Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

RALF BORNDÖRFER, JULIA BUWAYA, GUILLAUME SAGNOL, ELMAR SWARAT

Network Spot Checking Games: Theory and Application to Toll Enforcing in Transportation Networks

ZIB-Report 14-07 (March 2014)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: `bibliothek@zib.de`
URL: `http://www.zib.de`

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Network Spot Checking Games: Theory and Application to Toll Enforcing in Transportation Networks

Ralf Borndörfer, Julia Buwaya, Guillaume Sagnol, Elmar Swarat*
Zuse Institut Berlin (ZIB), Department Optimization, Berlin, Germany†

Abstract

We introduce the class of network spot-checking games (NSC games). These games can be used to model many applied problems where the goal is to distribute inspection teams over a network. In contrast to the previously introduced class of *security games* – which is encompassed by the present class of NSC games – the pure strategies of a player correspond to the paths joining two vertices in a graph. Although the game is not zero-sum, and although the number of involved strategies may grow exponentially with the size of the graph, we show that a Nash equilibrium can be computed efficiently by Linear Programming. We next give a Mixed Integer Programming formulation for the computation of a Stackelberg equilibrium. Finally, experimental results based on an application to the enforcement of a truck toll on German motorways are presented.

Keywords Game Theory; Stackelberg Equilibrium; Security Games; Mixed Integer Programming

1 Introduction

In 2005 Germany introduced a distance-based toll for trucks weighing twelve tonnes or more in order to fund growing investments for maintenance and extensions of motorways. The enforcement of the toll is the responsibility of the German Federal Office for Goods Transport (BAG), who has the task to carry out a network-wide control. To this end, 300 vehicles make control tours on the entire motorway network. In this paper, we present some theoretical work obtained in the framework of our cooperation with the BAG, whose final goal is to develop an optimization tool to schedule the control tours of the inspectors. This real-world problem is subject to a variety of legal constraints, which we handle by mixed integer programming [3]. In a follow-up work, we plan to use randomized schedules generated by the game-theoretical approach of the present paper as an input for the real-world problem.

In this paper, we study from a game-theoretic point of view the problem of allocating inspectors to spatial locations of a transportation network, in order to enforce the payment of a transit toll. In a precedent version of this article which appeared as a conference paper [1], we have used a Stackelberg game to represent the applied problem evoked above. We shall extend the model of [1] by formally defining the class of *network spot-checking games* in Section 2. Many practical situations can be represented by this new class of games, as will be seen in Section 3. In particular, it is also possible to take into account the displacement of inspectors in the network over time.

*{borndoerfer,buwaya,sagnol,swarat}@zib.de
†This work was funded by the German federal office for Good Transport (BAG).

Network spot-checking games generalize the class of *security games* (a reduction is given in appendix), by using a graph structure to represent the strategies of *attackers* and *defenders*. The latter has been introduced by Kiekintveld et. al. [6] to study problems where the goal is to randomize different kind of inspections, in a strategical way; this includes a work on the optimal selection of checkpoints and patrol routes to protect the LA Airport towards adversaries [8], a study of the scheduling and allocation of air marshals to a list of flights in the US [5], or the problem of optimally scheduling fare inspection patrols in LA Metro [10].

Contrary to standard security games, the players of a network spot-checking game might have a very large number of available strategies, arising from the multitude of origin-destination paths in a network. For our application to toll enforcement in a transportation network, this new model takes into account every possible detour that fare evaders could take to avoid frequently inspected sections. In contrast, previous approaches used the trivial topology of a single metro line [10], or assumed that each user takes the shortest path [2].

This article is not the first to consider a security game with a huge number of available actions, though. In [5], this difficulty was handled with a branch-and-price algorithm. The approach used in this article is different: if we represent user strategies by network flows, we can give a compact LP formulation for the computation of a Nash equilibrium of the game, cf. Section 4.1. We next use some ideas of [8] to find a Stackelberg equilibrium of the game by mixed integer programming (MIP), see Section 4.2. Experimental results based on real traffic data (for the application to the truck toll in Germany) are given in Section 5, and suggest that the Nash equilibrium strategy is a good trade-off between computation time and efficiency of the controls.

2 Network Spot-Checking Games

In this section we extend the game theoretic model presented in [2], which studies the interaction between the fare inspectors and the users of a transportation network, to handle the case where every user is free to choose his path in the network to reach his destination.

Preliminaries We first recall some basic notions of game theory. In a game with N players where each player may choose a strategy $\boldsymbol{p_i}$ in a set Δ_i, and wishes to maximize his own payoff $u_i(\boldsymbol{p_i}, \boldsymbol{p_{-i}})$, we say that $\boldsymbol{p_i}$ is a best response to the set $\boldsymbol{p_{-i}} \equiv \{\boldsymbol{p_1}, \ldots, \boldsymbol{p_{i-1}}, \boldsymbol{p_{i+1}}, \ldots, \boldsymbol{p_N}\}$ of strategies of the other players if

$$\forall \boldsymbol{p_i'} \in \Delta_i, \qquad u_i(\boldsymbol{p_i'}, \boldsymbol{p_{-i}}) \leq u_i(\boldsymbol{p_i}, \boldsymbol{p_{-i}}).$$

We denote by $BR_i(\boldsymbol{p_{-i}}) \subseteq \Delta_i$ the set of best responses to $\boldsymbol{p_{-i}}$ for Player i. This allows us to define the Nash equilibria of the game:

$$(\boldsymbol{p_1}, \ldots, \boldsymbol{p_N}) \text{ is a Nash equilibrium} \quad \Longleftrightarrow \quad \forall i \in \{1, \ldots, N\}, \quad \boldsymbol{p_i} \in BR_i(\boldsymbol{p_{-i}}).$$

In this paper we will also study some Stackelberg equilibria, which are arguably more adapted to the present class of spot-checking games because of the asymmetry between controllers and network users, and have already been used in several applications [8, 5, 10]. In a Stackelberg game, it is assumed that a player is the leader (in our case, the controller), who plays first, and the other players (called followers) react with a best response. If the leader of the game is denoted by the index 1, a Stackelberg equilibrium is a profile of strategies $\boldsymbol{p} \equiv (\boldsymbol{p_1}, \ldots, \boldsymbol{p_N})$ which maximizes the leader's payoff, among the set of all profiles such that the followers' strategies $\boldsymbol{p_{-1}}$ are in best response relationship to each other's action:

$$\boldsymbol{p}_{\text{eq}} \text{ is a Stackelberg equilibrium} \quad \iff$$
$$\boldsymbol{p}_{\text{eq}} \in \underset{\left\{\boldsymbol{p}\in\Delta_1\times\ldots\Delta_N \,:\, \forall i\neq 1,\ \boldsymbol{p_i}\in BR_i(\boldsymbol{p_{-i}})\right\}}{\arg\max} \quad u_1(\boldsymbol{p_1}, \boldsymbol{p_{-1}}).$$

This definition is quite complicated because of the possible interactions between the players $\boldsymbol{p_2}, \ldots, \boldsymbol{p_N}$. In fact, in the class of network spot-checking games that we will define hereafter, the set $BR_i(\boldsymbol{p_{-i}})$ depends only on the leader's strategy $\boldsymbol{p_1}$ for every follower ($i \neq 1$), see Lemma 2.1. We can hence denote by $BR(\boldsymbol{p_1}) := BR_2(\boldsymbol{p_1}) \times \ldots \times BR_N(\boldsymbol{p_1})$ the set of best response profiles for the followers to a leader's action $\boldsymbol{p_1}$, and the problem of finding an equilibrium reduces to the following optimization problem:

$$\max_{\boldsymbol{p_1}\in\Delta_1} \quad \max_{\boldsymbol{p_{-1}}\in BR(\boldsymbol{p_1})} \quad u_1(\boldsymbol{p_1}, \boldsymbol{p_{-1}}). \tag{1}$$

Note that the definition implicitly implies that when a follower has several best response actions available, he will select one that favors the leader most.

We next define the general class of *network spot-checking games* studied in this article. They describe the interaction between a *controller*, who represents all inspectors of a transportation network, and the *users* of the network, which are distributed over a set of commodities (Origin-Destination pairs of the network).

Problem definition A network spot-checking game (NSC game) $\mathcal{G} = (V, E, \mathcal{K}, \boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\sigma}, P, \alpha, \boldsymbol{\beta}, \mathcal{Q})$ is defined by the following elements:

- A directed graph $G = (V, E)$;
- A set of commodities $\mathcal{K} \subset V \times V$ representing Origin-Destination pairs (s_k, d_k) of the graph G;
- For all $k \in \mathcal{K}$, the number of users x_k of commodity k;
- For all $e \in E$, a cost $w_e \geq 0$ for a user taking edge e;
- For all $e \in E$, a reward β_e (resp. a penalty if $\beta_e < 0$) for the controller for each user using edge e; this β_e typically corresponds to a fare for taking edge e.
- For all $e \in E$, the probability σ_e for an individual passing on e to be controlled, conditionally to the presence of an inspector on e;
- The amount of the penalty P users must pay each time they are controlled;
- A fraction $\alpha \in [0, 1]$ of the penalties to be considered in the controller's payoff;
- A set $\mathcal{Q} \subset [0, 1]^E$ described by linear inequalities, representing possible distributions of the inspectors over the edges of the graph. The quantity q_e corresponds to the probability that some inspector is present on edge e.

