

RALF BORNDÖRFER, JULIA BUWAYA,
GUILLAUME SAGNOL, ELMAR SWARAT

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

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 spot-checking games (SC games). These games model problems where the goal is to distribute fare inspectors over a toll network. In an SC game, the pure strategies of network users correspond to paths in a graph, and the pure strategies of the inspectors are subset of arcs to be controlled. Although SC games are not zero-sum, we show that a Nash equilibrium can be computed by linear programming. The computation of a strong Stackelberg equilibrium is more relevant for this problem and we give a mixed integer programming (MIP) formulation for this problem. We show that the computation of such an equilibrium is NP-hard. More generally, we prove that it is NP-hard to compute a strong Stackelberg equilibrium in a polymatrix game, even if the game is pairwise zero-sum. Then, we give some bounds on the *price of spite*, which measures how the payoff of the inspector degrades when committing to a Nash equilibrium. Finally, we report computational experiments on instances constructed from real data, for an application to the enforcement of a truck toll in Germany. These numerical results show the efficiency of the proposed methods, as well as the quality of the bounds derived in this article.

Keywords Game Theory; Stackelberg Equilibrium; Security Games; Mixed Integer Programming; Price of Spite; Nash Equilibrium; Inspection

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), which has the task to carry

*{borndorfer,buwaya,sagnol,swarat}@zib.de

[†]This work was funded by the German federal office for Good Transport (BAG).

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 transportation networks, in order to enforce the payment of a transit toll. In a previous version of this article which appeared as a conference paper [1], we used a Stackelberg game to represent the applied problem evoked above. This article deals with a class of games called *spot-checking* (SC) games that generalize our former approach. SC games can be thought as particular *security games*, in which the strategies of *attackers* and *defenders* are represented by a graph structure. Security games have been introduced by Kiekintveld et al. [10], to study problems where the goal is to randomize different kind of inspections, in a strategic way; this includes work on the optimal selection of checkpoints and patrol routes to protect the LA Airport towards adversaries [13], a study of the scheduling and allocation of air marshals to a list of flights in the U.S. [8], or the problem of optimally scheduling fare inspection patrols in the LA Metro [18]. In a way, SC games are to security games as *routing games* (in which players choose a path on a network) are to *congestion games* (where players choose a subset of abstract congestible elements); see e.g. Roughgarden and Tardos [15]. Many practical situations can be represented by this new class of games, as will be seen in Section 4. In particular, it is also possible to take into account the moves of inspectors over the network over time.

Unlike standard security games, the players of a 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 [18], or assumed that each user takes the shortest path [2]. A similar network security game was studied in [8], but the authors used a branch-and-price approach to deal with the exponential number of actions. We represent the mixed strategies of network users by multicommodity flows, which yields a compact linear programming formulation for the computation of a Nash equilibrium; see Section 5.1. However, we argue that the inspector should rather commit to a Stackelberg equilibrium, but this strategy is much harder to compute. A mixed integer programming (MIP) formulation is given in Section 5.2.

Indeed, we show that the computation of a Stackelberg equilibrium is NP-hard in Theorem 6.1. As a corollary, we obtain that it is NP-hard to find a Stackelberg equilibrium in *pairwise zero-sum polymatrix games*. The principal novelty of this article concerns the *price of spite* of SC games, and is presented

in Section 6.2. We give two bounds on this quantity –which is a measure of how the payoff of the inspector degrades when committing to a Nash equilibrium (rather than a Stackelberg equilibrium)– as detailed in Proposition 6.5 and Theorem 6.7. We point out that the gap between the Stackelberg and the Nash equilibria has already been investigated in other works: conditions under which the gap is zero are studied in [5], and [12] bounds this gap for routing games.

Finally, we conduct numerical experiments and demonstrate the quality of these bounds for the application to German motorways in Section 7.

2 Preliminaries and Notation

We first recall some basic notions of game theory. In a game with N players where each player may choose a strategy \mathbf{p}_i in a set Δ_i , and wishes to maximize her own payoff $u_i(\mathbf{p}_i, \mathbf{p}_{-i})$, we say that \mathbf{p}_i is a best response to the set $\mathbf{p}_{-i} \equiv \{\mathbf{p}_1, \dots, \mathbf{p}_{i-1}, \mathbf{p}_{i+1}, \dots, \mathbf{p}_N\}$ of strategies of the other players if

$$\forall \mathbf{p}'_i \in \Delta_i, \quad u_i(\mathbf{p}'_i, \mathbf{p}_{-i}) \leq u_i(\mathbf{p}_i, \mathbf{p}_{-i}).$$

We denote by $BR_i(\mathbf{p}_{-i}) \subseteq \Delta_i$ the set of best responses to \mathbf{p}_{-i} for Player i . In the standard setting, the set Δ_i is a probability simplex $\{\mathbf{p} \geq 0 : \sum_i p_i = 1\}$. In this article, we allow Δ_i to be polyhedral for convenience, but one could easily come back to the standard case by expressing $\mathbf{p} \in \Delta_i$ as a convex combination of the extreme points of Δ_i .

The best responses allow us to define the *Nash equilibria* (NE) of the game:

$$(\mathbf{p}_1, \dots, \mathbf{p}_N) \text{ is a Nash equilibrium} \iff \forall i \in \{1, \dots, N\}, \quad \mathbf{p}_i \in BR_i(\mathbf{p}_{-i}).$$

In this paper we also study 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 similar applications [8, 13, 18]. In a Stackelberg game, it is assumed that a player is the leader (in our case, the inspector), who plays first, and the other players (called followers) react with a best response. Assume now that there are $N + 1$ players labelled from 0 to N , and that Player 0 is the leader; a *strong Stackelberg equilibrium* (SSE) is defined as a profile of strategies $\mathbf{p} \equiv (\mathbf{p}_0, \dots, \mathbf{p}_N)$ that maximizes the leader's payoff, among the set of all profiles such that the followers' strategies \mathbf{p}_{-0} are in best response relationship to each other's action:

$$\mathbf{p}_{\text{eq}} \text{ is a strong Stackelberg equilibrium} \iff \mathbf{p}_{\text{eq}} \in \arg \max_{\mathbf{p} \in \{\Delta_0 \times \dots \times \Delta_N : \forall i \neq 0, \mathbf{p}_i \in BR_i(\mathbf{p}_{-i})\}} u_0(\mathbf{p}_0, \mathbf{p}_{-0}).$$

In fact, in the class of spot-checking games that we define hereafter, $BR_i(\mathbf{p}_{-i})$ depends only on the leader's strategy \mathbf{p}_0 for every follower ($i \neq 0$) because we neglect congestion effects; see Proposition 3.1. We can hence denote by $BR(\mathbf{p}_0) := BR_1(\mathbf{p}_0) \times \dots \times BR_N(\mathbf{p}_0)$ the set of best response profiles for the

followers to a leader's action \mathbf{p}_0 , and the problem of finding an equilibrium reduces to the following optimization problem:

$$\max_{\mathbf{p}_0 \in \Delta_0} \max_{\mathbf{p}_{-0} \in BR(\mathbf{p}_0)} u_0(\mathbf{p}_0, \mathbf{p}_{-0}). \quad (1)$$

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

We will also evoke polymatrix games in this article. A game is a polymatrix game if every payoff function can be written as

$$u_i(\mathbf{p}_i, \mathbf{p}_{-i}) = \sum_{j \neq i} \mathbf{p}_i^T A_{ij} \mathbf{p}_j \quad (2)$$

for some matrices A_{ij} . A polymatrix game is called *pairwise zero-sum* if $A_{ij} = -A_{ji}^T$ for all $i \neq j$. We refer the reader to [4] for more background on this topic.

Notation We next introduce the notation used in this article. When a directed graph $G = (V, E)$ and a list of weighted commodities $\mathcal{C} := \{(s_k, d_k, x_k), k \in \mathcal{K}\}$ are given, F_k denotes the set of (s_k, d_k) -flows of unit value: for all $\mathbf{f} \in \mathbb{R}_+^{|E|}$, $\mathbf{f} \in F_k$ if and only if

$$\sum_{e' \in \delta^+(v)} f_{e'} - \sum_{e \in \delta^-(v)} f_e = \begin{cases} 1 & \text{if } v = s_k; \\ -1 & \text{if } v = d_k; \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

for all vertices $v \in V$, where $\delta^+(v)$ and $\delta^-(v)$ respectively denote the set of outgoing and incoming arcs at v . We also denote by $F(\mathcal{C})$ the set of multi-commodity flows satisfying a demand x_k for commodity k , that is, $F(\mathcal{C}) = \{\sum_{k \in \mathcal{K}} x_k \mathbf{f}_k : \forall k, \mathbf{f}_k \in F_k\}$. Let z_e represent the weight of an arc $e \in E$. We denote by $\text{SP}_k(\mathbf{z})$ the set of all (s_k, d_k) -shortest paths. Their length is denoted by $\text{spl}_k(\mathbf{z}) = \sum_{e \in R} z_e$ (the expression does not depend on the choice of $R \in \text{SP}_k(\mathbf{z})$).

The elementwise (Hadamard) product of two vectors \mathbf{u} and \mathbf{v} is $\mathbf{u} \circ \mathbf{v}$, so that $(\mathbf{u} \circ \mathbf{v})_i = u_i v_i$.

3 Spot-Checking Games

In this section we define the class of spot-checking games studied in this article. They generalize our approach [1, 2], and describe the interaction between the *users* of a toll network and an *inspector* (or Player 0), who represents all fare inspectors of the transportation network.

A *spot-checking game* (SC game) $\mathcal{G} = (G, \mathcal{C}, \mathcal{W}, \mathcal{Q}, \alpha)$ is formally defined by the following elements:

- A directed graph $G = (V, E)$;

- A set of weighted commodities $\mathcal{C} := \{(s_k, d_k, x_k), k \in \mathcal{K}\}$, representing origin-destination pairs $(s_k, d_k) \in V^2$ with $x_k > 0$ users;
- A set of arc weights $\mathcal{W} := \{(w_e, \beta_e, \sigma_e), e \in E\}$, representing:
 - the total cost $w_e \geq 0$ for a user taking arc e ;
 - a reward β_e (resp. a penalty if $\beta_e < 0$) for the inspector for each user using arc e ; this β_e typically corresponds to a fare for taking arc e ;
 - The expected amount $\sigma_e \geq 0$ of penalty for a user controlled on e . This can be interpreted as the amount of a fine, multiplied by the probability to be controlled on e (conditionally to the presence of an inspector on e);
- A set $\mathcal{Q} \subset [0, 1]^{|E|}$ described by linear inequalities, representing possible distributions of the inspectors over the arcs of the graph. The quantity q_e corresponds to the probability that some inspector is present on arc e ;
- A fraction $\alpha \in [0, 1]$ of penalties to be considered in the inspector's payoff.

At first sight, it may seem odd that every controlled user gets fined. The reason behind is that the strategy of any user is completely represented by his path in G . In fact, the graph G is obtained after a pre-processing of the true transportation network G_0 , so that the path chosen by a given user indicates whether he is paying or evading the fare; see Section 4.

Typically, every arc of the network is either a *pay-arc*, where $\sigma_e = 0$ and the reward $\beta_e > 0$ corresponds to a toll fare, which is included in the user costs ($w_e \geq \beta_e$). Or the arc is an *evasion arc*, where the reward is $\beta_e \leq 0$ and the expected fine is $\sigma_e > 0$. In other words, *paying a fare* and *getting fined* exclude each other. We refer the reader to Section 4 for more details and examples on how to model practical situations using SC games.

