], \forall i \tag{12}$$

The optimum solution of **LP11**, **LP12** must satisfy the following complementary slackness conditions:

$$\begin{cases} (X_{\mathcal{P}}[j] - 1)\alpha[j] = 0, \forall j & \cdots(i) \\ (X_{\mathcal{P}}[j] - \sum_{i|e_i \in p_j} X[i])\beta[j] = 0, \forall j & \cdots(ii) \\ (\sum_{i=1}^{m} c[i]X[i] - B)\gamma = 0 & \cdots(iii) \\ (X[i] - 1)\delta[i] = 0, \forall i & \cdots(iv) \\ (\alpha[j] + \beta[j] - f[j])X_{\mathcal{P}}[j] = 0, \forall j & \cdots(v) \\ (\delta[i] + c[i]\gamma - \sum_{j|p_j \ni e_i} \beta[j])X[i] = 0, \forall i & \cdots(vi) \end{cases}$$

While we can solve the case of r-commodities using linear programming, the time complexity is high. We focus on the case when $r = 2$, where combinatorial algorithms exist. By using the costs as capacities of the edges and route max-flow with corresponding source and sink, we can identify min-cost cuts C_1, C_2, and C_{12} which is a set of edges with minimum cost whose removal disconnects all $s_1 - t_1$ and $s_2 - t_2$ paths. Let c_1, c_2, c_{12} denote the costs of the cuts respectively. We ignore the case where the min-cut is not unique, as we discussed that such case doesn't affect the final utility in the previous section. We define attacks with full coverage of these cuts X_1, X_2, X_{12} in the same way in the previous section. We state below Hu's **Max Bi-flows Min-cut Theorem** [9] which shows feasibility of routing in the following subsections.

Theorem 3. *It is feasible to route a 2-commodity flow with requirement* (r_1, r_2) ***iff***

$$\begin{cases} r_1 \le c_1 \\ r_2 \le c_2 \\ r_1 + r_2 \le c_{12} \end{cases}$$

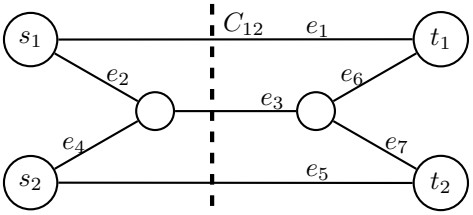
Without loss of generality, we assume $c_1 \le c_2$. Since $c_{12} \le c_1 + c_2$, we have $\frac{c_{12}}{2} \le c_2$. We further assume $B \le c_{12}$, for otherwise the attack may simply use X_{12} as her attack to capture the entire flow. Since the min-cost cuts carry different costs, the adversary attempts to capture the most beneficial one of the

them then turn to the rest. Based on the relationship between c_1 and c_{12} we divide the problem into several subcases.

Example: We use the same flow network with different cost on edges for the illustrative example in each following subsection. For simplicity we assume the cost of e_6, e_7 is $+\infty$; they will never be attacked. Note that in this network the direction of flow is obvious. The paths are $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$, where $p_1 = (e_1)$, $p_2 = (e_2 \to e_3 \to e_6)$, $p_3 = (e_4 \to e_3 \to e_7)$, $p_4 = (e_5)$. The Nash equilibria are computed by linear programming.

Case(I) $\frac{c_{12}}{2} \leq c_1 \leq c_2$

In this case attacking C_{12} is more beneficial than attacking C_1 or C_2. Following is an example of equilibrium where $B = 5, c_1 = 5, c_2 = 5, c_{12} = 8$. At the equilibrium, the adversary only attacks $C_{12} = \{e_1, e_3, e_5\}$ which covers boths designers' flow as C_{12} has better bang-per-buck than C_1 and C_2.



edge	e_1	e_2	e_3	e_4	e_5
$c[i]$	2	3	4	3	2
$f_e[i]$	1/2	1/2	1	1/2	1/2
$X[i]$	5/8	0	5/8	0	5/8

path	p_1	p_2	p_3	p_4
$f[j]$	1/2	1/2	1/2	1/2
$X_{\mathcal{P}}[j]$	5/8	5/8	5/8	5/8

The following lemma describes a 2-commodity flow design that satisfies specific properties on C_{12}.

Lemma 15. *In **Case(I)** the designers are able to route a 2-commodity flow pattern f^* with requirement $(1, 1)$ in $O(n^3)$ steps, satisfying the following **PROPERTY A3***

$$\textbf{\textit{Property A3}}: \begin{cases} \forall e_i \in E, & f_e^*[i] \leq 2c[i]/c_{12} \\ \forall e_i \in C_{12}, & f_e^*[i] = 2c[i]/c_{12} \end{cases}$$

We decompose the flow pattern into a set of paths $\mathcal{P}'_1, \mathcal{P}'_2$ with $|\mathcal{P}'_1| \leq m$, $|\mathcal{P}'_2| \leq m$. Let $\mathcal{P}' = \mathcal{P}'_1 \cup \mathcal{P}'_2$. The decomposition can be computed in $O(nm)$ steps. The adversary will focus the attack on $\mathcal{P}'$.