At first sight, it may seem odd that every controlled user gets fined and has to pay the penalty P. The reason behind is that the strategies of the users are completely represented by their paths in the network. Typically, to model the fact that not all users are fare evaders, we can consider artificial toll edges in the network with the following properties: the cost w_e for taking a toll edge e includes a fare, and no inspection may be conducted on e ($\sigma_e = 0$). In the situation where the controller's reward on e corresponds to the toll fare $\beta_e \geq 0$ that a network user must pay to use edge e, a fair assumption could be that $\beta_e\sigma_e = 0$ for all edges, i.e. *paying a fare* and *getting fined* exclude each other. We refer the reader to Section 3 for more details and examples on how to model practical situations using NSC games.

Controller's strategy In the simplest variant of the problem, there are γ teams of inspectors over the network, who can each control an edge $e \in E$. The set $\mathcal{S}$ of pure strategies for the controller hence corresponds to the subsets S of E of cardinality γ. The inspectors commit to a mixed strategy $\tilde{\boldsymbol{q}}$, i.e. for all $S \in \mathcal{S}$, $\tilde{q}_S$ indicates the probability of allocating one team of inspectors on each of the γ edges $e \in S$. In practice, we will see that our model only depends on the marginals $q_e = \sum_{\{S \in \mathcal{S}: S \ni e\}} \tilde{q}_S$, which represent the probability that a controller team is present on edge e. It is easy to see that the marginal strategy $\boldsymbol{q}$ satisfies

$$\sum_{e \in E} q_e = \gamma, \tag{2}$$

$$\forall e \in E,\ 0 \leq q_e \leq 1. \tag{3}$$

Conversely, if we are given a vector $\boldsymbol{q}$ satisfying Equations (2) and (3), we can find a mixed strategy $\tilde{\boldsymbol{q}}$ whose marginal equals $\boldsymbol{q}$. To see this, one can notice that the set of extreme points of the polyhedron $\mathcal{Q}$ defined by Equations (2) and (3) coincide with the set of pure strategies $\mathcal{S}$. More evolved strategy sets $\mathcal{Q}$ for the controller, taking into account the displacement of the inspectors over time, will be discussed in §3.3.

User flows and payoffs We associate the users of commodity k with a single player (called Player k). Let $\mathcal{R}_k$ denote the set of paths from s_k to d_k in $G = (V, E)$. Player k can choose any path $R \in \mathcal{R}_k$, so his mixed strategy can be interpreted as the distribution of the $k-$users over $\mathcal{R}_k$. Our model depends only on the probabilities p_e^k that Player k uses edge e, that must form a flow of value one through commodity k: $\forall v \in V$,

$$\sum_{e' \in \delta^+(v)} p_{e'}^k - \sum_{e \in \delta^-(v)} p_e^k = \begin{cases} 1 & \text{if } v = s_k; \\ -1 & \text{if } v = d_k; \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The probability to be controlled on an edge $e \in R$ is $q_e \sigma_e$, and hence the expected number of times Player k is subjected to a control during his trip is $\sum_{e \in R} p_e^k q_e \sigma_e$. The total expected cost of Player k can now be expressed as:

$$\text{Payoff}_k(\boldsymbol{p}, \boldsymbol{q}) = -\left(\sum_{e \in E} p_e^k w_e + \sum_{e \in E} p_e^k q_e \sigma_e P \right), \tag{5}$$

where the first term accounts for travel and toll costs, while the second is the expected fine. Note that we do as if evaders could be fined several times; in practice, this is only a simplifying assumption, since in most toll networks fare evaders can be fined only once (fine receipts count as a valid proof of payment). For a reasonable number of controllers, our results show that the probability of being controlled more than once is very small, though. A similar linear approximation has been used in [10] and [2]).

Observe that the payoff of any user does not depend on the strategy of other users. Hence the game is equivalent to a Bayesian game where the controller plays against a random user $k \in \mathcal{K}$, with probabilities proportional to x_k [8]. An important consequence of Equation (5) is a characterization of the best responses of Player k to the controller's strategy $\boldsymbol{q}$:

Lemma 2.1. *Let $\boldsymbol{q} \in \mathcal{Q}$ be a strategy of the controller, and denote by $\lambda_k(\boldsymbol{q})$ the length of the shortest path from s_k to d_k in the weighted graph $G = (V, E, \boldsymbol{c}(\boldsymbol{q}))$, where the weight of edge e is $c_e(\boldsymbol{q}) = w_e + q_e \sigma_e P$. A strategy $\boldsymbol{p^k}$ for Player k is a best response to $\boldsymbol{q}$ if and only if $-\text{Payoff}_k(\boldsymbol{p}, \boldsymbol{q}) = \lambda_k(\boldsymbol{q})$. In other words, best responses for Player k are flows through commodity k supported by shortest paths of $G = (V, E, \boldsymbol{c}(\boldsymbol{q}))$.*

Proof. If $\boldsymbol{p^k}$ is a flow of unit value through commodity k, then $-\text{Payoff}_k(\boldsymbol{p}, \boldsymbol{q}) = \sum_{e \in E} p_e^k (w_e + q_e \sigma_e P)$ corresponds to the expected length for Player k from s_k to d_k in the weighted graph $G = (V, E, \boldsymbol{c}(\boldsymbol{q}))$. This expression is minimized if and only if the flow $\boldsymbol{p^k}$ uses only shortest paths. $\square$

Controller's payoff The total payoff of the controllers is obtained by summing the collected rewards and penalties, and depend on the parameters α and $\boldsymbol{\beta}$ defined above:

$$\mathrm{Payoff}_C(\boldsymbol{p}, \boldsymbol{q}) = \sum_k x_k \sum_{e \in E} p_e^k (\alpha \sigma_e q_e P + \beta_e). \tag{6}$$

The extreme values of α correspond to two important situations. If $\alpha = 1$, the payoff defined in (6) corresponds to the total revenues from rewards and penalties, a setting which we denote by MAXPROFIT. If $\alpha = 0$, the controller's payoff comes from the fares only (assuming that the reward β_e is a fare for edge e). This setting, which we call MAXTOLL, might be well suited if the goal is solely to enforce the payment of a fare. In contrast, with MAXPROFIT it might be advantageous to have a bit of evasion on certain commodities, in order to earn money from fines. The parameter α may be seen as a parameter weighting the objectives of MAXTOLL and MAXPROFIT, for the scalarization of a biobjective problem.

3 Handling Practical Situations with NSC Games

The general model introduced in the previous section can be used to model a variety of practical situations. We next review important examples inspired from real-world applications. In addition, every security game can be encoded as an NSC game, as shown in appendix.

3.1 A transit network model

We start by showing that the model used in the conference paper [1] at the origin of this article can be cast as a NSC game. Here, the network users are assumed to travel over a network $G_0 = (V_0, E_0)$ with edge costs w_e and conditional inspection risks σ_e. For a given commodity $k = (s_k, d_k) \in \mathcal{K}_0 \subset V_0 \times V_0$, Player k can either decide to pay a fare τ_k (and in this case he will take the shortest path from s_k to d_k), or he can evade the fare and choose an arbitrary path from s_k to d_k in G_0.

To represent the strategies corresponding to paying the fare, we create a set of additional vertices $\bar{V}$, containing a node $\bar{s}$ for every source $s \in S = \{s_k : k \in \mathcal{K}_0\}$ and a node $\bar{\bar{d}}$ for every destination $d \in D = \{d_k : k \in \mathcal{K}_0\}$.

We then connect these new nodes by creating a set of artificial edges $\bar{E}$, that contains

- an edge $\bar{s} \to s$ for all $s \in S$, which corresponds to the action of evading the fare, with all weights equal to zero: $w_{[\bar{s}s]} = \beta_{[\bar{s}s]} = \sigma_{[\bar{s}s]} = 0$;
- an edge $d \to \bar{\bar{d}}$ for all $d \in D$, with all weights equal to zero as well;
- an edge $\bar{s_k} \to \bar{\bar{d_k}}$ for all $k \in \mathcal{K}_0$, which corresponds to the action of paying the fare. The user cost for taking this edge is defined by $w_{[\bar{s_k}\bar{\bar{d_k}}]} = l_k(\boldsymbol{w}) + \tau_k$, where $l_k(\boldsymbol{w})$ is the length of the shortest path from s_k to d_k in the weighted graph G_0 with weights $\boldsymbol{w}$. The controller's reward is $\beta_{[\bar{s_k}\bar{\bar{d_k}}]} = \tau_k$, and users cannot be fined on this edge ($\sigma_{[\bar{s_k}\bar{\bar{d_k}}]} = 0$).