Inspector's strategy In the simplest variant of the problem, there are γ teams of controllers over the network, who can each control an arc $e \in E$. The set \mathcal{S} of pure strategies for the *inspector* hence corresponds to the subsets S of E of cardinality γ . The inspectors commit to a mixed strategy $\tilde{\mathbf{q}}$, 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 arcs $e \in S$. 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 team of inspectors is present on arc e . It is easy to see that the marginal strategy \mathbf{q} satisfies

$$\sum_{e \in E} q_e = \gamma, \quad (4)$$

$$\forall e \in E, 0 \leq q_e \leq 1. \quad (5)$$

Conversely, if we are given a vector \mathbf{q} satisfying Relations (4) and (5), we can find a mixed strategy $\tilde{\mathbf{q}}$ whose marginal equals \mathbf{q} . To see this, one can notice that the set of extreme points of the polyhedron \mathcal{Q} defined by Relations (4)

and (5) coincides with the set of pure strategies \mathcal{S} . More evolved strategy sets \mathcal{Q} for the inspector, taking into account the moves of controllers over time, will be discussed in Section 4.3; see also Remark 4.2 for a discussion on the use of Markov patrolling policies [9].

User flows and payoffs For a given inspection strategy $\mathbf{q} \in \mathcal{Q}$, the loss of every user who selects a path R over the graph G is $\sum_{e \in R} w_e + \sigma_e q_e$, where the first term accounts for travel and toll costs, while the second term is the expected fine. Note that with this formula, we assume that evaders can 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 approximation has been used in [2, 18].

We associate the users of commodity k with a single player (called Player k). Her mixed strategy represents the distribution of k -users over the (s_k, d_k) -paths, and forms a flow $\mathbf{p}^k \in F_k$. In the terminology of routing games, this is a model of atomic splittable traffic (each player controls a non-vanishing fraction of the flow, which may be split among multiple links [14]); however, since we neglect congestion effects, see Remark 3.2, our model is essentially equivalent to non-atomic traffic. We denote by $\mathbf{p} = \sum_k x_k \mathbf{p}^k \in F(\mathcal{C})$ the multicommodity flow of all network users. The payoff of Player k is

$$\text{Payoff}_k(\mathbf{p}, \mathbf{q}) := - \sum_{e \in E} p_e^k (w_e + \sigma_e q_e). \quad (6)$$

The total payoff of the inspector is obtained by summing the collected rewards and penalties, and depend on the parameters α and β of the game \mathcal{G} :

$$\begin{aligned} \text{Payoff}_0(\mathbf{p}, \mathbf{q}) &:= \sum_k x_k \sum_{e \in E} p_e^k (\beta_e + \alpha \sigma_e q_e) \\ &= \sum_{e \in E} p_e (\beta_e + \alpha \sigma_e q_e). \end{aligned} \quad (7)$$

The extreme values of α correspond to two important situations. If $\alpha = 1$, the payoff defined in (7) corresponds to the total revenues from rewards and penalties, a setting which we denote by MAXPROFIT. If $\alpha = 0$, the inspector's payoff comes from the fares only (assuming that the reward β_e is a fare for arc 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 bi-objective problem.

Users' best responses Observe that the payoff of any user does not depend on the strategy of other users (because we ignore congestion effects). An important consequence of Equation (6) is a characterization of the best responses of

Player k to the inspector's strategy \mathbf{q} , in terms of particular shortest paths in G . We denote by $BR(\mathbf{q}) \subseteq F(\mathcal{C})$ the set of users' best response multicommodity flows obtained thereby.

Proposition 3.1. *Let $\mathbf{q} \in \mathcal{Q}$ be a strategy of the inspector. A strategy \mathbf{p}^k for Player k is a best response to \mathbf{q} if and only if $-\text{Payoff}_k(\mathbf{p}, \mathbf{q}) = \text{spl}_k(\mathbf{w} + \boldsymbol{\sigma} \circ \mathbf{q})$. In other words, best responses for Player k are flows supported by $\text{SP}_k(\mathbf{w} + \boldsymbol{\sigma} \circ \mathbf{q})$.*

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

Remark 3.2. We plan to investigate an extension of SC games with congestion in a follow-up work. Note that the number of agents using arc e can have an influence on both the travel time (and hence the costs w_e) and the probability to be controlled (and hence the expected penalties σ_e). However, we think that in most transportation networks, it is a reasonable assumption that a large fraction of *honest users* is always fare-compliant, regardless of the inspection rate. Thus, we expect the congestion to be caused mostly by users that do not even take part in the spot-checking game, and hence we expect the distribution of controls over the network to have a limited impact on the congestion only. This justifies the uncongested approach of the present article, which should rather be seen as a case of constant congestion.

4 Handling Practical Situations with SC 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.

4.1 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 SC game. Here, the network users are assumed to travel over a network $G_0 = (V_0, E_0)$ with arc costs w_e and expected penalties σ_e (conditionally to the presence of a controller on arc e). For a given commodity $(s_k, d_k, x_k) \in \mathcal{C}_0 \subset V_0 \times V_0 \times \mathbb{R}_+$ ($k \in \mathcal{K}$), Player k can either decide to pay a fare τ_k (and in this case she will take the shortest path from s_k to d_k), or she 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 vertex \bar{s} for every source $s \in S = \{s_k : k \in \mathcal{K}\}$ and a vertex \bar{d} for every destination $d \in D = \{d_k : k \in \mathcal{K}\}$.

We then connect these new vertices by creating a set of dummy arcs \bar{E} , that contains

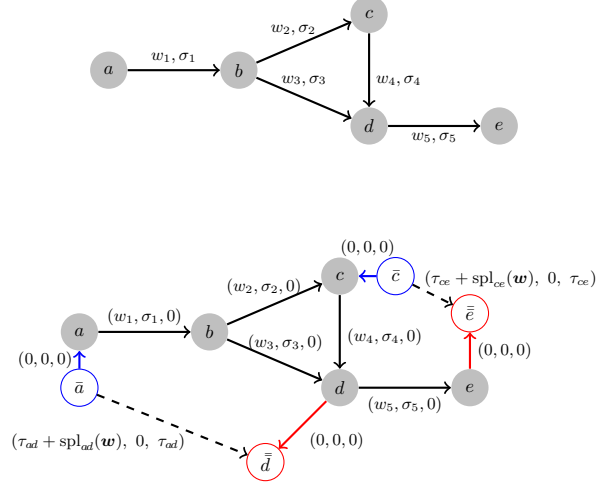


Figure 1: Original transit network G_0 (above), and extended structure G of the SC game (below). The arcs of G are labelled with triples (w_e, σ_e, β_e) . The list of commodities is $\mathcal{C}_0 = \{(a, d, x_{ad}), (c, e, x_{ce})\}$, which becomes $\mathcal{C} = \{(\bar{a}, \bar{d}, x_{ad}), (\bar{c},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

- an arc $\bar{s} \rightarrow s$ 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 arc $d \rightarrow \bar{d}$ for all $d \in D$, with all weights equal to zero as well;
- an arc $\tilde{e}_k = (\bar{s}_k, \bar{d}_k)$ for all $k \in \mathcal{K}$, which corresponds to the action of paying the fare. The user cost for taking this arc is defined by $w_{\tilde{e}_k} = \text{spl}_k(\mathbf{w}) + \tau_k$, the inspector's reward is $\beta_{\tilde{e}_k} = \tau_k$, and users cannot be fined on this arc ($\sigma_{\tilde{e}_k} = 0$).

By construction, the dummy toll arc $\tilde{e}_k = (\bar{s}_k, \bar{d}_k)$ can only be used by Player k . The original arcs $e \in E_0$ are taken by users evading the fare, so that the inspector's reward β_e can be set to 0 on these arcs; alternatively, β_e can be set to a negative value to penalize fare evasion in the inspector's payoff. (It is also possible to penalize evasion on a per-user basis, by setting a negative reward on the dummy evasion arcs $s \rightarrow \bar{s}$.)

The SC 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 $\mathcal{C} = \{(\bar{s}_k, \bar{d}_k, x_k) : k \in \mathcal{K}\}$. An example graph G_0 is represented together with its extension G and the associated arc weights in Figure 1.

4.2 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. Indeed on the German motorways the transit fee is proportional to the distance. 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 $\mathcal{C}_0 := \{(s_k, d_k, x_k) : k \in \mathcal{K}\}$ be a list of weighted commodities. For all $k \in \mathcal{K}$, s_k and d_k represent the origin and the destination of k . We denote by l_e the length of arc $e \in E_0$ and by f (resp. b) the toll rate (resp. the average basic costs such as fuel consumption) per kilometer. In addition, we assume that the expected penalty to pay on arc $e \in E_0$ 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 vertices, containing a vertex \bar{s} for every $s \in S = \{s_k : k \in \mathcal{K}\}$ and a vertex \bar{d} for every $d \in D = \{d_k : k \in \mathcal{K}\}$. To connect these new vertices, we create the following sets of arcs:

- $E_1 = \{(u', v') : \forall (u, v) \in E_0\};$
- $E_s = \{(\bar{s}, s), (\bar{s}, s') : \forall s \in S\};$
- $E_d = \{(d, \bar{d}), (d', \bar{d}) : \forall d \in D\};$
- $E_t = \{(v, v'), (v', v) : \forall v \in V_0\}.$

The arcs of level E_1 represent portions of a trip where the toll fee has been paid, while the matching arcs in E_0 correspond to fare evasion. Transition arcs $e \in E_t$ 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 arcs of E_s and E_d ensure the connectivity of the new commodities $\mathcal{C} := \{(\bar{s}_k, \bar{d}_k, x_k) : k \in \mathcal{K}\}$ with both layers. To sum up, the different arc weights are defined by

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

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

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

Remark 4.1. The transit model of Section 4.1 can basically be seen as the limit of the present two-layer model when the transition cost θ goes to ∞ , so no player has an incentive to switch between *paying* and *evading* in the middle of her trip.

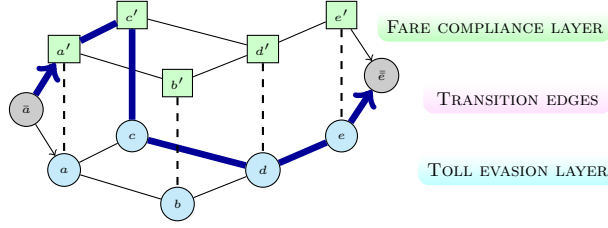


Figure 2: Example of a two-layer graph with a single commodity $\mathcal{K} = \{(a, e, x_{ae})\}$. The highlighted path indicates a user paying the fare on $a \rightarrow c$, and evading the toll on $c \rightarrow d \rightarrow e$.

4.3 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 arcs of the networks and their duties must not exceed a certain length. In consequence, the set \mathcal{Q} defined by Relations (4) and (5) might not be well suited to represent all possible marginal strategies of the inspector.

The authors of [18] have proposed to represent the duties of metro ticket inspectors by flows in an appropriate graph. Their approach provides exact schedules for each inspector. More precisely, a strategy consists of a sequence of trains that the inspector 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) allocate the inspectors to a set of predefined control areas, which they must patrol during a given time interval [3]. The graph structure which we next present combines ideas of [3] and [18].

We consider a time discretization $\mathcal{T} = \{0, \dots, T-1\}$ of the period of interest, typically one day, and we make the simplifying assumption that every network user starts and ends his trip within the same time 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 § 4.1 or § 4.2), and we make a time-expanded 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 vertices $(s_k, d_k) \in V^2$, such that $s_k = (u, t)$ and $d_k = (v, t)$ for a pair of vertices $(u, v) \in V_0^2$ and a time window $t \in \mathcal{T}$.

