Lecture #16 Notes Summary

Spot checking games, modelling tricks, Computation of Nash and Stackelberg equilibria.

Spot Checking Games

In this lecture, we define the class of spot checking games, which can be used to study and optimize the control tours of fare inspectors over a transportation network.

> **Definition 1.** 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.

Note that in this model, every user that is controlled must pay a fine. This is because the strategies of the users are completely represented by their paths in the network. In particular, honnest users who pay a fare chose a $(s_k, d_k)-$path P in the graph such that $\sigma_e = 0$ for all $e \in P$ (so they are not contolled), but the user costs w_e on these edges now comprise the ticket fare.

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 $\mathcal{R}_k$. We denote by $\hat{p}_R^k$ the proportion of $k-$users taking path $R \in \mathcal{R}_k$.

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 a trip on path R is $\sum_{e \in R} q_e\sigma_e$. The total expected cost of Player k can now be expressed as:

$$\mathrm{Payoff}_k(\hat{\boldsymbol{p}}, \boldsymbol{q}) = -\sum_{R \in \mathcal{R}_k} \hat{p}_R^k \left(\sum_{e \in R} w_e + q_e\sigma_e P \right), \tag{1}$$

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 however, the quantity $\sum_{e \in R} q_e\sigma_e$ can be seen as a fair approximation of the probability $\pi_R := 1 - \prod_{e \in R}(1 - q_e\sigma_e)$ to be controlled over R.

Inspector'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:

$$\mathrm{Payoff}_0(\hat{\boldsymbol{p}}, \boldsymbol{q}) = \sum_k x_k \sum_{R \in \mathcal{R}_k} \hat{p}_R^k \left(\sum_{e \in R} \beta_e + \alpha q_e \sigma_e P \right). \tag{2}$$

The extreme values of α correspond to two important situations. If $\alpha = 1$, the payoff defined in (2) 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.

If we assume (for simplicity) that $\mathcal{Q}$ is the standard simplex Δ_E, i.e. $\boldsymbol{q} \in \mathcal{Q}$ is a probability distribution over the edges of the graph (which can be well suited if there is a single fare inspector, who controls edge e with probability q_e), we will see in exercise 3 of Worksheet #16 that the game $\mathcal{G}$ is a star-shaped polymatrix game, with the fare inspector as the central player. If we further assume that $\alpha = 1$ (MAXPROFIT), then we can rewrite $\mathcal{G}$ as a pairwise zero-sum polymatrix game. Hence, it is not surprising that a NE of this game can be computed by a LP. However, with this approach we would need to enumerate all the paths $R \in \mathcal{R}_k$, which is not a possible option for most realistic networks. We will present a more efficient way to solve this problem.

Modelling tricks

A public transportation network model

We assume that network users travel over a network $G_0 = (V_0, E_0)$ with edge costs w_e (representing time) 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 an extended graph G with 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 $d \in D = \{d_k : k \in \mathcal{K}_0\}$, and a set of new edges $\bar{E}$ connecting these new vertices, cf. Figure 1. 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 $\mathcal{K} = \{(\bar{s_k}, \bar{\bar{d_k}}) : k \in \mathcal{K}_0\}$.

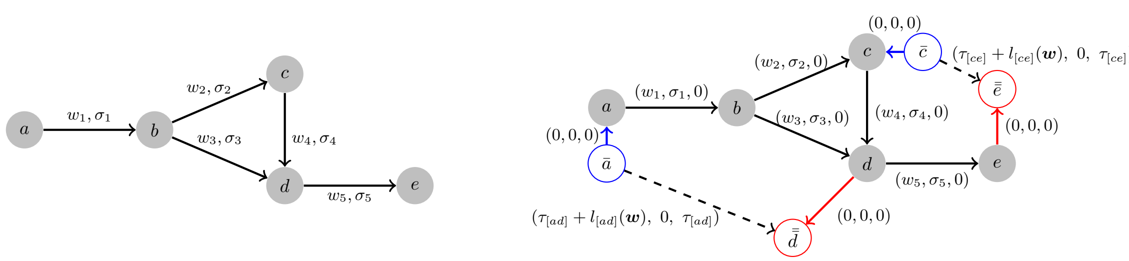

Figure 1: Original transit network G_0 (above), and extended structure G of the NSC game (below). The edges of G are labelled with triples (w_e, σ_e, β_e). The subset of commodities is $\mathcal{K}_0 = \{[ad], [ce]\}$, which becomes $\mathcal{K} = \{[a\bar{d}], [c\bar{e}]\}$ in the extended graph.

We will see in exercise 2 of Worksheet #16 that it is also possible to handle the case where users can buy a ticket for a part of their trip only.