By construction, the artificial toll edge $(\bar{s_k}, \bar{\bar{d_k}})$ can only be used by users of the commodity k. This ensures that the artificial toll edge $(\bar{s_k}, \bar{\bar{d_k}})$ can only be used by users of the commodity k. The original edges $e \in E_0$ are taken by users evading the fare, so that the controller's reward β_e can be set to 0 on these edges; alternatively, β_e can be set to a negative value to penalize fare evasion in the controller's payoff. (It is also

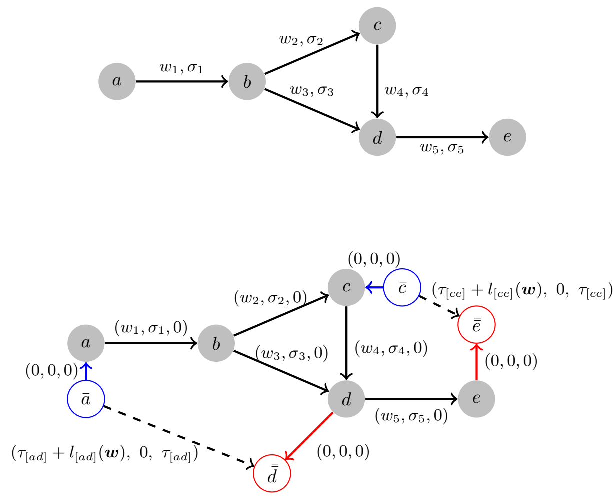


Figure 1: Original transit network G_0 (above), and extended structure G of the NSC game (below). The edges of G are labelled with triples (w_e, σ_e, β_e). The subset of commodities is $\mathcal{K}_0 = \{[ad], [ce]\}$, which becomes $\mathcal{K} = \{[\bar{a}\bar{\bar{d}}], [\bar{c}\bar{\bar{e}}]\}$ in the extended graph.

possible to penalize evasion on a per-user basis, by setting a negative reward on the artificial evasion edges $s \to \bar{s}$.)

The NSC game is obtained by considering the extended graph $G = (V, E)$, where $V = V_0 \cup \bar{V}$ and $E = E_0 \cup \bar{E}$, as well as the set of extended commodities $\mathcal{K} = \{(\bar{s_k}, \bar{\bar{d}}_k) : k \in \mathcal{K}_0\}$. An example graph G_0 is represented together with its extension G and the associated edge weights in Figure 1.

3.2 Transportation networks with a distance-based toll

With the previous model, it is not possible to consider network users who pay the fare on a portion of their trip only. This is of particular relevance for the application mentioned in the introduction of this article, since on the German motorways, a transit fee proportional to the distance must be paid by the truck drivers. So crafty drivers might take the chance to pay the toll on a short portion of their trip only, where they know that the frequency of controls is high.

To represent users' strategies in this situation, we shall now introduce a two-layer graph structure. Let $G_0 = (V_0, E_0)$ represent the physical transportation network, and $\mathcal{K}_0 \subset V_0 \times V_0$ be a subset of commodities. For all $k \in \mathcal{K}_0$, s_k and d_k represent the origin and the destination of k. We denote by l_e the length of edge $e \in E_0$ and by f (resp. b) the toll rate (resp. the average basic costs such as e.g. fuel consumption) per kilometer. In addition, we assume that the risk of inspection on an edge $e \in E_0$ is σ_e (conditionally to the presence of a team of inspectors on e).

We create a set V_1, containing a vertex v' for every $v \in V_0$, as well as a set $\bar{V}$ of artificial start and destination nodes, containing a vertex $\bar{s}$ for every $s \in S = \{s_k : k \in \mathcal{K}_0\}$ and a vertex $\bar{\bar{d}}$ for every $d \in D = \{d_k : k \in \mathcal{K}_0\}$. To connect these new vertices, we create the following sets of edges:

- $E_1 = \{(u', v') : \forall (u, v) \in E_0\}$;
- $E_s = \{(\bar{s}, s), (\bar{s}, s') : \forall s \in S\}$;

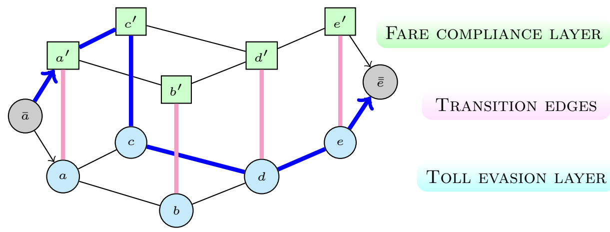


Figure 2: Example of two-layer Graph with a single commodity $\mathcal{K} = \{(a, e)\}$. The path highlighted in blue indicates a user paying the fare on $a \to c$, and evading the toll on $c \to d \to e$.

- $E_d = \{(d, \bar{\bar{d}}), (d', \bar{\bar{d}}) : \ \forall d \in D\}$;
- $E_t = \{(v, v'), (v', v) : \ \forall v \in V_0\}$.

The edges of the level E_1 represent portions of a trip where the toll fee has been paid, while the matching edges in E_0 correspond to fare evasion. Transition edges $e \in E_t$ allow the users to switch between these two layers, at a cost θ that should be set to represent the reluctance of users to change strategy during a trip. Artificial edges of E_s and E_d ensure the connectivity of the new commodities $\mathcal{K} := \{(\bar{s_k}, \bar{\bar{d_k}}) : \ k \in \mathcal{K}_0\}$ with both layers. To sum up, the different edge weights are defined by:

$$\beta_e = \begin{cases} f l_e & \text{if } e \in E_1; \\ 0 & \text{otherwise,} \end{cases}$$

$$w_e = \begin{cases} b l_e & \text{if } e \in E_0; \\ (b+f) l_e & \text{if } e \in E_1; \\ \theta & \text{if } e \in E_t; \\ 0 & \text{otherwise,} \end{cases}$$

and $\sigma_e = 0$ for all $e \notin E_0$. A simple example of two-layer graph is depicted on Figure 2.

3.3 Spatio-temporal aspects

The models presented so far do not take time into account. This is an important challenge, since the inspectors must move along edges of the networks and their duties must not exceed a certain length. In consequence, the set $\mathcal{Q}$ defined by Equations (2) and (3) might not be well-suited to represent all possible marginal strategies of the controller.

The authors of [10] have proposed to represent the duties of metro ticket inspectors by flows in an adapted graph. Their approach provides exact schedules for each inspector. More precisely, a strategy consists in a sequence of trains that the controller must control at a given time. However, this very fine model might not be very robust to any kind of delays or incidents that can occur in the inspection process, so that the inspectors might not be able to follow the prescribed schedule. To cope with this problem, planners of the BAG (the authority in charge of fare inspections on the German motorways, cf. Section 1) allocate the inspectors to a set of predefined control areas, on which they must patrol during a given time interval, see [3]. The graph structure which we next present combines ideas of [3] and [10].

We consider a time discretization $\mathcal{T} = \{0, \ldots, T-1\}$ of the period of interest, typically one day, and we make the simplifying assumption that every network user starts and ends her trip within the same time

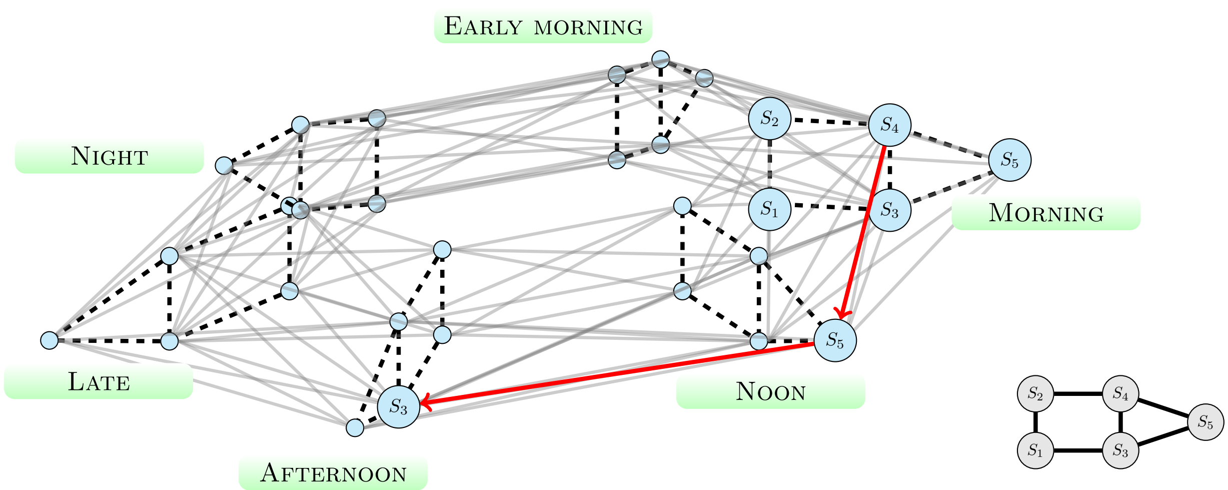


Figure 3: Example for a graph C connecting the control areas (lower right corner) and its associated cyclic duty graph D (main drawing), for a time discretization of one day with $T = 6$ time windows. The path highlighted in red represents the duty of a team controlling S_4 during the morning, S_5 at noon and S_3 during the afternoon.