A control area $S \in \mathcal{S}$ consists of a subset of arcs $S \subset E_0$ (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-expanded version $D = (\mathcal{S} \times \mathcal{T}, \bar{A})$

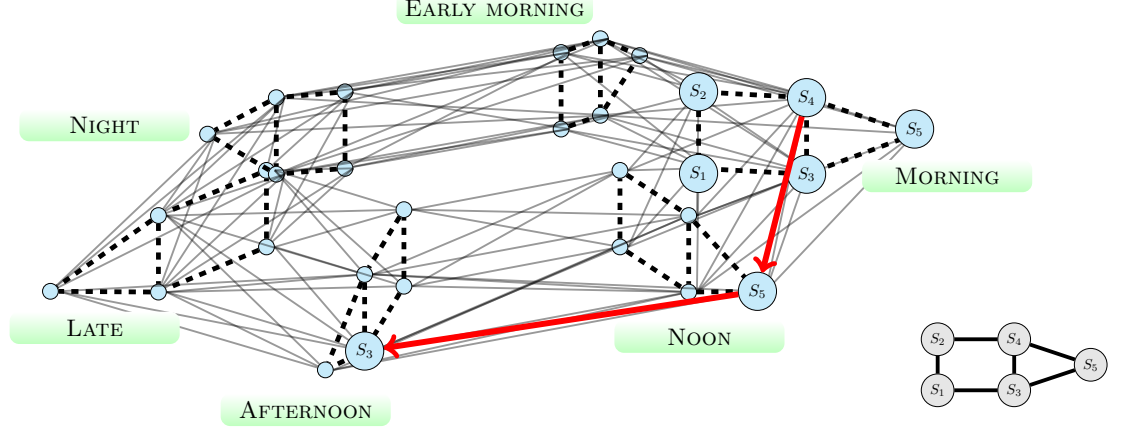


Figure 3: Example for a graph C connecting the control areas (lower right corner) and its associated cyclic duty graph D (main drawing), for a time discretization of one day with $T = 6$ time windows. The highlighted path represents the duty of a team controlling S_4 during the morning, S_5 at noon and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

of C , which we call the *cyclic duty graph*, as follows:

$$\bar{A} = \left\{ ((S, t), (S, t + 1 \bmod T)) : \forall S \in \mathcal{S} \right\} \cup \left\{ ((S, t), (S', t + 1 \bmod T)) : \forall (S, S') \in A \right\}.$$

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 vertices 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 [18] for details about this construction.

Now, we assume that there are γ teams of inspectors, as in the paragraph preceding Equation (4). The inspector's strategy can hence be represented by a (d_s, d_t) -flow \tilde{q} of value γ in $\tilde{D} = (\tilde{V}, \tilde{A})$:

$$\forall v \in \tilde{V}, \quad \sum_{a' \in \delta^+(v)} \tilde{q}_{a'} - \sum_{a \in \delta^-(v)} \tilde{q}_a = \begin{cases} \gamma & \text{if } v = d_s; \\ -\gamma & \text{if } v = d_t; \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

The vertex set of \tilde{D} is

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

and similar to [18] it can be seen that the expected number of inspectors in the control area $S \in \mathcal{S}$ at time t is

$$\hat{q}_{(S,t)} = \sum_{l=1}^L \sum_{a \in \delta^-(S,t,l)} \tilde{q}_a. \quad (9)$$

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

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

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

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

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 arc $(e,t) \in E$ of the time-expanded graph G . To summarize, the set of marginal strategies \mathcal{Q} of the inspector can be defined by:

$$\begin{aligned} \mathcal{Q} = \Big\{ \mathbf{q} \in (\mathbb{R}_+)^{|E|} : \exists \tilde{\mathbf{q}} \in (\mathbb{R}_+)^{|\tilde{A}|} \text{ s.t.} \\ (i) \ \tilde{\mathbf{q}} \text{ satisfies the flow conservation (8);} \\ (ii) \ \forall (e,t) \in E, \ q_{(e,t)} \leq \sum_{\{S \in \mathcal{S}: S \ni e\}} \kappa_{e|S} \sum_{l=1}^L \sum_{a \in \delta^-(S,t,l)} \tilde{q}_a; \\ (iii) \ \forall (e,t) \in E, \ q_{(e,t)} \leq 1 \Big\}. \end{aligned}$$

To conclude this section, we briefly mention some simple extensions that can be incorporated into 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 all controllers do not start from the same location in the network, it is possible to consider several start and end depot vertices 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 fewer controls) could be represented by adding arcs in G that link the different time copies of G_0 , with a cost ς for the delay.

Remark 4.2. A more advanced possibility would be to use a set \mathcal{Q} of Markov patrolling policies, as proposed e.g. in [9, 17]. The idea is to represent the location of inspectors (in space and time) as states of a Markov Decision Process (MDP). In particular, it is possible to add special states in the MDP to represent the fact that inspectors can be delayed, with a fixed rate of unexpected events. This allows one to use a very fine time discretization. Then, feasible controls $\mathbf{q} \in \mathcal{Q}$ correspond to transition probabilities in the MDP, and can be represented by flows in a transition graph.

We shall now return to the general model of SC games introduced in Section 3, and we study the problem of computing some equilibria for a generic game $\mathcal{G} = (G, \mathcal{C}, \mathcal{W}, \mathcal{Q}, \alpha)$.

5 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 [11]. However, there is a natural interpretation for the Stackelberg strategies of the inspector: the Stackelberg game model assumes that every user of the network plays with a best response to the inspector's strategy \mathbf{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 inspector'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, users could decide to walk), thus depriving the inspector from all sources of profit. However, there is no reason to assume that network users want to minimize the inspector's payoff, and the Stackelberg approach guards ourselves from crafty behaviours.

Remark 5.1. The optimization problems studied in this section solely focus on the computation of an optimal control strategy $\mathbf{q} \in \mathcal{Q}$. Of course, other parameters of the problems could be used as decision variables, too, without making the problem substantially harder. For example, it seems natural to optimize the quantity of resources γ used for the controlling task, cf. Eq. (4). This can be done by adding a cost term $-c\gamma$ in the objective functions. Similarly, it is also possible to optimize the toll fares, and we could imagine a suitable criterion to select the value of the penalty. However, these variables are usually not under the control of the same decision-maker. In the case of the application to German motorways, the decision of a toll fare per kilometer is highly political, and our partner does not wish to optimize the number of fare inspectors γ to hire.

5.1 Nash equilibria

We next show that the SC game \mathcal{G} can be transformed into a *zero-sum game* that has the same Nash equilibria.

Proposition 5.2. *Let $\mathcal{G} = (G, \mathcal{C}, \{\mathbf{w}, \beta, \sigma\}, \mathcal{Q}, \alpha)$ be a SC game. If $\alpha > 0$, then \mathcal{G} has the same set of NE as the zero-sum game $\mathcal{G}' = (G, \mathcal{C}, \{\mathbf{w}, \mathbf{w}, \sigma\}, \mathcal{Q}, 1)$, where the inspector's rewards β_e have been replaced by the arc costs w_e and $\alpha = 1$. If $\alpha = 0$, (\mathbf{p}, \mathbf{q}) is a NE for all $\mathbf{q} \in \mathcal{Q}$ and $\mathbf{p} \in BR(\mathbf{q})$.*

Proof. We start with the case $\alpha > 0$. First note that \mathcal{G}' is zero-sum indeed:

$$\text{Payoff}_0^{\mathcal{G}'}(\mathbf{p}, \mathbf{q}) + \sum_{k \in \mathcal{K}} x_k \text{Payoff}_k^{\mathcal{G}'}(\mathbf{p}, \mathbf{q}) = 0.$$

The NE 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 $BR_k(\mathbf{q})$ is the same in \mathcal{G} and \mathcal{G}' . Now, the set of best responses for the inspector in \mathcal{G} is

$$BR_0^{\mathcal{G}}(\mathbf{p}) = \arg \max_{\mathbf{q} \in \mathcal{Q}} \sum_{e \in E} p_e (\alpha \sigma_e q_e + \beta_e).$$

For a fixed \mathbf{p} , let us add $\sum_{e \in E} p_e (\alpha w_e - \beta_e)$ in the function to maximize. This term does not depend on \mathbf{q} and so it does not change the set of maximizers:

$$BR_0^{\mathcal{G}}(\mathbf{p}) = \arg \max_{\mathbf{q} \in \mathcal{Q}} \alpha \sum_{e \in E} p_e (\sigma_e q_e + w_e) = BR_0^{\mathcal{G}'}(\mathbf{p}).$$

Finally, the case $\alpha = 0$ is trivial, because $\text{Payoff}_0^{\mathcal{G}}(\mathbf{p}, \mathbf{q})$ does not depend on \mathbf{q} . \square

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 could allow us to formulate the game as a pairwise zero-sum polymatrix game [2], a class of games for which a Nash equilibrium can also be computed by linear programming [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 characterization of best responses. This yields a compact LP for the computation of a NE of \mathcal{G} .

By Proposition 3.1 we know that the loss of Player k at a NE (\mathbf{p}, \mathbf{q}) is $\text{spl}_k(\mathbf{w} + \sigma \circ \mathbf{q})$. The computation of a NE of \mathcal{G}' (and hence of \mathcal{G}) thus reduces to the computation of a strategy $\mathbf{q} \in \mathcal{Q}$ maximizing $\sum_k x_k \text{spl}_k(\mathbf{w} + \sigma \circ \mathbf{q})$, a weighted sum of shortest path lengths. This can be done by linear programming, by introducing some vertex potentials y_v^s for every source vertex $s \in S := \{s_k : k \in \mathcal{K}\}$ and for all $v \in V$:

$$\max_{\mathbf{q}, \mathbf{y}} \quad \sum_{k \in \mathcal{K}} x_k y_{d_k}^{s_k} \quad (10a)$$

$$\text{s. t.} \quad y_v^s - y_u^s \leq w_e + \sigma_e q_e, \quad \forall s \in S, \forall e \equiv (u, v) \in E; \quad (10b)$$

$$y_s^s = 0, \quad \forall s \in S; \quad (10c)$$

$$\mathbf{q} \in \mathcal{Q}. \quad (10d)$$

The constraints (10b)-(10c) 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 $G = (V, E, \mathbf{c}(\mathbf{q}))$. The objective function (10a) hence asks for the maximization of the weighted sum of shortest paths $\sum_k x_k \lambda_k(\mathbf{q})$ over the set of feasible inspector's strategies (10d).

We point out that the optimal dual variables of constraint (10b) 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 \mathbf{p}^k for every Player k .

Observe that in accordance with Proposition 5.2, the LP (10) for the computation of a Nash equilibrium does not depend on the parameters α and β . The concept of a Stackelberg equilibrium looks much more suitable for our application, but as we shall see, the computation of such an equilibrium is also much harder.