The following lemma describes the attack at Nash equilibrium.

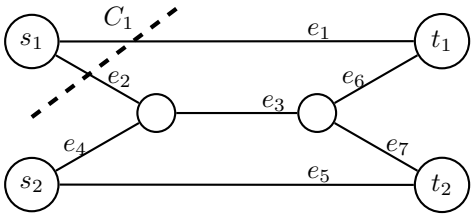
Lemma 16. *Let f^* be a flow design satisfying **PROPERTY A3**, the best response of the adversary is as follows:*

$$\begin{cases} X^* = (B/c_{12}) \cdot X_{12} \\ X^*_{\mathcal{P}}[j] = B/c_{12}, \ \forall j | p_j \in \mathcal{P}' \end{cases}$$

Lemma 17. *When* $\frac{c_{12}}{2} \leq c_1 \leq c_2$, f^* *and attack* X^*, *defined in* ***Lemma 16****, describe a Nash equilibrium solution for the 2-Designer Network Routing Game computable in* $O(n^3)$ *steps.*

Case(II) $c_1 < \frac{c_{12}}{2} \leq c_2$ and $B \leq c_1$

In this case the adversary attacks only C_1 since there is not enough budget to cover C_1 entirely and move onto C_{12}. Following is an example of equilibrium where $B = 4, c_1 = 5, c_2 = 8, c_{12} = 11$. At the equilibrium, the adversary only attacks $C_1 = \{e_1, e_2\}$ which only covers Designer 1's flow as C_1 has better bang-per-buck than C_{12} and C_2.



edge	e_1	e_2	e_3	e_4	e_5
$c[i]$	2	3	4	3	5
$f_e[i]$	1/2	1/2	2/3	1/6	5/6
$X[i]$	4/5	4/5	0	0	0

path	p_1	p_2	p_3	p_4
$f[j]$	1/2	1/2	1/6	5/6
$X_{\mathcal{P}}[j]$	4/5	4/5	0	0

The following lemma describes the 2-commodity flow design that satisfies specific properties on C_1.

Lemma 18. *In* ***Case(II)*** *the designers are able to route a 2-commodity flow pattern* f^* *with requirement* $(1, 1)$ *in* $O(n^3)$ *steps, satisfying the following* ***PROPERTY A4***

$$\textbf{\textit{Property A4}}: \begin{cases} \forall e_i \in E, & f_e^*[i] \leq c[i]/c_1 \\ \forall e_i \in C_1, & f_e^*[i] = c[i]/c_1 \end{cases}$$

We decompose the flow pattern into a set of paths $\mathcal{P}'_1, \mathcal{P}'_2$ with $|\mathcal{P}'_1| \leq m$, $|\mathcal{P}'_2| \leq m$. Let $\mathcal{P}' = \mathcal{P}'_1 \cup \mathcal{P}'_2$. The decomposition can be computed in $O(nm)$ steps. The adversary will focus the attack on $\mathcal{P}'$.

The following lemma describes the attack at Nash equilibrium.

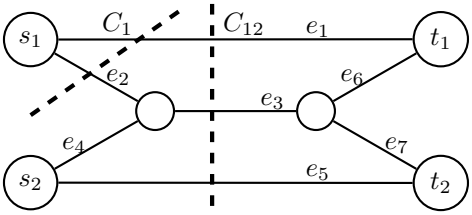
Lemma 19. *Let* f^* *be a flow design satisfying* ***PROPERTY A4****, the best response of the adversary is as follows*

$$\begin{cases} X^* = (B/c_1) \cdot X_1 \\ X^*_{\mathcal{P}'}[j] = B/c_1, \ \forall j | p_j \in \mathcal{P}'_1 \\ X^*_{\mathcal{P}'}[j] = 0, \ \forall j | p_j \in \mathcal{P}'_2 \end{cases}$$

Lemma 20. *When* $(c_1 < \frac{c_{12}}{2} \leq c_2) \& (B \leq c_1)$, f^* *and attack* X^*, *defined in* ***Lemma 19****, provide a Nash equilibrium solution for the 2-Designer Network Routing Game computable in* $O(n^3)$ *steps.*

Case(III) $c_1 < \frac{c_{12}}{2} \leq c_2$ and $c_1 < B \leq c_{12}$

In this case the adversary's attack is a combination of X_1 and X_{12} to have C_1 fully attacked and C_{12} partially attacked. Following is an example of equilibrium where $B = 7, c_1 = 5, c_2 = 8, c_{12} = 11$. Note that the attack comprises of $C_1 = \{e_1, e_2\}$ and $C_{12} = \{e_1, e_3, e_5\}$. After capturing the most beneficial cut C_1, the adversary has budget left so she moves on the attack C_{12} while maintaining the attack on C_1 to still be full. Therefore the adversary's benefit is 1.5 which includes 1 from Designer 1 and 0.5 from Designer 2.



edge	e_1	e_2	e_3	e_4	e_5
$c[i]$	2	3	4	3	5
$f_e[i]$	1/2	1/2	2/3	1/6	5/6
$X[i]$	1	1/2	1/2	0	1/2

path	p_1	p_2	p_3	p_4
$f[j]$	1/2	1/2	1/6	5/6
$X_{\mathcal{P}}[j]$	1	1	1/2	1/2

The flow design f^* in this case is decomposed into 2 flow routing components, f^1 and f^2, i.e. $f^* = f^1 + f^2$. The following lemma describes f^2.