window $t \in \mathcal{T}$. We denote by $G_0 = (E_0, V_0)$ the graph representing the static problem (obtained e.g. by using the construction of § 3.1 or § 3.2), and we make a time extended graph $G = (V, E)$ which contains T parallel copies of G_0: $V = V_0 \times \mathcal{T}$ and $E = E_0 \times \mathcal{T}$. A commodity k in G corresponds to a pair of nodes $(s_k, d_k) \in V^2$, such that $s_k = (u, t)$ and $d_k = (v, t)$ for a pair of nodes $(u, v) \in V_0^2$ and a time window $t \in \mathcal{T}$.

A control area $S \in \mathcal{S}$ consists of a subset of edges $S \subset E_0$ (control areas might overlap). We create a graph $C = (\mathcal{S}, A)$ which connects nearby control areas, i.e. $(S_i, S_j) \in A$ whenever it is possible for a team of inspectors to control S_i at time t and S_j at $t+1$. Again, we create a time extended version $D = (\mathcal{S} \times \mathcal{T}, \bar{A})$ of C, which we call the *cyclic duty graph*, as follows:

$$\begin{aligned} \bar{A} = &\Big\{ \big((S,t),(S, t+1 \bmod T)\big) :\ \forall S \in \mathcal{S} \Big\} \\ &\bigcup \ \Big\{ \big((S,t),(S', t+1 \bmod T) :\ \forall (S,S') \in A \big) \Big\}. \end{aligned}$$

We have depicted in Figure 3 a simple example for a graph C and the corresponding cyclic duty graph D. The inspectors' duties can be represented by paths in D. In practice, duties have a prescribed length, for example 8 hours, which corresponds to paths of a certain length L in D. With a simple construction, it is possible to create a modified duty graph $\tilde{D}$ with start and end depot nodes d_s and d_t, that enjoys the property that every $(d_s, d_t)-$path corresponds to a path of length L in D. Hence the mixed strategy of a single inspector can be represented by a $(d_s, d_t)-$flow of value one in $\tilde{D}$. We refer the reader to [10] for details about this construction.

Now, we assume that there are γ teams of inspectors, as in the paragraph preceding Equation (2). The controller's strategy can hence be represented by a $(d_s, d_t)-$flow $\tilde{\boldsymbol{q}}$ of value γ in $\tilde{D} = (\tilde{V}, \tilde{A})$:

$$\forall v \in \tilde{V}, \ \sum_{a' \in \delta^+(v)} \tilde{q}_{a'} - \sum_{a \in \delta^-(v)} \tilde{q}_a = \begin{cases} \gamma & \text{if } \ v = d_s; \\ -\gamma & \text{if } \ v = d_t; \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The vertex set of $\tilde{D}$ is

$$\tilde{V} = \mathcal{S} \times \mathcal{T} \times \{1, \dots, L\} \cup \{d_s, d_t\},$$

and similarly as in [10] it can be seen that the expected number of inspectors in the control area $S \in \mathcal{S}$ at time t is

$$\hat{q}_{(S,t)} = \sum_{l=1}^{L} \sum_{a \in \delta^-(S,t,l)} \tilde{q}_a. \tag{8}$$

As a simple approximation we can assume that the inspectors are spread uniformly on all the arcs of a control area, so that the an inspector on the control area S is present on edge $e \in S$ with probability

$$\kappa_{e|S} = \frac{l_e}{\sum_{e' \in S} l_{e'}},$$

where l_e denotes the length of edge e. It follows that the expected number of inspectors on $e \in E_0$ at time t is

$$\sum_{\{S \in \mathcal{S}\colon S \ni e\}} \kappa_{e|S}\ \hat{q}_{(S,t)}.$$

If this quantity is smaller than one, it can be interpreted as the marginal probability $q_{(e,t)}$ to find an inspector team on the edge $(e,t) \in E$ of the time extended graph G. To summarize, the set of marginal strategies $\mathcal{Q}$ of the controller can be defined by:

$$\begin{aligned} \mathcal{Q} = \Big\{ \boldsymbol{q} \in & (\mathbb{R}_+)^E :\ \exists \tilde{\boldsymbol{q}} \in (\mathbb{R}_+)^{\tilde{A}} \text{ s.t.} \\ & (i)\ \tilde{\boldsymbol{q}} \text{ satisfies the flow conservation (7);} \\ & (ii)\ \forall (e,t) \in E, \\ & \qquad q_{(e,t)} \leq \sum_{\{S \in \mathcal{S}\colon S \ni e\}} \kappa_{e|S} \sum_{l=1}^{L} \sum_{a \in \delta^-(S,t,l)} \tilde{q}_a; \\ & (iii)\ \forall (e,t) \in E,\quad q_{(e,t)} \leq 1 \Big\}. \end{aligned}$$

To conclude this section, we briefly mention some simple extensions that can be plugged in this model (by adapting the graph G or $\tilde{D}$ in an intuitive fashion):

- Several side constraints can be added in the above definition of $\mathcal{Q}$. For example, the proportion of duties starting at night can be bounded from above, or we can bound from below the inspection frequency of some control areas to ensure a network-wide control.
- If not all the controllers start from the same location in the network, it is possible to consider several start and end depot nodes in the duty graph $\tilde{D}$.
- The possibility for a user to advance or postpone her departure (in order to travel at a time with less controls) could be represented by adding edges in G that link the different time copies of G_0, with a cost ς for the delay.

We shall now return to the general model of NSC games introduced in Section 2, and we study the problem of computing some equilibria for a generic game $\mathcal{G} = (V, E, \mathcal{K}, \boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\sigma}, P, 1, \boldsymbol{\beta}, \mathcal{Q})$.

4 Computation of Equilibria

The notion of equilibrium is essential in game theory. Depending on the ability of the players to observe the others' actions, committing to a Nash or a Stackelberg equilibrium may be better suited [7]. However, there is a natural interpretation for the Stackelberg strategies of the controller: the Stackelberg game model assumes that every user of the network plays with a best response to the controller's strategy $\boldsymbol{q}$. In particular, a Stackelberg strategy for MAXTOLL ($\alpha = 0$) maximizes the (weighted) number of users who have an incentive to pay the fares (with weights corresponding to the fares). In fact, one can expect that many users are always honest and pay the network fares independently of the frequency of inspections. To some extent, the Stackelberg equilibrium can hence be considered as an approach to maximize the Controller's payoff in the worst case. This is not truly the worst-case situation, since network users could take only toll-free sections (in the case of a transit system for example, they could walk), thus depriving the Controller from all sources of profit. However, there is no reason to assume that network users want to minimize the Controller's payoff, and the Stackelberg approach guards ourselves from crafty behaviours.

4.1 Nash equilibrium for MAXPROFIT

We next show that in the case of MAXPROFIT ($\alpha = 1$), the NSC game can be transformed into a *zero-sum game* which has the same Nash equilibria. As a consequence, a Nash equilibrium strategy can be computed by linear programming.