5.2 Stackelberg equilibria

Using ideas similar to [13], a mixed integer program (MIP) can be formulated for the computation of a Stackelberg equilibrium (\mathbf{p}, \mathbf{q}) . We reduce drastically the number of required variables, by using a single-source multi-sink flow

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

for each $s \in S$, instead of using a flow \mathbf{p}^k for every commodity. With the use of *big-M* constraints, we ensure that this flow uses only arcs belonging to $\text{SP}_k(\mathbf{w} + \boldsymbol{\sigma} \circ \mathbf{q})$. By Proposition 3.1, $\boldsymbol{\rho}^s$ hence corresponds to best-response strategies to \mathbf{q} for the players whose commodity source is s .

$$\max_{\mathbf{q}, \mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\rho}} \quad \sum_{k \in \mathcal{K}} \alpha x_k y_{d_k}^{s_k} + \sum_{s \in S} \sum_{e \in E} \rho_e^s (\beta_e - \alpha w_e) \quad (11a)$$

s. t.

$$0 \leq w_e + \sigma_e q_e - (y_v^s - y_u^s) \leq M_e (1 - \mu_e^s), \quad \forall s \in S, \forall e \equiv (u, v) \in E; \quad (11b)$$

$$y_s^s = 0, \quad \forall s \in S; \quad (11c)$$

$$\mathbf{q} \in \mathcal{Q}, \quad (11d)$$

$$\sum_{e' \in \delta^+(v)} \rho_{e'}^s - \sum_{e \in \delta^-(v)} \rho_e^s = \begin{cases} \sum_{k \in \mathcal{K}_s} x_k & \text{if } s = v; \\ -x_{(s,v)} & \text{if } (s, v) \in \mathcal{K}_s; \\ 0 & \text{otherwise,} \end{cases} \quad \forall s \in S, \forall v \in V; \quad (11e)$$

$$0 \leq \rho_e^s \leq M^s \mu_e^s, \quad \forall s \in S, \forall e \in E; \quad (11f)$$

$$\mu_e^s \in \{0, 1\}, \quad \forall (s, e) \in S \times E. \quad (11g)$$

As in Problem (10), constraints (11b)-(11c) bound $y_{d_k}^{s_k}$ from above by the shortest path length for commodity k in the graph with weights $\mathbf{w} + \boldsymbol{\sigma} \circ \mathbf{q}$, and constraint (11d) forces \mathbf{q} to be a feasible strategy for the inspector. We introduce a binary variable μ_e^s which can take the value 1 only if arc e belongs to a shortest path tree rooted at s (second inequality in (11b)). Indeed, the first inequality in (11b) is saturated when the difference of potential ($y_v^s - y_u^s$) between the extreme points of an arc $e \equiv (u, v)$ equals the weight $w_e + \sigma_e q_e$ of e , which indicates that there is a shortest path originating at s that uses e .

Equation (11e) forces $\boldsymbol{\rho}^s$ to be a single-source multi-sink flow rooted at 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 (11f) ensures that the flow $\boldsymbol{\rho}^s$ only uses arcs from $\text{SP}_k(\mathbf{w} + \boldsymbol{\sigma} \circ \mathbf{q})$. Now, $\boldsymbol{\rho}^s$ can be decomposed as $\sum_{k \in \mathcal{K}_s} x_k \mathbf{p}^k$, where $\mathbf{p}^k \in F_k$ is a flow through commodity k of value one. By construction, \mathbf{p}^k is a flow of minimal cost $\text{spl}_k(\mathbf{w} + \boldsymbol{\sigma} \circ \mathbf{q}) = \sum_{e \in E} p_e^k (w_e + q_e \sigma_e)$, and it follows that \mathbf{p}^k is a best response to \mathbf{q} ; see Proposition 3.1.

Finally, the objective function (11a) reduces to the inspector's payoff (7) when replacing $y_{d_k}^{s_k}$ and ρ_e^s by their values as a function of p_e^k :

$$\begin{aligned} & \sum_{k \in \mathcal{K}} \alpha x_k y_{d_k}^{s_k} + \sum_{s \in S} \sum_{e \in E} \rho_e^s (\beta_e - \alpha w_e) \\ &= \sum_{k \in \mathcal{K}} \alpha x_k \sum_{e \in E} p_e^k (w_e + q_e \sigma_e) + \\ & \quad \sum_{s \in S} \sum_{e \in E} \sum_{k \in \mathcal{K}_s} x_k p_e^k (\beta_e - \alpha w_e) \\ &= \sum_{k \in \mathcal{K}} x_k \sum_{e \in E} p_e^k (\alpha q_e \sigma_e + \beta_e). \end{aligned}$$

5.3 Stackelberg formulation for MAXTOLL in the transit network model

In this section, we show that the Stackelberg MIP (11) can be simplified for the case of the transit network model introduced in §4.1 with $\alpha = 0$ (MAXTOLL). In this situation indeed, the inspector'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 ρ^s are not involved anymore:

$$\max_{\mathbf{q}, \mathbf{y}, \boldsymbol{\mu}} \quad \sum_{k \in \mathcal{K}} x_k \tau_k \mu_k \quad (12a)$$

$$\text{s. t.} \quad y_v^s - y_u^s \leq w_e + \sigma_e q_e, \quad \forall s \in S, \forall e \equiv (u, v) \in E; \quad (12b)$$

$$y_s^s = 0, \quad \forall s \in S; \quad (12c)$$

$$w_{\tilde{e}_k} - y_{\tilde{d}_k}^{\tilde{s}_k} \leq M_k(1 - \mu_k), \quad \forall k \in \mathcal{K}; \quad (12d)$$

$$\mu_k \in \{0, 1\}, \quad \forall k \in \mathcal{K}; \quad (12e)$$

$$\mathbf{q} \in \mathcal{Q}. \quad (12f)$$

The binary indicator variable μ_k can take the value 1 if and only if the inequality corresponding to $\tilde{e}_k = (\tilde{s}_k, \tilde{d}_k)$ in (12b) is saturated, i.e. when the single arc $(\tilde{s}_k, \tilde{d}_k)$ forms a shortest path for commodity k , which means that Player k has an incentive to pay the toll.

6 The Price of Spite of SC games

We have seen in the last section that computing a NE is easy (and can be done by LP), while the computation of a SSE is harder and requires integer variables. In this section, we will first show that the computation of a Stackelberg equilibrium is NP-hard indeed, which justifies the mixed integer programming approach of § 5.2. Then, we will see that under some additional assumptions –that are typically satisfied for the model structures presented in Section 4– the optimal Stackelberg payoff of the inspector can be reasonably well approximated by committing to a Nash strategy. Because the Nash strategy consists in allocating the controls in the most harmful way for the network users, we will introduce the concept of “price of spite” (*PoS*) for an SC game, which can be seen as the price of anarchy of the game when the performance of a solution is measured by the inspector's payoff in the Stackelberg model.

6.1 Computing a Stackelberg strategy is NP-hard

As evoked earlier, an SC game can be reformulated as a polymatrix game, see (2) and Section 2.3 in [2], in which the inspector simultaneously plays against all network users, but there is no interaction between users. Such games are called “Bayesian games” by some authors, because they are equivalent to the following situation: the inspector plays against the users of an unknown commodity, which is chosen to be (s_k, d_k) with probability proportional to x_k . It has been shown by

Conitzer and Sandholm [5] that computing a Stackelberg strategy of the leader in a Bayesian game is NP-hard, thanks to a reduction from SET-COVER.

However, SC games do not entail the whole generality of Bayesian games. In particular, the case $\alpha = 1$ (MAXPROFIT) appears to be an *easier* problem, because the sub-game played between the inspector and Player k has a particular *zero-sum plus costs* structure (the *zero-sum part* corresponds to the rewards and penalties paid to the inspector, and the *cost part* accounts for the difference $\sum_{e \in R} (w_e - \beta_e)$ charged to Player k when she chooses the route R). Hence, the MAXPROFIT game can be reformulated as a *pairwise zero-sum* polymatrix game, with a dummy player who has a single strategy, consisting in *receiving* the aforementioned cost part. It is known that pairwise zero-sum polymatrix games are polynomially solvable, i.e., a Nash equilibrium can be computed in polynomial time [4]. As a byproduct of the next theorem, we obtain that the computation of a Stackelberg strategy of the leader is NP-hard in a polymatrix game, even when the game is pairwise zero-sum. This could not have been deduced from any result in [5]. Our proof is based on a reduction from the 0/1 KNAPSACK problem and is given in Appendix A.

Theorem 6.1 (Computing a Stackelberg strategy is NP-hard). *Given an SC game $\mathcal{G} = (G, \mathcal{C}, \mathcal{W}, \mathcal{Q}, \alpha)$, it is NP-hard to compute a Stackelberg strategy of the inspector, even when $\alpha = 1$, Player k has only two routes from s_k to d_k , and $\mathcal{Q} = \{\mathbf{q} \in \mathbb{R}_+^{|E|} : \sum_{e \in E} q_e = 1\}$ is the standard probability simplex over the set of arcs E .*

Corollary 6.2. *Consider a pairwise zero-sum polymatrix game \mathcal{G} , in which one of the players is seen as the leader. It is NP-hard to compute a Stackelberg equilibrium of \mathcal{G} .*

Proof. Consider an arbitrary SC game \mathcal{G} as in Theorem 6.1 ($\alpha = 1$, two routes per commodity, \mathcal{Q} is the unit simplex). Since $\alpha = 1$, the game has a *zero-sum plus costs* structure, and so \mathcal{G} can be represented as a pairwise zero-sum polymatrix game. Moreover, the restrictions on \mathcal{Q} and the number of routes per commodity ensure that the reduction from \mathcal{G} to a polymatrix game in normal form is polynomial. \square

Remark 6.3. The reduction from KNAPSACK suggests that computing a SSE in a pairwise zero-sum polymatrix game is maybe *not so hard*. Indeed, it is well known that there is a fully polynomial-time approximation scheme (FPTAS) for KNAPSACK, which relies on dynamic programming. This leaves open the question of the approximability of MAXPROFIT in polynomial time. We want to investigate this issue in future work.

6.2 The price of spite

In game theory, the concept of Price of Anarchy (PoA) has been introduced to measure the loss of efficiency of a system when its agents adopt a selfish behavior, and has been popularized by the pioneering work of Roughgarden

and Tardos on congestion games [15]. The price of spite that we will next define has a similar vein, albeit a different meaning. Indeed, in SC games the Nash equilibrium should not be interpreted as the natural outcome of a game played by selfish players. We have argued that the inspector should commit to a Stackelberg strategy, but a Nash strategy could still be seen as a heuristic choice for the inspector: maximizing the total loss of his adversaries is much easier than maximizing his own payoff. The price of spite is thus a measure of how the payoff of the inspector degrades when choosing this *most harmful* strategy. Perhaps a better name for this concept would have been *price of non-commitment*. Indeed, the inspector does not really aim to spite when committing to a NE. Instead, the Nash model makes sense when the inspector cannot credibly commit to a strategy before the network users take their decisions, and hence he must play a best response.

Definition 6.4. Consider an SC game $\mathcal{G} = (G, \mathcal{C}, \mathcal{W}, \mathcal{Q}, \alpha)$. We measure the efficiency of an inspector strategy $\mathbf{q} \in \mathcal{Q}$ by the inspector's payoff in the Stackelberg model (i.e., network users select a best response route that favors the inspector most):

$$H(\mathbf{q}) := \max_{\mathbf{p} \in BR(\mathbf{q})} \sum_{e \in E} p_e(\alpha \sigma_e q_e + \beta_e). \quad (13)$$

The next proposition gives an upper bound which is very useful to measure the quality of a Nash strategy \mathbf{q}^* . We will need some additional notation. Given a strategy $\mathbf{p} \in F(\mathcal{C})$ of the network users, we define the metric $d(\mathbf{p}) = \sum_{e \in E} p_e(\alpha w_e - \beta_e)$. This corresponds to the total latency experienced by all network users when the time to travel on arc e is $\alpha w_e - \beta_e$. We also define d_{\min} to be the smallest possible value of $d(\mathbf{p})$, that is,

$$d_{\min} := \min_{\mathbf{p} \in F(\mathcal{C})} \sum_{e \in E} p_e(\alpha w_e - \beta_e) = \sum_k x_k \text{spl}_k(\alpha \mathbf{w} - \boldsymbol{\beta}).$$

Note that $(\alpha w_e - \beta_e)$ is nonnegative if $\alpha = 1$, because $w_e \geq \beta_e$. However this quantity could be negative if α is small. But we can still define $d_{\min} > -\infty$ if G has no cycle of negative weight (for the weights $\alpha w_e - \beta_e$). In particular, this happens in the transit model presented in Section 4.1, because the only arcs with a negative weight are the special toll-arcs (\bar{s}_k, \bar{d}_k) and they do not belong to any cycle.