Lemma 21. *In **Case(III)** the designers are able to route a 2-commodity flow pattern f^2 with requirement $(\frac{c_1}{c_{12}-c_1}, 1)$ in $O(n^3)$ steps, satisfying the following **PROPERTY A5***

$$\textbf{\textit{Property A5}}: \begin{cases} \forall e_i \in E, & f_e^2[i] \leq c[i]/(c_{12} - c_1) \\ \forall e_i \in C_1 \cup C_{12}, & f_e^2[i] = c[i]/(c_{12} - c_1) \end{cases}$$

Proof. Use the costs as capacities of edges. Since $c_1 \leq \frac{c_{12}}{2}$, by **Theorem** 3 it is possible to compute a feasible 2-commodity flow f^2 with requirement $(c_1, c_{12}-c_1)$ in $O(n^3)$ steps. This flow satisfies the following:

$$\begin{cases} \forall e_i \in E, & f_e^2[i] \leq c[i] \\ \forall e_i \in C_1 \cup C_{12}, & f_e^2[i] = c[i] \end{cases}$$

Scale f^* down by $\frac{1}{c_{12}-c_1}$, and it satisfies **PROPERTY A5** with requirement $(\frac{c_1}{c_{12}-c_1}, 1)$.

We decompose the flow pattern into sets of paths $\mathcal{P}'_1, \mathcal{P}'_2$ for Designer 1 and 2 respectively, with $|\mathcal{P}'_1| \leq m$, $|\mathcal{P}'_2| \leq m$. Let $\mathcal{P}' = \mathcal{P}'_1 \cup \mathcal{P}'_2$. The decomposition can be computed in $O(nm)$ steps. The adversary will focus the attack on $\mathcal{P}'$.

Lemma 22. *No path in $\mathcal{P}'_2$ goes through C_1.*

Proof. Consider the max 2-commodity flow f^2 we obtained in **Lemma** 21, before scaling. C_1 is a min cut for commodity 1 with capacity c_1, while the total flow amount of commodity 1 is also c_1. Therefore $\forall e \in C_1$, f_e^2 contains only commodity 1. Since $\mathcal{P}'_2$ contains paths with positive flow of commodity 2, no path in $\mathcal{P}'_2$ goes through C_1, *i.e.* $\forall p \in \mathcal{P}'_2$, $p \cap C_1 = \emptyset$.

Let p_l be one arbitrary path in $\mathcal{P}'_1$, we define flow f^1 as follows:

$$f^1 = \begin{cases} f^1[l] = \dfrac{c_{12} - 2c_1}{c_{12} - c_1} \\ f^1[j] = 0, \ \forall j \neq l \end{cases}$$

Note that f^1 contains only commodity 1, its flow amount is $(\frac{c_{12}-2c_1}{c_{12}-c_1}, 0)$.

Let $f^* = f^1 + f^2$ be a 2-commodity flow, it satisfies the requirement $(\frac{c_{12}-2c_1}{c_{12}-c_1}, 0) + (\frac{c_1}{c_{12}-c_1}, 1) = (1, 1)$. Let $\theta = \frac{B-c_1}{c_{12}-c_1}$.

Lemma 23. *Given the flow design f^*, the best response of the adversary is as follows*

$$\begin{cases} X^* = (1-\theta) \cdot X_1 + \theta \cdot X_{12} \\ X^*_{\mathcal{P}'}[j] = 1, \ \forall j | p_j \in \mathcal{P}'_1 \\ X^*_{\mathcal{P}'}[j] = \theta, \ \forall j | p_j \in \mathcal{P}'_2 \end{cases}$$

Proof. We construct the following dual variables of **LP**12

$$\begin{cases} \delta[i] = 0, \forall e_i \\ \alpha[j] = f^1[j], \forall j \\ \beta[j] = f^2[j], \forall j \\ \gamma = \dfrac{1}{c_{12} - c_1} \end{cases}$$

We proceed to show that all complementary slackness conditions of **LP**11, **LP**12 are satisfied.

Condition (i) is satisfied since

1. $\forall p_j \in \mathcal{P}'_1, X_{\mathcal{P}'}[j] = 1$.
2. $\forall p_j \in \mathcal{P}'_2, \alpha[j] = 0$.