Proposition 4.1 (Reduction to a zero-sum game)**.** *The game* $\mathcal{G} = (V, E, \mathcal{K}, \boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\sigma}, P, 1, \boldsymbol{\beta}, \mathcal{Q})$ *has the same set of Nash equilibria as the zero-sum game* $\mathcal{G}' = (V, E, \mathcal{K}, \boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\sigma}, P, 1, \boldsymbol{w}, \mathcal{Q})$*, where the controller's rewards* β_e *have been replaced by the edge costs* w_e*.*

Proof. First note that the game $\mathcal{G}'$ is zero-sum indeed:

$$\mathrm{Payoff}_C(\boldsymbol{p}, \boldsymbol{q}) + \sum_k x_k \mathrm{Payoff}_k(\boldsymbol{p}, \boldsymbol{q}) = 0.$$

The Nash equilibria are entirely defined by the set of best responses of every player. We are going to see that these sets coincide for $\mathcal{G}$ and $\mathcal{G}'$, from which the conclusion follows. The payoff of Player k is the same in both games, so it is clear that $BR_k(\boldsymbol{q})$ is the same in these two games (for all $k \in \mathcal{K}$). Now, observe that the set of best responses for the controller in $\mathcal{G}$ is

$$BR_C^{\mathcal{G}}(\boldsymbol{p}) = \arg\max_{\boldsymbol{q} \in \mathcal{Q}} \sum_k x_k \sum_{e \in E} p_e^k (\sigma_e q_e P + \beta_e).$$

For a fixed $\boldsymbol{p}$, let us add $\sum_k x_k (\sum_{e \in E} p_e^k (w_e - \beta_e))$ in the function to maximize. This does not change the set of maximizers, since the new term does not depend on $\boldsymbol{q}$. Hence,

$$\begin{aligned} BR_C^{\mathcal{G}}(\boldsymbol{p}) &= \arg\max_{\boldsymbol{q} \in \mathcal{Q}} \sum_k x_k \sum_{e \in E} p_e^k (\sigma_e q_e P + w_e) \\ &= BR_C^{\mathcal{G}'}(\boldsymbol{p}) \end{aligned}$$

□

It is well known that a Nash equilibrium of 2-player zero-sum games can be computed by linear programming. In our case, the game $\mathcal{G}'$ has more than two players, but its special structure (no interaction between the network users) could allow us to formulate the game as a *polymatrix game* with an underlying star-shaped graph, a class of games for which a Nash equilibrium can also be computed by linear programming, see [2, 4]. However, the LP that we would obtain would have a constraint for each pure strategy of the network users, a number which might be exponentially large. To cope with this problem, we shall next exploit the flow representation of the users' strategies and the shortest path chcharacterization of best reponses to give a compact LP formulation for computing a Nash equilibrium of $\mathcal{G}$.

By Lemma 2.1 we know that the loss of Player k at a Nash equilibrium $(\boldsymbol{p}, \boldsymbol{q})$ is $\lambda_k(\boldsymbol{q})$, the length of the shortest path from s_k to d_k in $G = (V, E, \boldsymbol{c}(\boldsymbol{q}))$. The computation of a Nash equilibrium of $\mathcal{G}'$ (and hence of $\mathcal{G}$) thus reduces to the computation of a strategy $\boldsymbol{q} \in \mathcal{Q}$ maximizing $\sum_k x_k \lambda_k(\boldsymbol{q})$. This can be done by linear programming, by introducing some node potentials y_v^s for every source node $s \in S := \{s_k : k \in \mathcal{K}\}$ and for all $v \in V$:

$$\max_{\boldsymbol{q},\boldsymbol{y}} \quad \sum_{k \in \mathcal{K}} x_k y_{d_k}^{s_k} \tag{9a}$$

$$\text{s.t.} \quad y_v^s - y_u^s \le w_e + \sigma_e q_e P, \quad \forall s \in S,\ \forall e \equiv (u, v) \in E; \tag{9b}$$

$$y_s^s = 0, \qquad \forall s \in S; \tag{9c}$$

$$\boldsymbol{q} \in \mathcal{Q}. \tag{9d}$$

The constraints (9b)-(9c) are from the classical linear programming formulation of the single-source shortest path problem, and bound the potential y_v^s from above by the shortest path length from s to v in the graph $G = (V, E, \boldsymbol{c}(\boldsymbol{q}))$. The objective function (9a) hence asks for the maximization of the weighted sum of shortest paths $\sum_k x_k \lambda_k(\boldsymbol{q})$ over the set of feasible controller's strategies (9d).

We point out that the optimal dual variables of constraint (9b) for a given $s \in S$ define a single-source multi-sink flow on the subset of commodities $\mathcal{K}_s := \{k \in \mathcal{K} : s_k = s\}$ originating in s. This flow can be decomposed as a sum of $(s_k, d_k)-$flows, which yields the corresponding Nash equilibrium strategy $\boldsymbol{p^k}$ for every Player k.

Observe that in accordance with Proposition 4.1, the LP (9) for the computation of a Nash equilibrium does not depend on the controller's rewards $\boldsymbol{\beta}$. The concept of Stackelberg equilibrium looks much more suitable for our application, but as we shall see, the computation of such an equilibrium is also much harder.

4.2 Stackelberg equilibria

Using ideas similar as in [8], a mixed integer program (MIP) can be formulated for the computation of a Stackelberg equilibrium $(\boldsymbol{p}, \boldsymbol{q})$. We reduce drastically the number of required variables, by using a single-source multi-sink flow

$$\boldsymbol{\rho^s} = \sum_{\{k: s_k = s\}} x_k \, \boldsymbol{p^k}$$

for each $s \in S$, instead of using a flow $\boldsymbol{p^k}$ for every commodity. With the use of *big-M* constraints, we ensure that this flow uses only edges belonging to a shortest path tree rooted in s in the graph $G = (V, E, \boldsymbol{c}(\boldsymbol{q}))$. By Lemma 2.1, $\boldsymbol{\rho^s}$ hence corresponds to best-response strategies to $\boldsymbol{q}$ for the players whose commodity source is s.

$$\max_{\boldsymbol{q},\boldsymbol{y},\boldsymbol{\mu},\boldsymbol{\rho}} \quad \sum_{k\in\mathcal{K}} \alpha x_k y^{s_k}_{d_k} + \sum_{s\in S}\sum_{e\in E} \rho^s_e(\beta_e - \alpha w_e) \tag{10a}$$

s. t.

$$0 \le w_e + \sigma_e q_e P - (y^s_v - y^s_u) \le M_e(1-\mu^s_e), \qquad \forall s\in S,\ \forall e\equiv(u,v)\in E; \tag{10b}$$

$$y^s_s = 0, \qquad \forall s\in S; \tag{10c}$$

$$\boldsymbol{q}\in\mathcal{Q}, \tag{10d}$$

$$\sum_{e'\in\delta^+(v)} \rho^s_{e'} - \sum_{e\in\delta^-(v)} \rho^s_e = \begin{cases} \sum_{k\in\mathcal{K}_s} x_k & \text{if } s=v;\\ -x_{(s,v)} & \text{if } (s,v)\in\mathcal{K}_s;\\ 0 & \text{otherwise,}\end{cases} \qquad \forall s\in S,\ \forall v\in V; \tag{10e}$$

$$0\le \rho^s_e \le M^s\mu^s_e, \qquad \forall s\in S,\ \forall e\in E; \tag{10f}$$

$$\mu^s_e\in\{0,1\}, \qquad \forall (s,e)\in S\times E. \tag{10g}$$

As in Problem (9), constraints (10b)-(10c) bound $y^{s_k}_{d_k}$ from above by the shortest path length for commodity k in the graph $G=(V,E,\boldsymbol{c}(\boldsymbol{q}))$, and constraint (10d) forces $\boldsymbol{q}$ to be a feasible strategy for the controller. We introduce a binary variable μ^s_e which can take the value 1 only if edge e belongs to a shortest path tree rooted in s (second inequality in (10b)). Indeed, the first inequality in (10b) is saturated when the difference of potential $(y^s_v - y^s_u)$ between the extreme points of an edge $e\equiv(u,v)$ equals the length of e, which indicates that there is a shortest path originating in s that uses e.

Equation (10e) forces $\boldsymbol{\rho^s}$ to be a single-source multi-sink flow rooted in s, whose demand on the commodity $k\in\mathcal{K}_s := \{k\in\mathcal{K}:\ s_k = s\}$ corresponds to the number of users x_k. Constraint (10f) ensures that the flow $\boldsymbol{\rho^s}$ only uses edges from a shortest path tree rooted in s (in the weighted graph with weights given by $\boldsymbol{c}(\boldsymbol{q})$). Now, $\boldsymbol{\rho^s}$ can be decomposed as $\sum_{k\in\mathcal{K}_s} x_k \boldsymbol{p^{(s,d_k)}}$, where $\boldsymbol{p^{(s,d_k)}}$ is a flow through commodity k of value one. By construction, $\boldsymbol{p^{(s_k,d_k)}}$ is a flow of minimal cost $\lambda_k(\boldsymbol{q}) = \sum_{e\in E} p^k_e(w_e + q_e\sigma_e P)$, and it follows that $\boldsymbol{p^k}$ is a best response to $\boldsymbol{q}$, see Lemma 2.1.

Finally, the objective function (10a) rewrites to the controller's payoff (6) when replacing $y^{s_k}_{d_k}$ and ρ^s_e by their values as a function of p^k_e:

$$\begin{aligned}
&\sum_{k\in\mathcal{K}} \alpha x_k y^{s_k}_{d_k} + \sum_{s\in S}\sum_{e\in E}\rho^s_e(\beta_e-\alpha w_e)\\
=&\sum_{k\in\mathcal{K}} \alpha x_k \sum_{e\in E} p^k_e(w_e + q_e\sigma_e P) +\\
&\qquad \sum_{s\in S}\sum_{e\in E}\sum_{k\in\mathcal{K}_s} x_k p^k_e(\beta_e - \alpha w_e)\\
=&\sum_{k\in\mathcal{K}} x_k \sum_{e\in E} p^k_e(\alpha q_e\sigma_e P + \beta_e).
\end{aligned}$$

We point out that the *big-M* constants M_e and M^s can all be chosen in the same order of magnitude as the other coefficients of the problem.