We denote the inspector's payoff by $I(\mathbf{p}, \mathbf{q})$, so that $H(\mathbf{q}) = \max_{\mathbf{p} \in BR(\mathbf{q})} I(\mathbf{p}, \mathbf{q})$. For short, we say that a strategy $\mathbf{p} \in BR(\mathbf{q})$ satisfying $H(\mathbf{q}) = I(\mathbf{p}, \mathbf{q})$ is a *tie-breaking best response* to \mathbf{q} . We denote the total loss of all network users by $\Lambda(\mathbf{p}, \mathbf{q}) := \sum_{e \in E} p_e(\sigma_e q_e + w_e)$. A simple calculation shows that the following relation holds for all strategy profiles $(\mathbf{p}, \mathbf{q}) \in F(\mathcal{C}) \times \mathcal{Q}$:

$$I(\mathbf{p}, \mathbf{q}) = \alpha \Lambda(\mathbf{p}, \mathbf{q}) - d(\mathbf{p}). \quad (14)$$

Proposition 6.5. Let $\mathcal{G} = (G, \mathcal{C}, \mathcal{W}, \mathcal{Q}, \alpha)$ be a SC game such that G has no negative weight cycle (with respect to the weights $\alpha \mathbf{w} - \boldsymbol{\beta}$). Let $(\mathbf{p}^*, \mathbf{q}^*)$ be a

NE of \mathcal{G} and let $\hat{\mathbf{p}}$ be a tie-breaking best response to \mathbf{q}^* . Then, for all $\mathbf{q} \in \mathcal{Q}$ we have

$$H(\mathbf{q}) \leq H(\mathbf{q}^*) + d(\hat{\mathbf{p}}) - d_{\min}.$$

Proof. Let \mathbf{p} be a tie-breaking best response to \mathbf{q} , so we have $H(\mathbf{q}) = I(\mathbf{p}, \mathbf{q})$ and $H(\mathbf{q}^*) = I(\hat{\mathbf{p}}, \mathbf{q}^*)$. By definition of the NE, it holds that $\Lambda(\mathbf{p}^*, \mathbf{q}^*) \geq \Lambda(\mathbf{p}, \mathbf{q})$, because $\mathbf{p} \in BR(\mathbf{q})$. Moreover, we have $\Lambda(\mathbf{p}^*, \mathbf{q}^*) = \Lambda(\hat{\mathbf{p}}, \mathbf{q}^*)$ because $\hat{\mathbf{p}} \in BR(\mathbf{q}^*)$. We now apply the relation (14) two times:

$$\begin{aligned} H(\mathbf{q}) &= I(\mathbf{p}, \mathbf{q}) = \alpha \Lambda(\mathbf{p}, \mathbf{q}) - d(\mathbf{p}) \\ &\leq \alpha \Lambda(\hat{\mathbf{p}}, \mathbf{q}^*) - d(\mathbf{p}) \\ &= I(\hat{\mathbf{p}}, \mathbf{q}^*) + d(\hat{\mathbf{p}}) - d(\mathbf{p}) \\ &\leq H(\mathbf{q}^*) + d(\hat{\mathbf{p}}) - d_{\min}, \end{aligned}$$

where the last inequality follows from $d(\mathbf{p}) \geq d_{\min}$. \square

In words, this proposition bounds the loss of efficiency of the Nash strategy \mathbf{q}^* by the *detour* done by the tie-breaking best response flow $\hat{\mathbf{p}}$ (for the metric d). Given the strategy \mathbf{q}^* , we point out that d_{\min} and $d(\hat{\mathbf{p}})$ correspond respectively to shortest path lengths (for the metric $d(\cdot)$) in the whole graph and in the directed acyclic subgraphs $\text{SP}_k(\mathbf{w} + \boldsymbol{\sigma} \circ \mathbf{q}^*)$. In real-world applications, this bound turns out to be excellent for $\alpha = 1$ (MAXPROFIT); cf. Section 7.

In the remainder of this section, we assume that the rewards β_e are nonnegative. This ensures that the payoff of the inspector is always nonnegative, and justifies the following definition:

Definition 6.6 (price of spite). Define by $N \subset \mathcal{Q}$ the set of Nash equilibrium strategies of the inspector. The price of spite is defined as the worst possible ratio between a Stackelberg and a Nash strategy:

$$PoS := \frac{\max_{\mathbf{q} \in \mathcal{Q}} H(\mathbf{q})}{\min_{\mathbf{q} \in N} H(\mathbf{q})}.$$

Without further assumptions on the model however, there is no hope to obtain an upper bound for the price of spite of a SC game \mathcal{G} . Indeed, a reviewer has suggested an example with the following properties: the efficiency of the Nash strategy is $H(\mathbf{q}^*) = 0$, while the Stackelberg strategy has efficiency $H(\mathbf{q}) = 1$. This example is sketched in Appendix B. A bound can be obtained though, under a set of additional assumptions. We next consider a set of hypotheses that are satisfied by the distance-based toll in Germany. In Remark 6.8, we will also give a similar result for the model of public transportation network presented in Section 4.1.

We assume that every arc $e \in E$ has a *length* $l_e \geq 0$; in addition, there are some constants $b > 0$ (basic costs per km) and $f > 0$ (fare per km) such that

- (A1) There exists a partition of the arc set $E = E_P \cup E_E \cup E_D$ (pay, evasion, and dummy arcs) such that

$$\begin{cases} w_e = (b + f)l_e, & \beta_e = fl_e, & \sigma_e = 0 & \text{if } e \in E_P \\ w_e \geq bl_e, & \beta_e = 0, & \sigma_e > 0 & \text{if } e \in E_E \\ w_e \geq 0, & \beta_e = \sigma_e = l_e = 0 & & \text{if } e \in E_D \end{cases}$$

- (A2) The shortest (s_k, d_k) -path R for commodity k has length $\ell_k := \sum_{e \in R} l_e$, and there exists an (s_k, d_k) -path R_{pay}^k that uses only arcs of type E_P and E_D (i.e., no evasion arc) where the total user costs are $\sum_{e \in R_{\text{pay}}^k} w_e = (b + f)\ell_k$.
- (A3) There exists a uniform control strategy $\mathbf{q}^U \in \mathcal{Q}$ such that $q_e^U = 0$ for all toll-free arcs $e \in E_P \cup E_D$, and q_e^U is proportional to $\sigma_e^{-1}l_e$ on the evasion arcs, that is, $\exists u > 0 : \forall e \in E_E, q_e^U = u \frac{l_e}{\sigma_e}$.

Note that the quantity u from Assumption (A3) may be interpreted as the average penalty per evaded kilometer. For example, consider the simple control set where the pure strategies of the inspectors are all subsets of E of cardinality γ ; see (4)-(5). It is easy to verify that u must be equal to

$$u(\gamma) := \frac{\gamma}{\sum_{e \in E_E} \sigma_e^{-1}l_e}, \quad (15)$$

so that Assumption (A3) is satisfied whenever $\gamma \sigma_e^{-1}l_e \leq \sum_{e \in E_E} \sigma_e^{-1}l_e$ for all $e \in E_E$.

Theorem 6.7. *Consider an SC game \mathcal{G} satisfying assumptions (A1)-(A3). Then,*

$$PoS \leq \max\left(1, \frac{f}{\alpha u}\right).$$

Proof. Recall that $\text{spl}_k(l) = \ell_k$. We denote by \mathfrak{L} the minimal distance covered by all network users: $\mathfrak{L} := \sum_{k \in \mathcal{K}} x_k \ell_k$. We first claim that the efficiency of any feasible strategy $\mathbf{q} \in \mathcal{Q}$ is bounded by $H(\mathbf{q}) \leq f\mathfrak{L}$. Otherwise, let $\mathbf{p} \in BR(\mathbf{q})$ be such that $I(\mathbf{p}, \mathbf{q}) > f\mathfrak{L}$. This implies the existence of a commodity k for which $\sum_{e \in E} p_e^k (\alpha \sigma_e q_e + \beta_e) > f\ell_k$. Denote by ℓ_k^+ the average length covered by flow \mathbf{p}^k : $\ell_k^+ = \sum_{e \in E} p_e^k l_e \geq \ell_k$. Observe that $bl_e \leq w_e - \beta_e$ for all $e \in E$ (this follows from (A1)). Thus, the total loss of Player k is

$$\sum_{e \in E} p_e^k (\sigma_e q_e + w_e) \geq \sum_{e \in E} p_e^k (\sigma_e q_e + \beta_e) + b\ell_k^+ > f\ell_k + b\ell_k.$$

But then, this contradicts $\mathbf{p} \in BR(\mathbf{q})$, because the flow placing all the weight on the (s_k, d_k) -path R_{pay}^k of Assumption (A2) yields a loss $(f + b)\ell_k$.

Then, let $(\mathbf{p}^*, \mathbf{q}^*)$ be a NE of \mathcal{G} , and observe that $H(\mathbf{q}^*) \geq I(\mathbf{p}^*, \mathbf{q}^*)$. Since $\mathbf{q}^* \in BR(\mathbf{p}^*)$, $\Lambda(\mathbf{p}^*, \mathbf{q}^*) = \max_{\mathbf{q} \in \mathcal{Q}} \sum_{e \in E} p_e^* (\sigma_e q_e + w_e)$, and Equation (14) gives

$$I(\mathbf{p}^*, \mathbf{q}^*) = \max_{\mathbf{q} \in \mathcal{Q}} \sum_{e \in E} p_e^* (\alpha \sigma_e q_e + \beta_e).$$

Therefore,

$$\begin{aligned} I(\mathbf{p}^*, \mathbf{q}^*) &\geq \min_{\mathbf{p} \in F(\mathcal{C})} \max_{\mathbf{q} \in \mathcal{Q}} \sum_{e \in E} p_e (\alpha \sigma_e q_e + \beta_e) \\ &= \max_{\mathbf{q} \in \mathcal{Q}} \min_{\mathbf{p} \in F(\mathcal{C})} \sum_{e \in E} p_e (\alpha \sigma_e q_e + \beta_e), \end{aligned}$$

where the last equality follows e.g. from Sion's minimax theorem. Now, for the homogeneous control strategy \mathbf{q}^U from Assumption (A3), we obtain

$$\begin{aligned} I(\mathbf{p}^*, \mathbf{q}^*) &\geq \min_{\mathbf{p} \in F(\mathcal{C})} \left(\alpha u \sum_{e \in E_E} p_e l_e + f \sum_{e \in E_P} p_e l_e \right) \\ &\geq \min(f, \alpha u) \mathfrak{L}. \end{aligned}$$

We finally obtain the desired inequality, by combining the inequalities $H(\mathbf{q}^*) \geq \min(f, \alpha u) \mathfrak{L}$ and $H(\mathbf{q}) \leq f \mathfrak{L}$ for an arbitrary $\mathbf{q} \in \mathcal{Q}$:

$$\frac{H(\mathbf{q})}{H(\mathbf{q}^*)} \leq \frac{f}{\min(f, \alpha u)} = \max \left(1, \frac{f}{\alpha u} \right).$$

□

Remark 6.8. We can also adapt the proof of Theorem 6.7 to work in the framework of the transit network model of Section 4.1. Here, we need to assume $\sigma_e > 0$ ($\forall e \in E_0$) and the existence of a homogeneous control $q_e^U = u \sigma_e^{-1} l_e$ ($\forall e \in E_0$). By using a similar reasoning, we can prove that

$$PoS \leq \frac{\sum_k x_k \tau_k}{\sum_k x_k \min(\tau_k, \alpha u l_k)}.$$

Remark 6.9. The key point in the previous proof is that the graph G contains no arc such that $\beta_e = \sigma_e = 0$ (except possibly on dummy arcs that have a length $l_e = 0$). Such arcs typically correspond to shortcut (or detour) arcs that avoid the paying network, and can yield an arbitrarily large PoS ; see Appendix B. However, we show in Section 7.4 that the presence of shortcuts only has a limited impact on the PoS for the German truck toll scenario.