Condition (ii) is satisfied since

1. $\forall p_j \in \mathcal{P}'_1$,

$$X_{\mathcal{P}'}[j] - \sum_{i|e_i \in p_j} X[i] = 1 - \sum_{i|e_i \in p_j} X[i] = 1 - \Big(\sum_{i|e_i \in (p_j \cap C_1)} X[i] + \sum_{i|e_i \in (p_j \cap C_{12})} X[i] \Big)$$
$$= 1 - ((1-\theta) + \theta) = 0$$

2. $\forall p_j \in \mathcal{P}'_2$,

$$X_{\mathcal{P}'}[j] - \sum_{i|e_i \in p_j} X[i] = \theta - \sum_{i|e_i \in p_j} X[i] = \theta - \Big(\sum_{i|e_i \in (p_j \cap C_1)} X[i] + \sum_{i|e_i \in (p_j \cap C_{12})} X[i] \Big)$$
$$= \theta - \Big(\sum_{i|e_i \in \emptyset} X[i] + \sum_{i|e_i \in (p_j \cap C_{12})} X[i] \Big) = \theta - (0 + \theta) = 0$$

Condition (iii) is satisfied since

$$\sum_i X[i] - B = (\sum_{i|e_i \in X_1} c[i]X[i] + \sum_{i|e_i \in X_{12}} c[i]X[i]) - B = ((1-\theta) \cdot \sum_{i|e_i \in X_1} c[i] + \theta \cdot \sum_{i|e_i \in X_{12}} c[i]) - B$$
$$= ((1-\theta) \cdot c_1 + \theta \cdot c_{12}) - B = c_1 + \theta(c_{12} - c_1) - B = 0$$

Condition (iv) is satisfied since $\delta[i] = 0, \forall i$.
Condition (v) is satisfied since $\alpha[j] + \beta[j] - f^*[j] = f^1[j] + f^2[j] - (f^1[j] + f^2[j]) = 0, \forall j$.

Condition (vi) is satisfied since

1. $\forall e_i \notin (C_1 \cup C_{12})$, $X[i] = 0$.
2. $\forall e_i \in (C_1 \cup C_{12})$, $c[i]\gamma - \sum_{j|p_j \ni e_i} \beta[j] = c[i]\gamma - f_e^2 = c[i]\gamma - \frac{c[i]}{c_{12}-c_1} = c[i] \cdot \frac{1}{c_{12}-c_1} - \frac{c[i]}{c_{12}-c_1} = 0$

Since all the conditions are satisfied, **LP**11, **LP**12 are at optimum with the given variables. The adversary is not incentivized to deviate.

Lemma 24. *When* $(c_1 < \frac{c_{12}}{2} \leq c_2) \& (c_1 < B \leq c_{12})$, f^* *and attack* X^*, *defined in* ***Lemma*** *23, describe a Nash equilibrium solution for the 2-Designer Network Routing Game that can be computed in* $O(n^3)$ *steps.*

Proof. Since C_1 and C_{12} are cuts of commodity 1, every possible path of Designer 1 goes through both C_1 and C_{12} at least once, thus $\forall p_j \in \mathcal{P}_1, X_{\mathcal{P}}[j] \geq (1-\theta)+\theta = 1$. By the decomposition of f^2 and construction of f^1, $\forall p_j \in \mathcal{P}_1 | f[j] > 0, X_{\mathcal{P}}[j] = (1-\theta)+\theta = 1$. Therefore for Designer 1, the attack satisfies **PROPERTY A1** with $\chi = 1$.
Every possible path of Designer 2 goes through C_{12} at least once, therefore the attack satisfies
PROPERTY A1 with $\chi = \theta$ for Designer 2.
By **Lemma** 12, neither of the designers wants to move from the current flow design.
Since neither the designers nor the adversary wants to mover from the current strategy, this is a Nash equilibrium.

Combining the results of **Lemmas** 17, 20, 24, we come to the following conclusion:

Theorem 4. *For any input of the* ***2-Commodity Network Routing Adversarial Game****, there exists a Nash equilibrium computable in* $O(n^3)$ *steps.*

6 Conclusions

In this paper we have illustrated how to determine Nash equilibrium for adversarial routing problems with player-specific costs on the edges. For the case when adversaries and designers are budget constrained and the graph is bipartite, we provide an efficient solution. We also provide efficient polynomial solutions to the general graph network routing adversary games, when the costs on the resources are not player-specific, and when there are 1 and 2 designers routing flow. It would be of interest to extend this to the more general case. This work was partially supported by NSF grant No: 2028274.

A Appendix

A.1 Proofs

Proof of Lemma 1

Proof.

$$B_D[k] - B_D[k-1] = \frac{CD(k)}{C(k)} - \frac{CD(k-1)}{C(k-1)} = \frac{C(k-1)\cdot CD(k) - C(k)\cdot CD(k-1)}{C(k)\cdot C(k-1)}$$
$$= \frac{c[k]}{C(k)\cdot C(k-1)} \cdot (d[k]\cdot C(k-1) - CD(k-1))$$

$$d[k] > d[i], \forall i < k \implies (d[k]\cdot C(k-1) - CD(k-1)) > 0 \implies B_D[k] - B_D[k-1] > 0$$

Proof of Lemma 2

Proof. Denote the attacks used in calculating $B_A[t-1]$ and $B_A[t]$ by X' and X'' respectively.