4.3 Stackelberg for MAXTOLL in the transit network model

In this section, we show that the Stackelberg MIP (10) can be simplified for the case of the transit network model introduced in §3.1 with $\alpha = 0$ (MAXTOLL). In this situation indeed, the controller's payoff can be expressed as $\sum_k x_k \tau_k \mu_k$, where μ_k is a binary variable indicating whether Player k has an incentive to pay the toll. So the flows of network users $\boldsymbol{\rho^s}$ is not involved anymore:

$$
\begin{aligned}
\max_{\boldsymbol{q},\boldsymbol{y},\boldsymbol{\mu}} \quad & \sum_{k\in\mathcal{K}} x_k \tau_k \mu_k && && \text{(11a)}\\
\text{s. t.} \quad & && && \\
& y_v^s - y_u^s \leq w_e + \sigma_e q_e P, && \forall s \in S,\ \forall e \equiv (u,v) \in E; && \text{(11b)}\\
& y_s^s = 0, && \forall s \in S; && \text{(11c)}\\
& w_{[\bar{s_k}\bar{\bar{d}}_k]} - y_{\bar{\bar{d}}_k}^{\bar{s_k}} \leq M_k(1-\mu_k), && \forall k \in \mathcal{K}_0; && \text{(11d)}\\
& \mu_k \in \{0,1\}, && \forall k \in \mathcal{K}_0; && \text{(11e)}\\
& \boldsymbol{q} \in \mathcal{Q}. && && \text{(11f)}
\end{aligned}
$$

The binary indicator variable μ_k can take the value 1 if and only if the inequality corresponding to $e = (\bar{s_k}, \bar{\bar{d}}_k)$ in (11b) is saturated, i.e. when the single edge $(\bar{s_k}, \bar{\bar{d}}_k)$ forms a shortest path for commodity k, which means that Player k has an incentive to pay the toll.

5 Experimental Results on German Motorways

We have solved the models presented in this paper for several instances based on real data from the German motorways network. We present here a brief analysis of our results.

Mobile controllers drive on the network and are able to control every truck they overtake. Hence, the probability to be caught on a section where an inspector is driving can be approximated by $(\frac{v_I}{v_T} - 1)$, where v_I and v_T represent the average speeds of the inspectors and the truck drivers, respectively. In our experiments we have set $\sigma_e = 0.15$ for all $e \in E_0$. The amount of the fine was set to $P = 200$.

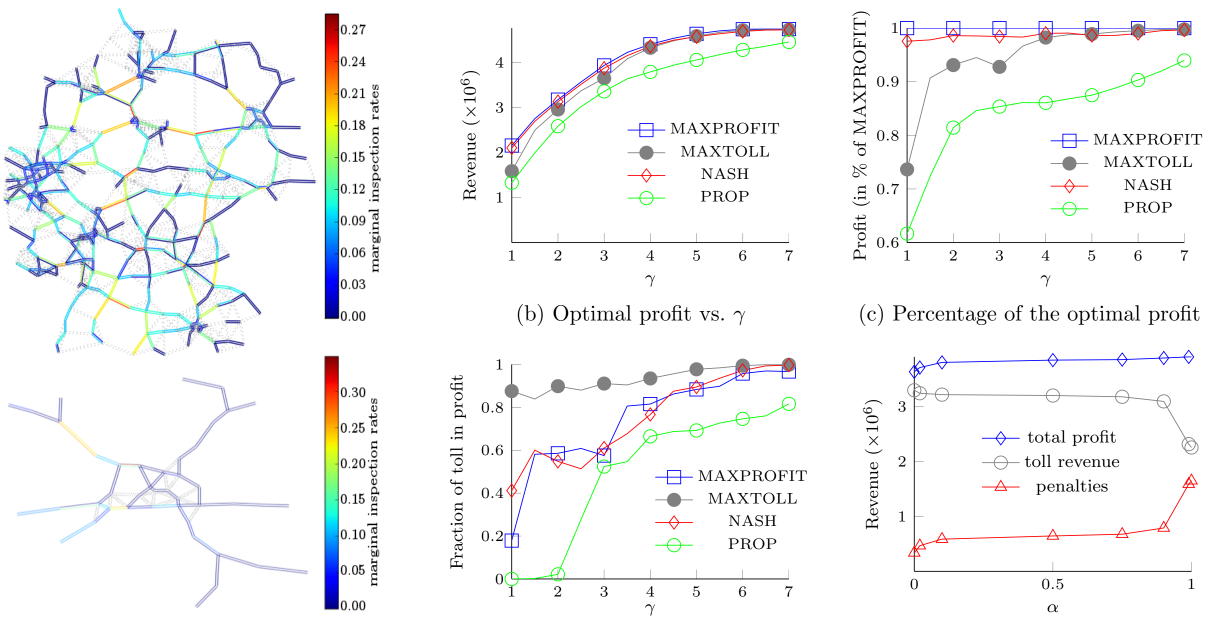
5.1 Static case, transit network model

We first present results for two instances in the static case (i.e., time is not taken into account), with the simple transit network model of § 3.1. The results presented below rely on the set of strategies $\mathcal{Q}$ described in (2)–(3), where a set of γ inspectors can be arbitrarily distributed over the edges of the network.

In upper part of Figure 4(a), a near-Stackelberg equilibrium strategy of the inspectors on the whole German network is represented, for the MAXPROFIT case. Here it was assumed that $\gamma = 50$ controllers are simultaneously present on the network, which has 319 nodes, 2948 edges and 5013 commodities. The dotted edges on the figure represent toll-free roads, where the costs w_e are relatively higher than on the motorway network, but no control can occur (σ_e=0).

To find this near-optimal strategy, we first computed a Nash equilibrium $\boldsymbol{q}$ with the LP (9); this took 29s on a PC with 8 cores at 3.2GHz. Then, we computed a best response strategy for the players of each commodity. When several shortest paths existed for a commodity k in the graph $G = (V, E, \boldsymbol{c}(\boldsymbol{q}))$, we chose a flow $\boldsymbol{p^k}$ that favoured the controller most. This can be done by solving another shortest path problem, in the directed acyclic subgraph of all shortest paths from s_k to d_k in $G = (V, E, \boldsymbol{c}(\boldsymbol{q}))$. This collection of strategies yields a feasible solution for the MIP (10), that can be used for a *warm start*. We used CPLEX, and an optimality gap of 1.5% was reached after $350s$. We point out that the Nash equilibrium differs from the strategy MAXPROFIT on a few edges only, and captures 99.7% of the optimal profit.

Further tests on a smaller network representing the region of Berlin-Brandenburg (45 nodes, 130 edges, 596 commodities) confirm that the Nash equilibrium strategy might be a good trade-off between the computation time and the efficiency of the controls. This network is represented in the lower part of Figure 4(b), with



(a) (b) Optimal profit vs. γ (c) Percentage of the optimal profit (d) Fraction of toll in profit (e) Toll and penalties vs. α

Figure 4: (a): Graphs of the two instances discussed in §5.1, with colors showing a near-optimal strategy for MAXPROFIT and $\gamma = 50$ on the network of Germany (above) and an optimal Stackelberg strategy for MAXTOLL and $\gamma = 3$ for Berlin-Brandenburg (below). (b)-(e): Experimental results for the region of Berlin-Brandenburg.

colors showing a Stackelberg strategy for MAXTOLL and $\gamma = 3$. Figures 4(b)-4(d) compare 4 strategies in function of the number of controllers γ: the strategies MAXPROFIT and MAXTOLL, the Nash equilibrium strategy computed by LP (9), and a strategy in which control intensities are proportional to traffic volumes on each edge of the physical network G_0 (of course, no inspector is allocated to the artificial toll edges where $\sigma_e = 0$). Plot (b) shows the profit collected when committing to one of these strategies (in the Stackelberg model, i.e. drivers select a best response which favors the controller most). We see on Plot (c) that the Nash strategy is always near-optimal in terms of profit; we want to investigate this fact in future research. However, we point out that the MAXTOLL strategy outperforms the others in terms of toll enforcement (Plot (d)), at the price of a small loss in total profit (7% for $\gamma = 2$ and 2% for $\gamma = 4$). In another experiment, we have set $\gamma = 3$ and we have played with the parameter α, which joins MAXPROFIT ($\alpha = 1$) to MAXTOLL ($\alpha = 0$). Plot (e) shows that for $\alpha = 0.75$, one can find a solution with almost the same profit as in MAXPROFIT, but with a higher fraction coming from the toll, and hence less evasion.