7 Numerical Results

We have solved the models presented in this paper for several instances. On the one hand, random instances were generated to demonstrate the scalability of our model; cf. Section 7.1. On the other hand, we have used real data from the German motorways network, and we shall analyze our results for both static instances (Section 7.2-7.3) and instances where the time is taken into account (Section 7.4). In particular, we will investigate the quality of the bounds derived in Section 6.2 for the price of spite.

Instance Size			Efficiency of Nash vs. Stackelberg		Lower bound Nash eff. for $\alpha = 1$ cf. Prop 6.5	Computation time (s) and Gap (%)				
$ V_0 $	$ E_0 $	$ \mathcal{C} $	$\alpha = 1$	$\alpha = 0$		Nash	$\alpha = 1$	Gap	$\alpha = 0$	Gap
10	32	27	100.00%	88.24%	100.00%	0.001	0.01	0.00%	0.03	0.00%
15	73	65	99.74%	86.33%	99.65%	0.006	0.49	0.01%	0.18	0.01%
20	70	131	99.79%	69.90%	99.19%	0.012	0.80	0.01%	18.58	0.01%
25	199	194	98.47%	87.83%	98.27%	0.082	6.25	0.01%	18.11	0.01%
30	342	283	99.42%	94.70%	99.41%	0.336	7.34	0.01%	10.92	0.01%
35	431	382	99.43%	93.13%	99.29%	0.250	39.15	0.01%	45.16	0.01%
40	526	462	99.74%	95.24%	99.60%	0.274	54.08	0.01%	51.24	0.01%
45	543	623	99.53%	95.23%	99.23%	0.341	3216.10	0.01%	546.31	0.01%
50	440	783	98.54%	84.71%	97.64%	2.154	3600.00	0.74%	3600.00	6.41%
55	1261	962	100.00%	100.00%	100.00%	5.404	80.49	0.00%	50.78	0.00%
60	896	1123	99.32%	94.70%	99.09%	8.316	3600.00	0.20%	3600.00	0.33%
65	757	1346	98.82%	91.81%	97.96%	6.116	3600.00	0.84%	3600.00	5.39%
70	948	1510	99.14%	92.86%	98.77%	5.402	3600.00	0.31%	3600.00	3.37%
75	1157	1705	99.33%	95.73%	99.09%	18.115	3600.00	0.21%	3600.00	0.44%
100	2018	3099	99.92%	98.94%	99.87%	60.670	3600.00	0.06%	3600.00	0.41%
125	3037	4820	99.99%	100.00%	99.64%	268.145	3600.00	0.36%	3600.00	2.36%
150	2660	7017	100.00%	100.00%	98.63%	481.834	3600.00	1.39%	3600.00	9.62%

Table 1: Computational results for randomly generated instances. The columns of this table are explained in Section 7.1.

Mobile controllers drive on the network and are able to control every truck they overtake. Hence, the expected penalty to pay on a section where an inspector is driving can be approximated by $(\frac{v_I}{v_T} - 1)P$, where P is the amount of a fine, and v_I and v_T represent the average speeds of the inspectors and the truck drivers, respectively. In all our experiments, we have set $\sigma_e = 60\text{€}$ on every evasion arc (the penalty is $P = 400\text{€}$ and controllers drive 15% faster than truck drivers). The basic costs (e.g. fuel and driver's salary) were set to $b = 0.5\text{€}/\text{km}$, and the toll fares are $f = 0.17\text{€}/\text{km}$.

All our instances were solved with CPLEX [6], on a PC with 8 cores at 3.2 GHz.

7.1 Randomly generated instances

We have generated instances of the SC games as follows. The graph $G_0 = (V_0, E_0)$ contains n vertices and for every ordered pair of vertices we generate a directed arc with a probability chosen between 0.1 and 0.4, depending on the instance. Then, the graph G of the SC game is created according to the transit model described in Section 4.1. A traffic volume x_k is randomly generated for each of the $n(n - 1)$ origin-destination pairs of the network, with a log-normal distribution. To construct the set of commodities \mathcal{C} , we sort the x_k and keep only the top commodities, so that 80% of the total traffic is considered in the instance. Finally, we considered the strategy set \mathcal{Q} defined by Relations (4)–(5), with γ chosen such that the ratio $\frac{f}{u(\gamma)} = 2$; see (15).

The results are displayed in Table 1. Each row shows the size of the considered instance (number of vertices $n = |V_0|$, number of arcs $|E_0|$ and number of commodities $|\mathcal{C}|$; the efficiency $\frac{H(\mathbf{q}^*)}{H(\tilde{\mathbf{q}})}$ of the Nash strategy \mathbf{q}^* , compared to the best known strategy $\tilde{\mathbf{q}}$ for the MAXPROFIT problem ($\alpha = 1$) and for the MAXTOLL problem ($\alpha = 0$); the bound on the efficiency (for $\alpha = 1$)

of the Nash strategy calculated with the result of Proposition 6.5, that is, $\text{eff} \geq \frac{H(\mathbf{q}^*)}{H(\mathbf{q}^*) + d(\hat{\mathbf{p}}) - d_{\min}}$. Besides, some information on the computation is displayed: each instance was solved with CPLEX, with a time limit of one hour. The computing time of the Nash LP (10), the Stackelberg MIP (11) and the MAXTOLL MIP (12) are displayed in each row, as well as the optimality gap reached within the allocated time for these two MIPs.

The instances with 3600s of CPU time are those that stopped because the time limit had been reached; in almost all cases, we see that CPLEX returned a solution guaranteed to be within 1% of the optimum for MAXPROFIT, and within 5% for MAXTOLL. Note also that the bound of Proposition 6.5 is often very tight. This bound guarantees e.g. that the efficiency of the NE is at least 99.41% for the instance where $|V_0| = 30$, and indeed its efficiency compared to the SSE is 99.42%.

For several instances, the efficiency of the Nash equilibrium equals that of the best known solution for MAXPROFIT or MAXTOLL. In these cases, it means that the MIP solver was not able to improve on the Nash equilibrium, which was used to make a *warm start*. We think that simple greedy or exchange heuristics could help to quickly identify a better solution (in particular for MAXTOLL), and consequently reduce the computing time required to close the optimality gap.

7.2 Static case, transit network model

We next present results for two instances taken from the application to the German motorways, in the static case (i.e., time is not taken into account), with the simple transit network model of Section 4.1. The results presented below rely on the set of strategies \mathcal{Q} described in (4)–(5), where a set of γ inspectors can be arbitrarily distributed over the arcs of the network.

In the upper part of Figure 4(a), a Nash equilibrium strategy \mathbf{q}^* of the inspector on the whole German network is represented. Here it was assumed that $\gamma = 50$ controllers are simultaneously present on the network, which has 319 vertices, 2948 arcs and 5013 commodities. The dotted arcs on the figure represent toll-free shortcuts, where the cost per kilometer is twice as high as on the the motorway network ($w_e = 2bl_e$), but no control can occur ($\sigma_e=0$).

For $\alpha = 1$ (MAXPROFIT), the bound $H(\mathbf{q}) \leq H(\mathbf{q}^*) + d(\hat{\mathbf{p}}) - d_{\min}$ of Proposition 6.5 guarantees that the payoff from this Nash strategy is at least 99.3% of the optimal Stackelberg payoff. To compute this bound, note that we must first compute a Nash strategy \mathbf{q}^* and its tie-breaking best response $\hat{\mathbf{p}}$. In contrast, the bound of Theorem 6.7 depends only on the per kilometer fares f and the uniform penalty rate u . For the values f and u of this instance, the upper bound on the price of spite is $\frac{f}{u} = 1.44$.

Further tests on a smaller network representing the region of Berlin-Brandenburg (45 vertices, 130 arcs, 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

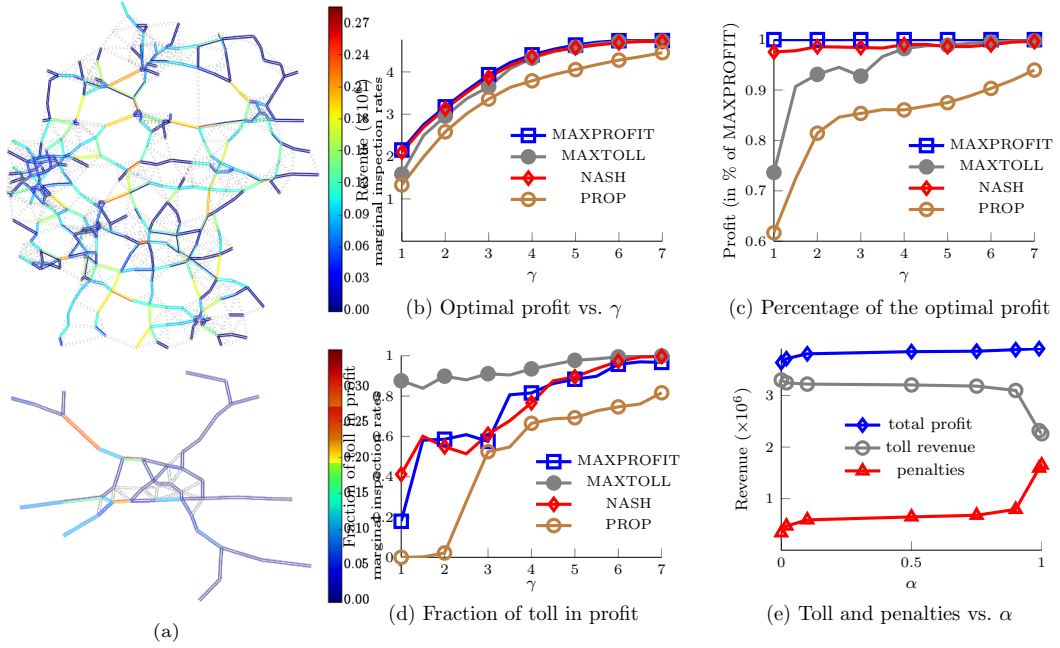


Figure 4: (a): Graphs of the two instances discussed in Section 7.2, with shades showing a near-optimal strategy for MAXPROFIT and $\gamma = 50$ on the network of Germany (above) and an optimal Stackelberg strategy for MAXTOLL and $\gamma = 3$ for Berlin-Brandenburg (below). (b)-(e): Experimental results for the region of Berlin-Brandenburg.

of Figure 4(a), with shades showing a Stackelberg strategy for MAXTOLL and $\gamma = 3$. Figures 4(b)-4(d) compare four strategies as a function of the number of controllers γ : the strategies MAXPROFIT and MAXTOLL, the Nash equilibrium strategy computed by LP (10), and a strategy in which control intensities are proportional to traffic volumes on each arc of the physical network G_0 (of course, no inspector is allocated to the dummy toll arcs where $\sigma_e = 0$). Figure 4(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 inspector most). We see in Figure 4(c) that the Nash strategy is always near-optimal in terms of profit. However, we point out that the MAXTOLL strategy outperforms the others in terms of toll enforcement (Figure 4(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 varied the parameter α , which joins MAXPROFIT ($\alpha = 1$) to MAXTOLL ($\alpha = 0$). Figure 4(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.