$$\begin{cases} X'[i] = \dfrac{d[t-1]-d[i]}{d[t-1]-d[1]}, & \forall i = 1 \dots t-1 \\ X''[i] = \dfrac{d[t]-d[i]}{d[t]-d[1]}, & \forall i = 1 \dots t \end{cases} \implies$$

$$X''[i] - X'[i] = \frac{d[t]d[i] + d[t-1]d[1] - d[t-1]d[i] - d[t]d[1]}{(d[t]-d[1])(d[t-1]-d[1])}$$
$$= \frac{(d[i]-d[1])(d[t]-d[t-1])}{(d[t]-d[1])(d[t-1]-d[1])} > 0, \forall i = 2 \dots t-1$$

Therefore

$$\begin{cases} X'[1] = X''[1] = 1 \\ X'[t] = X''[t] = 0 \\ X'[i] < X''[i], \\ \forall i = 2 \dots t-1 \end{cases} \implies B_A[t-1] = \sum_{i=1}^{m} c[i]X'[i] < \sum_{i=1}^{m} c[i]X''[i] = B_A[t]$$

Proof of Lemma 3

Proof.

1. $\gamma \geq 0$
   The numerator:
   $B_D[\hat{k}] = \frac{CD(\hat{k})}{C(\hat{k})} \leq \frac{d[\hat{k}]C(\hat{k})}{C(\hat{k})} = d[\hat{k}]$, $d[\hat{k}] \geq B_D[\hat{k}] \geq B_D \implies d[\hat{k}] - B_D \geq 0$
   The denominator:
   $d[\hat{k}] \geq d[i], \forall i < \hat{k} \implies d[\hat{k}]C(k-1) - CD(\hat{k}-1) > d[\hat{k}]C(k-1) - d[\hat{k}]C(k-1) = 0$

2. $0 \leq f^*[i] \leq 1, \forall i$
   $\forall i = 1 \ldots \hat{k} - 1, f^*[i] \geq 0$, since $f^*[i] = c[i]\gamma$.
   For $f^*[\hat{k}]$, the numerator:
   $B_D > B_D[\hat{k}-1] \implies B_D \cdot C(\hat{k}-1) > \frac{CD(\hat{k}-1)}{C(\hat{k}-1)} C(\hat{k}-1) \implies B_D \cdot C(\hat{k}-1) - CD(\hat{k}-1) > 0$.
   The denominator > 0 is proven above.
   Since $f^*[i] \geq 0$, $\sum_{i=1}^{\hat{k}} f^*[i] = 1$, $f^*[i] \leq 1, \forall i$. Therefore $0 \leq f^*[i] \leq 1, \forall i$.
3. $f^*[i] \leq c[i]\gamma, \forall i$
   $\forall i = 1 \ldots \hat{k} - 1$, $f^*[i] = c[i]\gamma$.
   $c[\hat{k}]\gamma - f^*[\hat{k}] = \frac{c[\hat{k}](d[\hat{k}]-B_D) - B_D \cdot C(\hat{k}-1) + CD(\hat{k}-1)}{c[\hat{k}]S(\hat{k}-1) - SQ(\hat{k}-1)}$
   The numerator:
   $c[\hat{k}](d[\hat{k}] - B_D) - B_D \cdot C(\hat{k}-1) + CD(\hat{k}-1) = CD(\hat{k}) - B_D \cdot C(\hat{k}) \geq CD(\hat{k}) - B_D[\hat{k}] \cdot C(\hat{k}) = CD(\hat{k}) - \frac{CD(\hat{k})}{C(\hat{k})} C(\hat{k}) = 0$.
   The denominator > 0 is proven above.
   Thus $\forall i = 1 \ldots \hat{k}$, $f^*[i] \leq c[i]\gamma$.

Therefore f^*, γ are feasible.

Proof of Lemma 4

Proof.

1. $X_D, \Omega \geq 0$
   The numerators > 0 obviously.
   The denominator:
   $d[\hat{k}] > d[i], \forall i < k \implies d[\hat{k}]C(\hat{k}) - CD(\hat{k}) > d[\hat{k}]C(\hat{k}) - d[\hat{k}]C(\hat{k}) = 0$
2. $0 \leq X^*[i] \leq 1$
   The numerator:
   $d[\hat{k}] > d[i], \forall i < k \implies (d[\hat{k}] - d[i])B_A > 0$
   The denominator > 0 is proven above. Therefore $X^*[i] \geq 0$.
   Since $X^*[1] \geq X^*[i], \forall i = 1 \ldots \hat{k}$, we need only to show $X^*[1] \leq 1$.

$$
\begin{aligned}
X^*[1] &= \frac{d[\hat{k}] - d[1]}{d[\hat{k}]C(\hat{k}) - CD(\hat{k})} B_A \leq \frac{d[\hat{k}] - d[1]}{d[\hat{k}]C(\hat{k}) - CD(\hat{k})} B_A[\hat{k}] \\
&= \frac{d[\hat{k}] - d[1]}{d[\hat{k}]C(\hat{k}) - CD(\hat{k})} \cdot \frac{d[\hat{k}]C(\hat{k}) - CD(\hat{k})}{d[\hat{k}] - dc[1]} = 1
\end{aligned}
$$

   Thus $0 \leq X^*[i] \leq 1, \forall i = 1 \ldots \hat{k}$.