5.2 Static case, two-layer network model

We shall now present some results for an instance constructed with the two-layer network model of §3.2. The graph of this instance represents the federal state of Rhineland-Palatinate; the graph G_0 has 37 nodes, 142 edges and 323 commodities, which yields 148 nodes and 506 edges in the two-layer graph G.

The Stackelberg strategy of the controller for MAXPROFIT and $\gamma = 6$ is indicated in Figure 5, as well

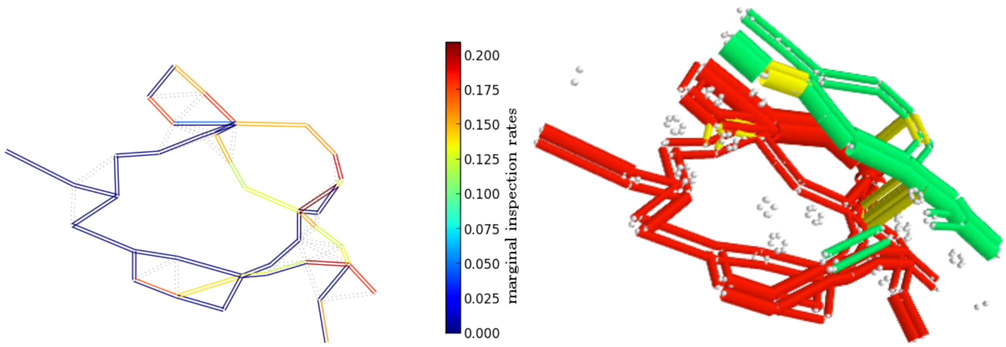


Figure 5: Controller's Stackelberg strategy (above) and Driver flows (below) for the MAXPROFIT problem with $\gamma = 6$. In the lower 3D-figure, the width of the edges is proportional to the number of network users in the two-layer graph. The red edges represent fare evasion, the green edges fare compliance, and the yellow ones correspond to toll-free sections.

as the flows of network users in the two-layer graph. In the northern part of the graph, the red edges in the toll evasion level are much wider than the corresponding green edges in the fare compliance level, which indicates that many users have no incentive to pay the toll here.

The evolution of the controller's payoff with γ is plotted in Figure 6, for the Stackelberg, Nash, and proportional strategies (as a percentage of the maximum, i.e. the profit reached with the Stackelberg strategy). Here again, the Nash strategy is optimal in most cases (except for $\gamma = 10$ where it captures 98.7% of the optimum). Curiously, the controller's payoff sinks between $\gamma = 7$ and $\gamma = 8$ with the proportional strategy. This is explained by the lower graph of Figure 6: when $\gamma = 8$, this strategy assigns a too high control frequency on certain edges, which creates a situation where many drivers have an incentive to take a toll-free trunk road, thus depriving the controller of the incomes from both fares and fines on this edge.

5.3 An example with time dynamics

We have considered an instance with the model of cyclic duty graphs presented in §3.3. The network corresponds to a control region of Germany located around the federal state of Saxony-Anhalt and was represented by a graph G_0 with 26 nodes and 74 edges. The graph C connecting 17 control areas is depicted in the upper part of Figure 7, with nodes whose location coincide with the barycenter of each control area $S \in \mathcal{S}$. We have used a time discretization of two hours, so that an inspector's duty of 8 hours has length $L = 4$ in the cyclic duty graph D.

The lower part of Figure 7 shows the marginal control rates corresponding to the Nash strategy for $\gamma = 20$ on the edges of G_0, during different time windows. Interestingly, the inspection frequencies remain constant on certain parts of the graph (the northern east-west axis and the most southern edges), where the volume of traffic remains high all day long. The equilibrium is such that drivers using these edges have an incentive to pay the toll at all times; the inspection frequencies depicted on the figure correspond to a point where the users of the major commodities have the same loss if they choose fare evasion or fare compliance (so they will break the tie in favour of the controller by paying the toll in our Stackelberg game model). Controlling more on this northern axis would be a waste of inspection capacity, and controlling less would lead to fare

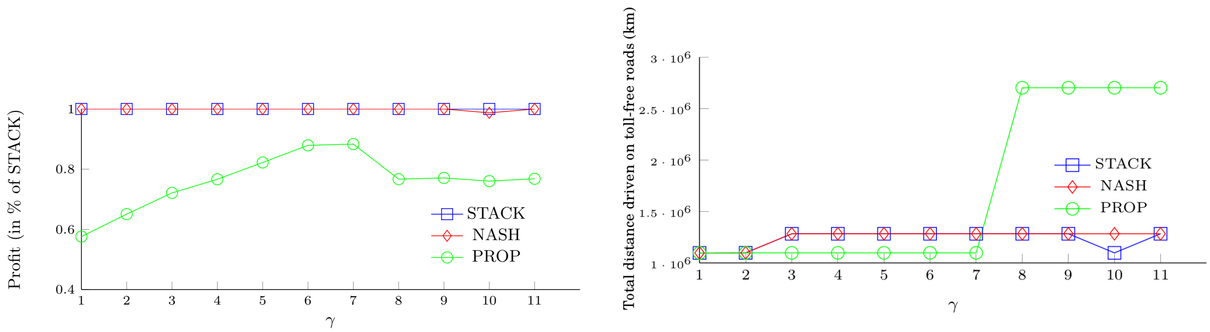


Figure 6: Evolution of the total profit and number of kilometers driven on the alternative toll-free trunk roads with γ, for the Rhineland-Palatinate instance in the MAXPROFIT setting.

evasion on this axis. The remaining inspectors are allocated to sections and time windows with less traffic.

6 Conclusion and Perspectives

We have introduced the class of network spot-checking games, which generalizes the class of security games, and can be used to represent many situations in which controls must be distributed over a network. We have proposed a LP / MIP based approach to compute Nash and Stackelberg equilibria of this game. Users' strategies are represented by flows in the network, which makes it possible to compute a Nash equilibrium of the game very efficiently. Our experiments suggest that this "Nash strategy" approximates a Stackelberg equilibrium maximizing the inspectors' profit. This is a feature which we would like to investigate in a follow-up work: given a class $\mathcal{C}$ of NSC games, how far can the payoff of the inspector for a Nash Strategy be from the optimum payoff obtained with a Stackelberg strategy ? This question is related to the *price of anarchy* $\mathrm{PoA}(\mathcal{C})$ of the class $\mathcal{C}$, and we would like to bound this value from above for certain class of games (for example, games with certain bounds on the edge weights $\boldsymbol{w}, \boldsymbol{\beta}$ and $\boldsymbol{\sigma}$).

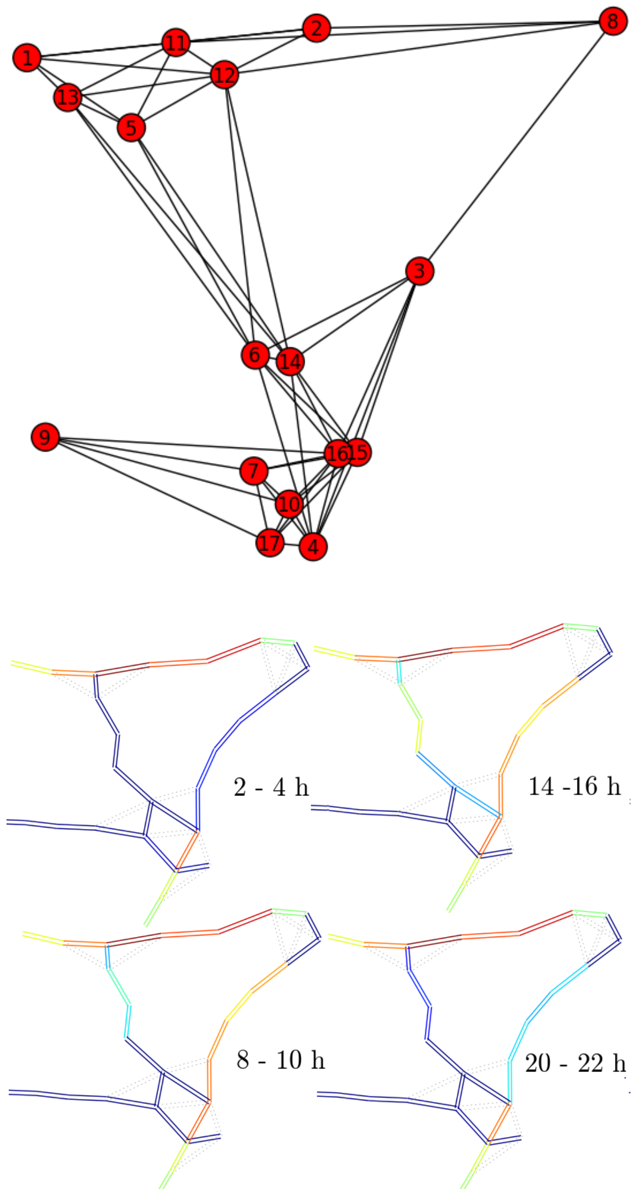


Figure 7: Graph C joining the control areas for the Saxony-Anhalt instance (above); Nash strategies of the controller during four time windows, computed for $\gamma = 20$ inspector teams (below). Red/blue indicates higher/lower inspection frequencies.