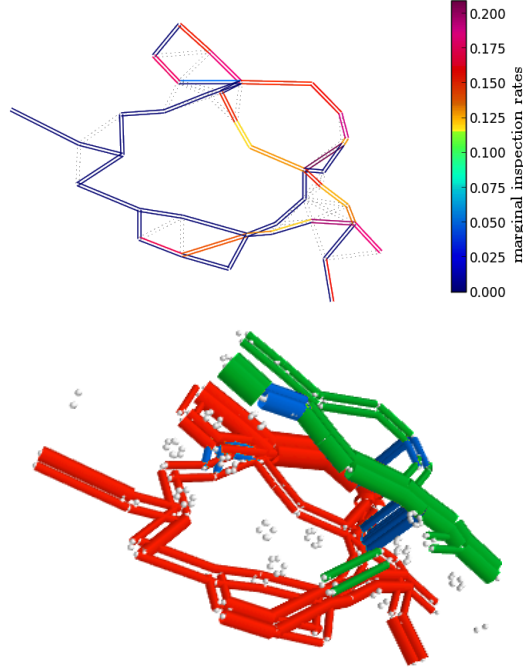


Figure 5: Inspector's Stackelberg strategy (left) and driver flows (right) for the MAX-PROFIT problem with $\gamma = 6$. In the lower 3D-figure, the width of the arcs is proportional to the number of network users in the two-layer graph. The lower level represents fare evasion, the upper level represents fare compliance, and darker arcs correspond to toll-free sections and transitions between the two levels.

7.3 Static case, two-layer network model

We shall now present some results for an instance constructed with the two-layer network model of Section 4.2. The graph of this instance represents the federal state of Rhineland-Palatinate; the graph G_0 has 37 vertices, 142 arcs and 323 commodities, which yields 148 vertices and 506 arcs in the two-layer graph G .

The Stackelberg strategy of the inspector for MAXPROFIT and $\gamma = 6$ is indicated in Figure 5, as well as the flows of network users in the two-layer graph. In the northern part of the graph, the arcs of the toll evasion level are much wider than the corresponding arcs in the fare compliance level, which indicates that many users have no incentive to pay the toll here.

The evolution of the inspector'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 inspector's payoff sinks between $\gamma = 7$ and

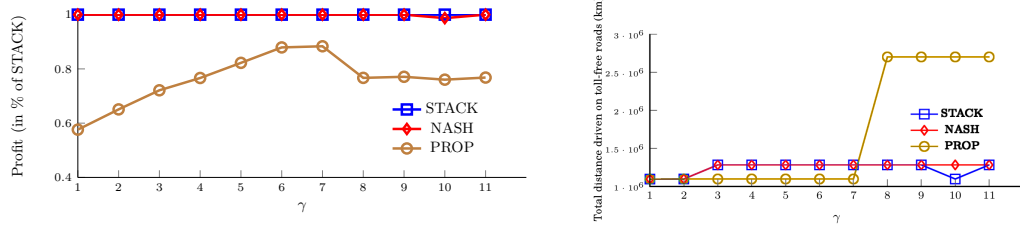


Figure 6: Evolution of the total profit and number of kilometers driven on the alternative toll-free trunk roads with γ , for the Rhineland-Palatinate instance in the MAXPROFIT setting.

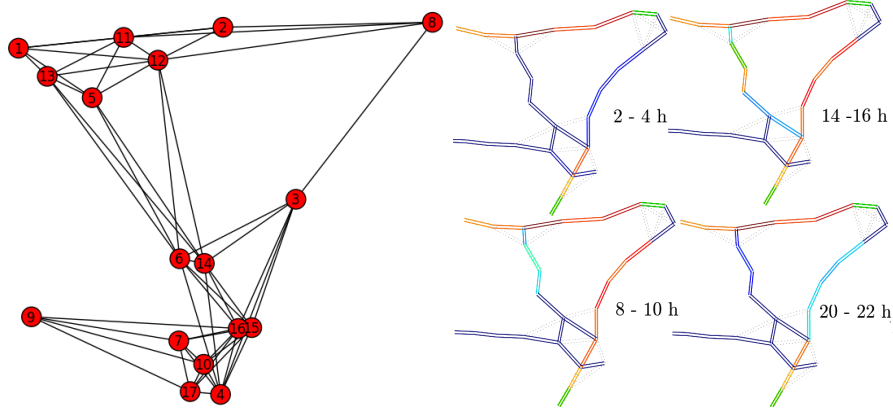


Figure 7: Graph C joining the control areas for the Saxony-Anhalt instance (left); Nash strategies of the inspector during four time windows, computed for $\gamma = 20$ inspector teams (right). Shades indicate inspection frequencies.

$\gamma = 8$ with the proportional strategy. This is explained by the lower graph of Figure 6: when $\gamma = 8$, this strategy assigns too high a control frequency on certain arcs, which creates a situation where many drivers have an incentive to take a toll-free trunk road, thus depriving the inspector of the incomes from both fares and fines on this arc.

7.4 Examples with time dynamics

We have considered several real-world instances corresponding to several control regions of Germany, for the model with cyclic duty graphs presented in Section 4.3. For example, the network corresponding to the control region located around the federal state of Saxony-Anhalt is depicted in Figure 7. The upper part of the figure shows the graph $C = (\mathcal{S}, \mathcal{A})$ connecting 17 sub-control areas, with vertices whose location coincide with the barycenter of each control

Instance	shortcuts	$ V_0 $	$ E_0 $	$ V $	$ E $	$ \mathcal{C} $	eff (%)	$\frac{f}{u}$	CPU (s)
Saxony Anhalt	no	30	60	903	2453	1190	99.85	1.94	2.32
Saxony Anhalt	yes	30	106	903	3005	1190	99.85	–	10.97
Lower Saxony	no	38	78	1263	3759	2016	100.00	1.41	6.48
Lower Saxony	yes	38	108	1263	4119	2016	99.91	–	15.76
Berlin-Brandenburg	no	62	128	1992	7769	4985	99.96	2.06	20.17
Berlin-Brandenburg	yes	62	328	1992	10169	4985	99.29	–	254.11
Northern Bavaria	no	56	116	1458	4073	1895	100.00	2.17	9.09
Northern Bavaria	yes	56	240	1458	5561	1895	99.85	–	78.80
Southern Bavaria	no	65	130	1985	6582	3817	98.63	3.05	20.21
Southern Bavaria	yes	56	304	1985	8670	3817	96.05	–	248.40
Schleswig-Holstein	no	46	92	1371	3543	1620	100.00	2.10	6.91
Schleswig-Holstein	yes	46	384	1371	7047	1620	95.34	–	13.22

Table 2: Bounds on the price of spite, for several real-world instances.

area $S \in \mathcal{S}$. We have used a time discretization of two hours, so that an inspector’s duty time of 8 hours corresponds to a length $L = 4$ in the cyclic duty graph D . The lower part of Figure 7 shows the intensity of the controls in the Nash strategy, during four different time windows.

Table 2 summarizes the results for several control regions of Germany, for the MAXPROFIT problem ($\alpha = 1$). For each instance, the table gives the size of both the graph $G_0 = (V_0, E_0)$ of the physical network, and its time-expanded version $G = (V, E)$ that was constructed along the lines of Section 4. In addition, the table shows the ratio $\text{eff} = \frac{H(\mathbf{q}^*)}{H(\mathbf{q}^*) + d(\hat{\mathbf{p}}) - d_{\min}}$ which bounds from below the efficiency of the Nash strategy \mathbf{q}^* (Proposition 6.5); the bound $\frac{f}{u}$ on the PoS (Theorem 6.7); and the CPU time to solve the LP (10) with CPLEX.

To evaluate the impact of shortcut arcs that allow the user to avoid the paying network (Remark 6.9) we have considered two variants of each instance. An instance *without shortcuts*, where we consider only the arcs of the motorway network (solid arcs in Figure 8), and an instance *with shortcuts* (dashed arcs). Since we have no data on secondary, toll-free trunk roads in Germany, shortcuts were generated according to the following procedure. An arc with weights $\sigma_e = \beta_e = 0, w_e = 2bd_{u,v}$ is added between the vertices u and v whenever the distance $d_{u,v}$ between u and v is below a threshold, and the strategy consisting in taking only the arc $e = (u, v)$ is not dominated by the *pay*-strategy from u to v on the motorway network: $2bd_{u,v} < (b + f) \text{spl}_{u \rightarrow v}(\mathbf{l})$.

Table 2 shows that the bound on the quality of \mathbf{q}^* degrades a bit when shortcut arcs are added, but for the considered instances we always have a guarantee that $H(\mathbf{q}^*)$ is at least 95% of the optimum. The upper bound on the PoS ($f/u = \text{toll rate per km/uniform penalty per km}$) is given only for the instances with no shortcut, so that Theorem 6.7 applies.

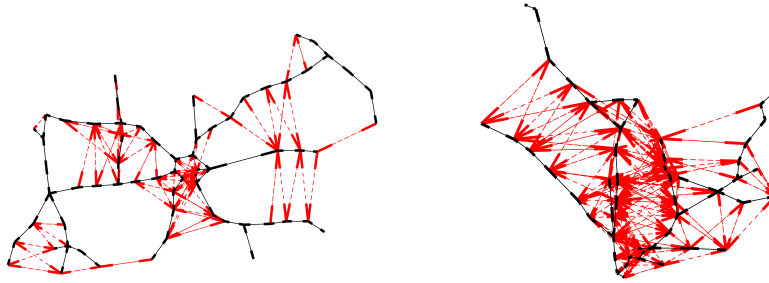


Figure 8: Motorway network (solid) and generated shortcut arcs (dashed), for the regions of southern Bavaria (left) and Schleswig-Holstein (right).

8 Conclusion and Perspectives

We have introduced the class of spot-checking games, which are particular security games played on a graph. These games can be used to represent many situations in which controls must be distributed over a network. We have proposed an LP / MIP based approach to compute Nash and Stackelberg equilibria of this game. Users' strategies are represented by multicommodity flows, which makes it possible to compute a Nash equilibrium of the game very efficiently. The computation of a Stackelberg equilibrium is NP-hard, which justifies our MIP approach. We have introduced the price of spite of spot-checking games, which measures how the payoff of the inspector degrades when committing to a Nash strategy, rather than to a Stackelberg strategy. Two bounds are given: the first one analyzes the quality of a Nash strategy \mathbf{q}^* in terms of detours of the tie-breaking best response multicommodity flow $\hat{\mathbf{p}}$. The second bound is for a distance-based toll network, and depends only on the ratio between the toll fares per kilometer and the average penalty per evaded kilometer (if the fare inspectors are uniformly spread over the network). Finally, we have reported computational experiments that show the efficiency of our approach.

For future work, we would like to extend our model, in order to take into account congestion effects. More precisely, we want to investigate the situation where the weight w_e of an arc and the expected penalty σ_e are functions of the number of agents taking this arc.

Another research direction suggested by this article is the investigation of polytime approximation algorithms for the computation of a Stackelberg equilibrium in a spot-checking game. We think that a FPTAS might exist for the case where $\alpha = 1$, and more generally for all pairwise zero-sum polymatrix games. We also want to investigate heuristics and approximation algorithms for MAXTOLL ($\alpha = 0$).

Finally, we would also like to investigate the tightness of the bound of Theorem 6.7.

9 Acknowledgement

The authors kindly thank the editor as well as two anonymous reviewers, for their valuable comments that were of a great help to strengthen these results and their presentation. We also wish to thank four anonymous reviewers of the SAGT'14 conference –where the theoretical part of this article has been presented as a brief announcement [16]– for their feedback. Finally, we are grateful to Thomas Schlechte for his support.