Therefore X^*, X_D, Ω are feasible.

Proof of Lemma 5

Proof. The complementary slackness conditions $(ii)(iii)(v)$ are obviously satisfied due to the equations.

$\forall i = 1 \dots \hat{k} - 1$, $c[i]\gamma = f[i]$, $\forall i = \hat{k} \dots m$, $X^*[i] = 0$. Therefore condition (i) is satisfied.

$\forall i = 1 \dots \hat{k}$, $X_D - X^*[i] = d[i]\Omega$, $\forall i = \hat{k} + 1 \dots m$, $f^*[i] = 0$. Therefore condition (iv) is satisfied.

$\forall i = 1 \dots m$, $\delta[i] = 0$. Therefore condition (vi) is satisfied.

All the complementary slackness conditions are met, the proposed solution is the optimal.

Proof of Lemma 6

Proof.

1. $\gamma \geq 0$
   The numerator: $B_D - d[1] \geq 0$ by assumption.
   The denominator: $CD(t) - d[1]C(t) = \sum_{i=1}^{t} c[i](d[i] - d[1]) > 0$.
2. $\delta[1] \geq 0$
   The denominator> 0 is proven above.
   Recall that $\hat{k}$ is the largest integer with $B_D(\hat{k} - 1) < B_D$. Since $t \geq \hat{k}$, the numerator $CD(t) - B_D C(t) \geq CD(t) - B_D[t]C(t) = CD(t) - \frac{CD(t)}{C(t)}C(t) = 0$.
3. $0 \leq f^* \leq 1$
   Since $\gamma, \delta[1] \geq 0$, $f^* \geq 0$. Since $\sum_{i=1}^{t} f^*[i] = 1$, $f^* \leq 1$.

Therefore $f^*, \gamma, \delta[1]$ are feasible.

Proof of Lemma 7

Proof.

1. $\Omega \geq 0$
   The denominator: $c[i] > c[1], \forall i > 1 \implies CD(t) - d[1]C(t) > 0$.
   The numerator: $C(t) - B_A > C(t) - B_A[t+1] = C(t) - \sum_{i=1}^{t} c[i]X[i] + 0 \cdot X[t+1] > C(t) - \sum_{i=1}^{t} c[i] \cdot 1 = C(t) - C(t) = 0$.
2. $X_D \geq 0$
   $X_D - X^*[1] = d[1]\Omega \implies X_D = 1 + d[1]\Omega \geq 0$.
3. $0 \leq X^*[i] \leq 1, \forall i = 1 \dots t$
   Since $X_D - X^*[i] = d[i]\Omega$, $X^*[1] > X^*[2] > \cdots > X^*[t]$. $X^*[1] = 1 \implies X^*[i] \leq 1$.
   To show $X^*[i] \geq 0$, it suffices to show $X^*[t] \geq 0$.
   The denominator > 0 is proven above.
   The numerator: $CD(t) + (d[t] - d[1])B_A - d[t]C(t) \geq CD(t) + (d[t] - d[1])B_A[t] - d[t]C(t) = CD(t) + (d[t] - d[1])\frac{d[t]C(t) - CD(t)}{d[t] - d[1]} - d[t]C(t) = 0$.
   Thus $0 \leq X^*[i] \leq 1, \forall i = 1 \dots t$.

Therefore X_D, Ω, X^* are feasible.

***Proof of Lemma** 8*

Proof. The complementary slackness conditions $(ii)(iii)(v)$ are obviously satisfied due to the equations.

$\forall i = 1 \dots t$, $c[i]\gamma + \delta[i] = f^*[i]$, $\forall i = t+1 \dots m$, $X^*[i] = 0$. Therefore condition (i) is satisfied.

$\forall i = 1 \dots t$, $X_D - X^*[i] = d[i]\Omega$, $\forall i = t+1 \dots m$, $f^*[i] = 0$. Therefore condition (iv) is satisfied.

$X^*[1] = 1$, $\forall i = 2 \dots m$, $\delta[i] = 0$. Therefore condition (vi) is satisfied.

All the complementary slackness conditions are met, the proposed solution is the optimal.

***Proof of Lemma** 9*

Proof.

1. $\gamma \geq 0$
   The numerator: $B_D - d[1] \geq 0$ by assumption.
   The denominator: $CD(m) - d[1]C(m) = \sum_{i=1}^{m} c[i](d[i] - d[1]) > 0$.
2. $\delta[1] \geq 0$
   The denominator > 0 is proven above.
   The numerator:
   Recall that $\hat{k}$ is the largest integer with $B_D(\hat{k} - 1) < B_D$. Since $m \geq \hat{k}$, the numerator $CD(m) - B_D \cdot C(m) \geq CD(m) - B_D[m]C(m) = CD(m) - \frac{CD(m)}{C(m)}C(m) = 0$.
3. $0 \leq f^* \leq 1$
   Since $\gamma, \delta[1] \geq 0$, $f^* \geq 0$. Since $\sum_{i=1}^{t} f^*[i] = 1$, $f^* \leq 1$.

Therefore $f^*, \gamma, \delta[1]$ are feasible.