References

[1] R. Borndörfer, J. Buwaya, G. Sagnol, and E. Swarat. Optimizing toll enforcement in transportation networks: a game-theoretic approach. In *Proceedings of the INOC 2013, ENDM*, volume 41, pages 253–260, Tenerife, Spain, 2013.

[2] R. Borndörfer, B. Omont, G. Sagnol, and E. Swarat. A Stackelberg game to optimize the distribution of controls in transportation networks. In *Proceedings of GAMENETS 2012, LNICST*, volume 105, pages 224–235, Vancouver, Canada, 2012.

[3] R. Borndörfer, G. Sagnol, and E. Swarat. An IP approach to toll enforcement optimization on german motorways. In *Operations Research Proceedings 2011*, pages 317–322. Springer, 2011.

[4] C. Daskalakis and C. Papadimitriou. On a network generalization of the minmax theorem. *Automata, Languages and Programming*, pages 423–434, 2009.

[5] M. Jain, E. Kardes, C. Kiekintveld, F. Ordóñez, and M. Tambe. Security games with arbitrary schedules: A branch and price approach. In *Proc. of The 24th AAAI Conference on Artificial Intelligence*, pages 792–797. AAAI Press, 2010.

[6] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 689–696. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[7] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. nash in security games: an extended investigation of interchangeability, equivalence, and uniqueness. *J. Artif. Int. Res.*, 41(2):297–327, May 2011.

[8] P. Paruchuri, J.P. Pearce, J. Marecki, M. Tambe, F. Ordóñez, and S. Kraus. Playing games for security: an efficient exact algorithm for solving bayesian stackelberg games. In *Proc. of the 7th AAMAS*, pages 895–902. ACM, 2008.

[9] J. Pita, M. Jain, F. Ordóñez, M. Tambe, S. Kraus, and R. Magori-Cohen. Effective solutions for real-world stackelberg games: When agents must deal with human uncertainties. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 369–376. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[10] Z. Yin, A. Jiang, M. Johnson, M. Tambe, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, and J. Sullivan. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems. In *Proc. of The 24th Conference on Innovative Applications of Artificial Intelligence (IAAI)*, 2012.

A Relation between Security Games and NSC Games

The class of *security games* has been formally introduced by Kiekintveld et. al. [6], and was at the origin of a series of papers (see e.g. [5, 7, 9, 10]). We shall see that under a certain condition, a security game can be cast as an NSC game. Moreover, this condition can be dropped if we use a class of extended NSC games, whose definition relies on a vector parameter $\boldsymbol{\alpha} \in \mathbb{R}^E$ instead of the scalar $\alpha \in [0, 1]$.

A security game is defined between an *attacker* and a *defender*. There is a set of *targets* $T = \{t_1, \ldots, t_n\}$ which may be attacked. The mixed strategy of the attacker is a probability distribution $\boldsymbol{p}$ over $\{1, \ldots, n\}$. The pure strategies for the defender are given by a set of schedules $\mathcal{S} = \{S_1, \ldots, S_m\}$ (a schedule S_i is a subset of targets that can be covered by some defender's resources), and there is a linear function φ which maps a mixed strategy of the controller $\tilde{\boldsymbol{q}}$ (i.e. a probability distribution over $\{1, \ldots, m\}$) to a vector $\boldsymbol{q} = \varphi(\tilde{\boldsymbol{q}})$, that gives the marginal probability q_t that a target t is covered by some defender's resource. The defender's payoff $\Pi_d(\boldsymbol{p}, \boldsymbol{q})$ is defined through the quantities $U_d^c(t)$ and $U_d^u(t)$, which represent respectively the payoffs if t is attacked while being covered by some resource or uncovered:

$$\Pi_d(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=1}^{n} p_i[U_d^c(t_i)q_i + U_d^u(t_i)(1 - q_i)].$$

Similarly, the quantity $U_a^c(t)$ (resp. $U_a^u(t)$) defines the attacker's payoff when the target t is covered (resp. uncovered):

$$\Pi_a(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=1}^{n} p_i[U_a^c(t_i)q_i + U_a^u(t_i)(1 - q_i)].$$

It is natural to assume that the quantities $\Delta U_a(t_i) = U_a^u(t_i) - U_a^c(t_i)$ and $\Delta U_d(t_i) = U_d^c(t_i) - U_d^u(t_i)$ are nonnegative (cf. [7]).

We next show that if the ratio

$$\alpha_i = \frac{\Delta U_d(t_i)}{\Delta U_a(t_i)}$$

takes the same value $\alpha \leq 1$ for every target t_i, then the security game defined above can be cast as a NSC game. This condition is related to our assumption that the factor $\alpha \leq 1$ that represents the fraction of fines entering the controller's payoff does not depend on the edges where evaders get fined. It would be straightforward however to define an extended NSC game with a different factor $\alpha_e \in \mathbb{R}$ for every edge $e \in E$.

We create the graph $G = (V, E)$ by connecting a set of vertices $\{t_1, \ldots, t_n\}$ to a start depot node s and a terminal node d:

$$V = \{s, d, t_1, \ldots, t_n\},\ E = \{(s, t), (t, d) :\ t \in T\}.$$

Define $\delta = \max_{t\in T} U_a^u(t)$ and $P = \max_{t\in T} \Delta U_a(t)$. For all $t \in T$, define $w_{(s,t)} = \delta - U_a^u(t) \geq 0$, $\sigma_{(s,t)} = P^{-1}\Delta U_a(t) \in [0, 1]$, $\beta_{(t,d)} = U_d^u(t)$, and $w_{(t,d)} = \sigma_{(t,d)} = \beta_{(s,t)} = 0$. We consider a single commodity $\mathcal{K} = \{(s, d)\}$ on this graph, and we call Player 1 the unique user travelling on this commodity. His mixed strategy can be represented by a probability vector $\boldsymbol{p}$ indicating the probability p_i that he chooses the $(s, d)-$path through t_i. We assume that the controller inspects the section (s, t_i) whenever the defender is covering target t_i in the original security game, so that the vector of marginal inspection rates lies in the polyhedron

$$\begin{aligned}\mathcal{Q} = \{\hat{\boldsymbol{q}} \in [0,1]^E : \exists(\boldsymbol{q}, \tilde{\boldsymbol{q}}) \in [0,1]^n \times [0,1]^m :\\ \sum_{j=1}^m \tilde{q}_j = 1, \quad \boldsymbol{q} = \varphi(\tilde{\boldsymbol{q}}),\\ \hat{q}_{(s,t_i)} = q_i, \quad \hat{q}_{(t_i,d)} = 0\}.\end{aligned}$$

The payoff of Player 1 can be written as follows, see (5):

$$\begin{aligned}\text{Payoff}_1(\boldsymbol{p}, \boldsymbol{q}) &= -\sum_{i=1}^n p_i[\sigma_{(s,t_i)}\hat{q}_{(s,t_i)}P + w_{(s,t_i)}]\\ &= -\sum_{i=1}^n p_i[\Delta U_a(t_i)q_i + \delta - U_a^u(t_i)]\\ &= -\sum_{i=1}^n p_i[(U_a^u(t_i) - U_a^c(t_i))q_i + \delta - U_a^u(t_i)]\\ &= \underbrace{\sum_{i=1}^n p_i[U_a^c(t_i)q_i + U_a^u(t_i)(1-q_i)]}_{\Pi_a(\boldsymbol{p},\boldsymbol{q})} - \delta.\end{aligned}$$

Similarly, the payoff of the controller rewrites

$$\begin{aligned}\text{Payoff}_C(\boldsymbol{p}, \boldsymbol{q}) &= \sum_{i=1^n} p_i[\alpha\Delta U_a(t_i)q_i + U_d^u(t)]\\ &= \sum_{i=1^n} p_i[\Delta U_d(t_i)q_i + U_d^u(t)]\\ &= \underbrace{\sum_{i=1}^n p_i[U_d^c(t_i)q_i + U_d^u(t_i)(1-q_i)]}_{\Pi_d(\boldsymbol{p},\boldsymbol{q})}.\end{aligned}$$

This shows that the present NSC game is equivalent to the generic security game presented above (up to a constant term $-\delta$ in the Player 1's payoff).

Conversely, we point out that every NSC game can be reformulated to a Bayesian security game, i.e. a security game in which the defender faces simultaneously several types of attacker, see e.g. [8]. By doing so, however, we loose the compactness of the representation of the NSC games, since all paths through the commodities must be enumerated.