References

- [1] R. Borndörfer, J. Buwaya, G. Sagnol, and E. Swarat, Optimizing toll enforcement in transportation networks: A game-theoretic approach, Proc INOC 2013, Tenerife, Spain, ENDM, Vol. 41, Elsevier, 2013, pp. 253–260.
- [2] R. Borndörfer, B. Omont, G. Sagnol, and E. Swarat, A Stackelberg game to optimize the distribution of controls in transportation networks, Proc GAMENETS 2012, Vancouver, Canada, LNICST, Vol. 105, 2012, pp. 224–235.
- [3] R. Borndörfer, G. Sagnol, and E. Swarat, An IP approach to toll enforcement optimization on German motorways, Proc Operations Research (OR 2011), Springer, 2011, pp. 317–322.
- [4] Y. Cai and C. Daskalakis, On minmax theorems for multiplayer games, Proc 22nd Ann ACM-SIAM Symp on Discrete Algorithms (SODA'11), 2011, pp. 217–234.
- [5] V. Conitzer and T. Sandholm, Computing the optimal strategy to commit to, Proc 7th ACM Conf on Electronic Commerce, 2006, pp. 82–90.
- [6] IBM ILOG CPLEX, V12.1, Users manual for CPLEX, Technical report, International Business Machines Corporation, 2009.
- [7] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W. H. Freeman & Co., New York, 1979.
- [8] M. Jain, E. Kardes, C. Kiekintveld, F. Ordóñez, and M. Tambe, Security games with arbitrary schedules: A branch and price approach, Proc 24th AAAI Conf on Artificial Intelligence, AAAI Press, 2010, pp. 792–797.
- [9] A.X. Jiang, Z. Yin, C. Zhang, M. Tambe, and S. Kraus, Game-theoretic randomization for security patrolling with dynamic execution uncertainty, Proc 12th Int Conf on Autonomous Agents and Multiagent Systems (AAMAS 2013), 2013, pp. 207–214.
- [10] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe, Computing optimal randomized resource allocations for massive security games, Proc 8th Int Conf on Autonomous Agents and Multiagent Systems (AAMAS 2009), Vol. 1, 2009, pp. 689–696.
- [11] D. Korzhuk, 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 Intell Res 41(2) (2011), 297–327.
- [12] J. Letchford, D. Korzhuk, and V. Conitzer, On the value of commitment, Auton Agent Multi-Ag 28(6) (2014), 986–1016.

- [13] 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. Proc 7th Int Conf on Autonomous Agents and Multiagent Systems (AAMAS 2008), 2008, pp. 895–902.
- [14] T. Roughgarden, Selfish routing with atomic players, Proc 16th Ann ACM-SIAM Symp on Discrete Algorithms (SODA’05), 2005, pp. 1184–1185.
- [15] T. Roughgarden and E. Tardos. How bad is selfish routing? J ACM 49(2) (2002), 236–259.
- [16] G. Sagnol, R. Borndörfer, T. Schlechte and E. Swarat, The price of spite in spot-checking games (brief announcement), Proc 7th Int Symp on Algorithmic Game Theory (SAGT’2014), Lecture Notes in Computer Science, Vol. 8768, 2014, p. 293.
- [17] Y. Vorobeychik, B. An, M. Tambe, and S. Singh, Computing solutions in infinite-horizon discounted adversarial patrolling games, Proc Int Conf on Automated Planning and Scheduling (ICAPS14), 2014, pp. 314–322.
- [18] 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, AI Magazine 33(4) (2012), 59–72.

A NP-hardness of SC games

In this section we prove that it is NP-hard to compute a Stackelberg strategy of a SC game, even when $\alpha = 1$ (Theorem 6.1). We recall that such a strategy is a maximizer of $H(\mathbf{q})$ over \mathcal{Q} ; see (13).

Our proof is based on a reduction from the binary knapsack problem, which is well known to be NP-complete [7]:

(0/1 Knapsack) *Given are a finite set of objects Υ , a weight $\omega(a) \in \mathbb{N}$, a utility $u(a) \in \mathbb{N}$ for each $a \in \Upsilon$, and positive integers W (knapsack capacity) and U (desired utility). Is there a subset Υ' of Υ such that*

$$\sum_{a \in \Upsilon'} \omega(a) \leq W \text{ and } \sum_{a \in \Upsilon'} u(a) \geq U?$$

Proof of Theorem 6.1. Given an instance \mathfrak{K} of the 0/1 knapsack problem, we construct an SC game \mathcal{G} in which $\alpha = 1$, there are exactly two possible routes for each commodity $k \in \mathcal{K}$, and such that \mathcal{G} has the following property: there exists an inspector's strategy \mathbf{q} that yields a payoff $H(\mathbf{q}) \geq U$ if and only if \mathfrak{K} is a *yes*-instance (where $H(\mathbf{q})$ has been defined in (13)).

Our instance is constructed as follows. For each object $a \in \Upsilon$ there is a commodity (s_a, d_a) , with two arcs \hat{a} and \check{a} joining s_a to d_a : $G = (\{s_a, d_a : a \in \Upsilon\}, \{\hat{a}, \check{a} : a \in \Upsilon\})$. The arcs \hat{a} and \check{a} are respectively *pay*- and *evasion*-arcs, with weights

$$\begin{aligned} \beta_{\hat{a}} &= u(a), & \sigma_{\hat{a}} &= 0, & w_{\hat{a}} &= u(a) \\ \beta_{\check{a}} &= 0, & \sigma_{\check{a}} &= \frac{1}{2}, & w_{\check{a}} &= u(a) - \frac{\omega(a)}{2W}. \end{aligned}$$

Note that we can assume without loss of generality that $\omega(a) \leq W$ for each object a , so that $w_{\check{a}} \geq 0$. There is a single user on each commodity ($x_a = 1$), $\alpha = 1$, and the set of all inspector's strategies is the standard probability simplex over the evasion arcs $\mathcal{Q} = \{\mathbf{q} \geq \mathbf{0} : \sum_{a \in \Upsilon} q_{\check{a}} = 1\}$.

Given an inspector's strategy $\mathbf{q} \in \mathcal{Q}$, it is easy to compute the inspector's payoff $H(\mathbf{q})$; see Equation (13). Indeed, for a commodity $a \in \Upsilon$ two cases can appear. Either $q_{\check{a}} < \frac{\omega(a)}{W}$, and in this case the only best response is to take the evasion path. Or $q_{\check{a}} \geq \frac{\omega(a)}{W}$ and paying the toll is a best response. When there is a tie ($q_{\check{a}} = \frac{\omega(a)}{W}$), the user of commodity a chooses the strategy that favours the inspector most; that is, he will pay the toll because $\sigma_{\check{a}} q_{\check{a}} = \frac{\omega(a)}{2W} \leq \frac{1}{2} \leq u(a)$. Hence,

$$\begin{aligned} H(\mathbf{q}) &= \sum_{a: q_{\check{a}} < \frac{\omega(a)}{W}} \sigma_{\check{a}} q_{\check{a}} + \sum_{a: q_{\check{a}} \geq \frac{\omega(a)}{W}} \beta_{\hat{a}} \\ &= \frac{1}{2} \sum_{a: q_{\check{a}} < \frac{\omega(a)}{W}} q_{\check{a}} + \sum_{a: q_{\check{a}} \geq \frac{\omega(a)}{W}} u(a). \end{aligned} \tag{16}$$

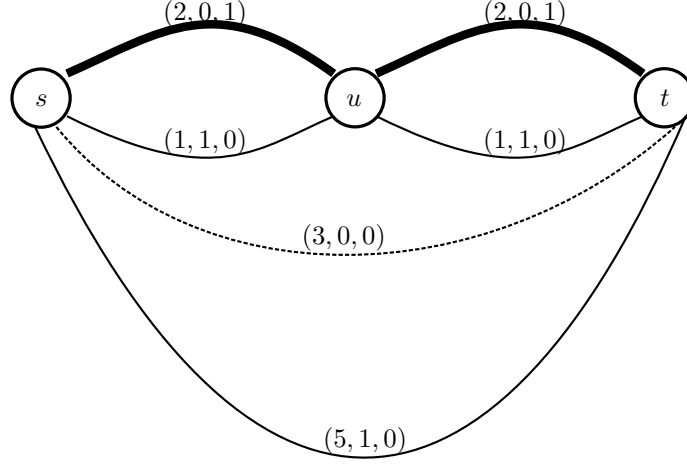


Figure 9: An SC game with a single commodity. Edges are labelled with weight triples (w_e, σ_e, β_e) .

We first assume that \mathfrak{K} is a *yes*-instance. Then, let Υ' be a knapsack of value $\geq U$. We define a control strategy by setting $q_{\bar{a}} \geq \frac{\omega(a)}{W}$ for all $a \in \Upsilon'$, and $q_{\bar{a}} = 0$ for all $a \notin \Upsilon'$ (this is possible because $\sum_{a \in \Upsilon'} \omega(a) \leq W$). Then, Equation (16) gives $H(\mathbf{q}) \geq \sum_{a \in \Upsilon'} u(a) \geq U$.

Conversely, let \mathbf{q} be such that $H(\mathbf{q}) \geq U$. Define $\Upsilon' := \{a \in \Upsilon : q_{\bar{a}} \geq \frac{\omega(a)}{W}\}$. Equation (16) gives $H(\mathbf{q}) = \frac{1}{2} \sum_{a \notin \Upsilon'} q_{\bar{a}} + \sum_{a \in \Upsilon'} u(a) \leq \frac{1}{2} + \sum_{a \in \Upsilon'} u(a)$. This implies that the integer $\sum_{a \in \Upsilon'} u(a)$ is larger than $U - \frac{1}{2}$, and thus it is larger than or equal to U . This proves that \mathfrak{K} is a *yes* instance.

Finally, the reduction from \mathfrak{K} to \mathcal{G} is clearly polynomial. Hence, finding a Stackelberg strategy of the inspector in a SC game is NP-hard, even when $\alpha = 1$ and each user has only two routes. \square

B An instance with unbounded price of spite

Consider the SC game depicted in Figure 9. There is a single commodity $\mathcal{C} = \{(s, t, 1)\}$. There are two teams of controllers, who can defend any subset of two arcs. The mixed strategy of the inspector is thus $\mathcal{Q} = \{\mathbf{0} \leq \mathbf{q} \leq \mathbf{1} : \sum_e q_e = 2\}$. Edge weights are labelled as triples (w_e, σ_e, β_e) . The wide solid arcs represent pay-arcs ($w_e \geq \beta_e \geq 0, \sigma = 0$), the thin solid arcs represent fare evasion ($\beta_e = 0, \sigma_e > 0$), and the dashed arc is a detour path from s to t that avoids the paying network ($\beta_e = \sigma_e = 0$).

The strategy consisting in taking the long evasion path with weights $(5, 1, 0)$ is clearly dominated and will never be used by the network user. Therefore, there is a NE in which the inspector does not control this arc. A Nash strategy for the inspector is thus to defend the evasion arcs $s \rightarrow u$ and $u \rightarrow t$ at 100%

each, so that the user's best response is to take the free road from s to t (the dashed arc with weights $(3, 0, 0)$), giving no revenue to the inspector.

But the inspector has a better strategy. If he defends the evasion arc $s \rightarrow u$ and the long evasion arc $s \rightarrow t$ at 100% each, the network user has two possible best responses: the path $s \rightarrow u \rightarrow t$ (paying the toll on $s \rightarrow u$ and evading the toll on $u \rightarrow t$), and the free road from $s \rightarrow t$, each having a cost of 3 with respect to the weights $\mathbf{w} + \boldsymbol{\sigma} \circ \mathbf{q}$. The tie-breaking best response is the former path, which yields a revenue of 1 to the inspector (independently of the value of $\alpha \in [0, 1]$ because the revenue comes from the toll between s and t , not from penalties). This shows that this instance has an infinite *PoS*.