***Proof of Lemma** 10*

Proof.

1. $\Omega \geq 0$
   The numerator $S(m) - B_A \geq 0$ by assumption.
   The denominator $CD(m) - d[1]C(m) = \sum_{i=1}^{m} c[i](d[i] - d[1]) > 0$.

2. $X_D \geq 0$
   $X_D - X^*[1] = c[1]\Omega \implies X_D = 1 + c[1]\Omega \geq 0$.
3. $0 \leq X^*[i] \leq 1, \forall i = 1 \dots m$
   Since $X_D - X^*[i] = d[i]\Omega$, $X^*[1] > X^*[2] > \dots > X^*[m]$. $X^*[1] = 1 \implies X^*[i] \leq 1$.
   To show $X^*[i] \geq 0$, it suffices to show $X^*[m] \geq 0$.

The denominator > 0 is proven above.
The numerator:

$$CD(m) + (d[m] - d[1])B_A - d[m]C(m) \geq CD(m) + (d[m] - d[1])B_A[m] - d[m]C(m)$$
$$= CD(m) + (d[m] - d[1])\frac{d[m]C(m) - CD(m)}{d[m] - d[1]} - d[m]C(m) = 0$$

Thus $0 \leq X^*[i] \leq 1, \forall i = 1 \ldots m$.

Therefore X_D, Ω, X^* are feasible.

Proof of Lemma 11

Proof. The complementary slackness conditions $(ii)(iii)(v)$ are obviously satisfied due to the equations.

$\forall i = 1 \ldots m$, $c[i]\gamma + \delta[i] = f^*[i]$. Therefore condition (i) is satisfied.

$\forall i = 1 \ldots m$, $X_D - X^*[i] = d[i]\Omega$. Therefore condition (iv) is satisfied.

$X^*[1] = 1$, $\forall i = 2 \ldots m$, $\delta[i] = 0$. Therefore condition (vi) is satisfied.

All the complementary slackness conditions are met, the proposed solution is the optimal.

Proof of Theorem 2

Proof. Since every possible path in the network has to go through C_{min} at least once, $X_{\mathcal{P}}[j] \geq B/c^*$, $\forall j$. The attack above satisfies **Property A1** with $\chi = B/c^*$, therefore the adversary's gain is B/c^*, and the designer doesn't want to change the flow design and use any path $p_j \notin \mathcal{P}'$.

Since neither the designer nor the adversary wants to move from the current strategy, this is a Nash equilibrium.

Proof of Lemma 15

Proof. Use the costs as capacities of edges. Since $\frac{c_{12}}{2} \leq c_1 \leq c_2$, by **Theorem** 3 it is possible to compute a feasible 2-commodity flow f^* with requirements $(\frac{c_{12}}{2}, \frac{c_{12}}{2})$ in $O(n^3)$ steps [10]. This flow satisfies the following:

$$\begin{cases} \forall e_i \in E, & f_e^*[i] \leq c[i] \\ \forall e_i \in C_{12}, & f_e^*[i] = c[i] \end{cases}$$

Scale f^* down by $\frac{2}{c_{12}}$, and it satisfies **PROPERTY A3** and each designer's flow sums up to be 1.

Proof of Lemma 16

Proof. We construct the following dual variables feasible for **LP**12, satisfying complementary slackness conditions.

$$\begin{cases} \delta[i] = 0, \forall e_i \\ \alpha[j] = 0, \forall p_j \\ \beta[j] = f^*[j], \forall p_j \\ \gamma = 2/c_{12} \end{cases}$$

Since the adversary's program does not distinguish the 2 designers, by substituting C^* in the proof by C_{12}, the proof becomes identical to that of **Lemma** 14.

Proof of Lemma 17

Proof. Every possible path of each designer goes through C_{12} at least once, therefore this attack satisfies **PROPERTY A1** with $\chi = \frac{B}{c_{12}}$. By **Lemma** 12 the designers don't want to change the flow designs. The adversary's gain is $\frac{2B}{c_{12}}$.

Since neither the designers nor the adversary wants to move from the current strategy, this is a Nash equilibrium.

Proof of Lemma 18

Proof. Use the costs as capacities of edges. Since $c_1 \leq \frac{c_{12}}{2}$, by **Theorem** 3 it is possible to compute a feasible 2-commodity flow f^* with requirements (c_1, c_1) in $O(n^3)$ steps. This flow satisfies the following:

$$\begin{cases} \forall e_i \in E, & f_e^*[i] \leq c[i] \\ \forall e_i \in C_1, & f_e^*[i] = c[i] \end{cases}$$

Scale f^* down by $\frac{1}{c_1}$, and it satisfies **PROPERTY A4** and each designer's flow sums up to be 1.

Proof of Lemma 19

Proof. We construct the following dual variables of **LP**12

$$\begin{cases} \delta[i] = 0, \forall e_i \\ \alpha[j] = 0, \forall p_j \\ \beta[j] = f^*[j], \forall p_j \\ \gamma = 1/c_1 \end{cases}$$

We proceed to show that all complementary slackness conditions of **LP**11, **LP**12 are satisfied.

Condition (i)(iv)(v) are clearly satisfied.

Condition (ii) is satisfied since

1. $\forall p_j \in \mathcal{P}'_1$, since every path goes through C_1 only once, $\sum_{i|e_i \in p_j} X[i] = B/c_1 = X_{\mathcal{P}}[j]$.
2. $\forall p_j \in \mathcal{P}'_2$, since no edge or path of Designer 2 is attacked, $\sum_{i|e_i \in p_j} X[i] = X_{\mathcal{P}}[j] = 0$.

Condition (iii) is satisfied since

$$\sum_{i=1}^{m} c[i]X[i] = \sum_{i|e_i \in C_1} c[i]X[i] = B/c_1 \cdot \sum_{i|e_i \in C_1} c[i] = B/c_1 \cdot c_1 = B$$

Condition (vi) is satisfied since

1. $\forall e_i \notin C_1$, $X[i] = 0$.
2. $\forall e_i \in C_1$,

$$\delta[i] + c[i]\gamma - \sum_{j|p_j \ni e_i} \beta[j] = c[i]\gamma - \sum_{j|p_j \ni e_i} \beta[j] = c[i]\gamma - f_e^* = c[i]\gamma - \frac{c[i]}{c_1} = 0$$

Since all the conditions are satisfied, **LP**11, **LP**12 are at optimum with the given variables. The adversary is not incentivized to deviate.

Proof of Lemma 20

Proof. No path in $\mathcal{P}_2$ is attacked, so Designer 2 doesn't want to deviate from the current flow design. Every possible path of Designer 1 goes through C_1 at least once, therefore the attack satisfies **PROPERTY A1** with $\chi = \frac{B}{c_1}$ for Designer 1. The adversary's gain is $\frac{B}{c_1}$, and the designers don't want to change the flow designs.

Since neither the designers nor the adversary wants to change from the current strategy, this is a Nash equilibrium.

References

1. Ahmadinejad, A., Dehghani, S., Hajiaghayi, M., Lucier, B., Mahini, H., Seddighin, S.: From duels to battlefields: computing equilibria of blotto and other games. Math. Oper. Res. **44**, 1304–1325 (2019)
2. Barahona, F.: Packing spanning trees. Math. Oper. Res. **20**(1), 104–115 (1995)
3. Calinescu, G., Kapoor, S., Quinn, M., Shin, J.: Adversary games in secure/reliable network routing. In: Jain, R., Kannan, R. (eds.) GameNets 2011. LNICST, vol. 75, pp. 249–264. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30373-9_18
4. Chang, S.L., Piraveenan, M., Pattison, P., Prokopenko, M.: Game theoretic modelling of infectious disease dynamics and intervention methods: a review. J. Biol. Dyn. **14**(1), 57–89 (2020)
5. Chen, X., Deng, X., Teng, S.H.: Settling the complexity of computing two-player Nash equilibria. J. ACM (JACM) **56**(3), 14 (2009)
6. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**(2), 169–197 (1981)

7. Guan, P., He, M., Zhuang, J., Hora, S.C.: Modeling a multitarget attacker-defender game with budget constraints. Decis. Anal. **14**(2), 87–107 (2017)
8. Gueye, A., Walrand, J.C., Anantharam, V.: A network topology design game: How to choose communication links in an adversarial environment. In: Proc. of the 2nd International ICST Conference on Game Theory for Networks, GameNets, vol. 11, p. 5 (2011)
9. Hu, T.C.: Multi-commodity network flows. Oper. Res. **11**(3), 344–360 (1963)
10. Itai, A.: Two-commodity flow. J. ACM (JACM) **25**(4), 596–611 (1978)
11. Jin, K., Liu, M.: Multi-planner intervention in network games with community structures. arXiv preprint arXiv:2103.16351 (2021)
12. Lye, K.W., Wing, J.M.: Game strategies in network security. Int. J. Inf. Secur. **4**(1–2), 71–86 (2005)
13. Orlin, J.B.: Max flows in o (nm) time, or better. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, pp. 765–774. ACM (2013)
14. Papadimitriou, C.H.: On the complexity of the parity argument and other inefficient proofs of existence. J. Comput. Syst. Sci. **48**(3), 498–532 (1994)
15. Sandler, T.: Terrorism & game theory. Simul. Gaming **34**(3), 319–337 (2003)
16. Srivastava, V., et al.: Using game theory to analyze wireless ad hoc networks. IEEE Commun. Surv. Tutorials **7**(4), 46–56 (2005)
17. Wang, S., Liu, F., Shroff, N.: Non-additive security games. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
18. Wang, S., Shroff, N.: Security game with non-additive utilities and multiple attacker resources. Proc. ACM Meas. Anal. Comput. Syst. **1**(1), 1–32 (2017)
19. Washburn, A., Wood, K.: Two-person zero-sum games for network interdiction. Oper. Res. **43**(2), 243–251 (1995)
20. Xu, H.: The mysteries of security games: Equilibrium computation becomes combinatorial algorithm design. In: Proceedings of the 2016 ACM Conference on Economics and Computation, pp. 497–514 (2016)