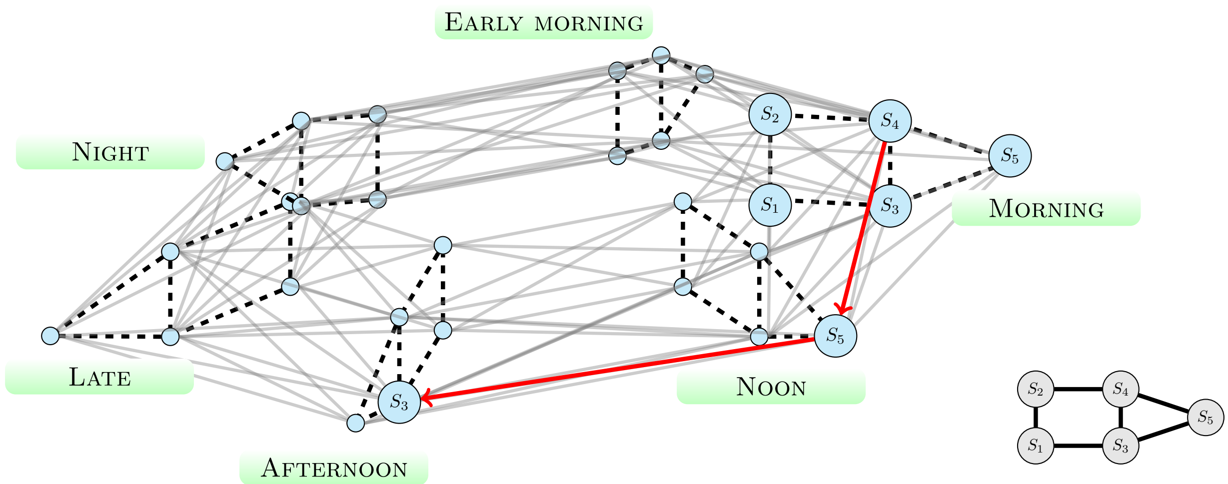

Figure 2: Example for a graph C connecting the control areas (lower right corner) and its associated cyclic duty graph D (main drawing), for a time discretization of one day with $T = 6$ time windows. The path highlighted in red represents the duty of a team controlling S_4 during the morning, S_5 at noon and S_3 during the afternoon.

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 $\mathcal{Q} = \Delta_E$ might not be well-suited to represent all possible marginal strategies of the controller.

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 window $t \in \mathcal{T}$. We denote by $G_0 = (E_0, V_0)$ the graph representing the static problem (e.g. the public transportation network model of last section), 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:

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

We have depicted in Figure 2 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 (explained during the lecture), 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 $\tilde{D}$.

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

$$\forall v \in \tilde{V}, \quad \sum_{a' \in \delta^+(v)} \tilde{q}_{a'} - \sum_{a \in \delta^-(v)} \tilde{q}_a = \begin{cases} \gamma & \text{if } v = d_s; \\ -\gamma & \text{if } v = d_t; \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

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

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

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

$$\hat{q}_{(S,t)} = \sum_{l=1}^{L} \sum_{a \in \delta^-(S,t,l)} \tilde{q}_a. \tag{4}$$

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:

$$\begin{aligned} \mathcal{Q} = \Big\{ \boldsymbol{q} \in & (\mathbb{R}_+)^E : \; \exists \tilde{\boldsymbol{q}} \in (\mathbb{R}_+)^{\tilde{A}} \text{ s.t.} \\ & (i) \; \tilde{\boldsymbol{q}} \text{ satisfies the flow conservation (3);} \\ & (ii) \; \forall (e,t) \in E, \\ & \qquad q_{(e,t)} \leq \sum_{\{S \in \mathcal{S} \colon S \ni e\}} \kappa_{e|S} \sum_{l=1}^{L} \sum_{a \in \delta^-(S,t,l)} \tilde{q}_a; \\ & (iii) \; \forall (e,t) \in E, \quad q_{(e,t)} \leq 1 \Big\}. \end{aligned}$$

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.

Nash equilibrium of MAXPROFIT

The next result characterizes the best response of the network users:

Proposition 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 $\hat{\boldsymbol{p}}^{\boldsymbol{k}}$ for Player k is a best response to $\boldsymbol{q}$ if and only if $-\mathrm{Payoff}_k(\hat{\boldsymbol{p}}, \boldsymbol{q}) = \lambda_k(\boldsymbol{q})$. 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 $\boldsymbol{p}^{\boldsymbol{k}}$ is a flow of unit value through commodity k, then we have

$$-\mathrm{Payoff}_k(\hat{\boldsymbol{p}}, \boldsymbol{q}) = \sum_{R \in \mathcal{R}_k} \hat{p}_R^k \sum_{e \in R} c_e(\boldsymbol{q}),$$

which is 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 $\hat{\boldsymbol{p}}^{\boldsymbol{k}}$ uses only shortest paths. □

We next show that in the case $\alpha = 1$ (MAXPROFIT), the game $\mathcal{G}$ has the same Nash equilibria as a zero-sum game.

Proposition 2 (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:

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

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

$$BR_0^{\mathcal{G}}(\hat{\boldsymbol{p}}) = \arg\max_{\boldsymbol{q} \in \mathcal{Q}} \sum_k x_k \sum_{R \in \mathcal{R}_k} \hat{p}_R^k \left(\sum_{e \in R} \beta_e + q_e \sigma_e P \right).$$

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

$$BR_0^{\mathcal{G}}(\hat{\boldsymbol{p}}) = \arg\max_{\boldsymbol{q} \in \mathcal{Q}} \sum_k x_k \sum_{R \in \mathcal{R}_k} \hat{p}_R^k \left(\sum_{e \in R} w_e + q_e \sigma_e P \right) = BR_0^{\mathcal{G}'}(\hat{\boldsymbol{p}}).$$

□

The problem of finding a NE of $\mathcal{G}$ hence reduces to the problem of finding a NE of $\mathcal{G}'$. In this game, a NE strategy of the inspectors must maximize the loss of the users (because the game is zero-sum), and we know from Proposition 1 that the loss must be equal to $\lambda_k(\boldsymbol{q})$, so it is obtained by solving

$$\max_{\boldsymbol{q} \in \mathcal{Q}} \quad \sum_k x_k \lambda_k(\boldsymbol{q}) \tag{5}$$

It is a standard result from graph theory that the shortest path problem has a LP formulation, in which the objective function must be maximized. It is obtained by introducing node potentials y_v^s so that the shortest path between s and v is at least y_v^s. Combining these LPs with Problem (5), we obtain:

$$\max_{\boldsymbol{q},\boldsymbol{y}} \quad \sum_{k\in\mathcal{K}} x_k y_{d_k}^{s_k} \tag{6a}$$

$$\text{s.t.} \quad y_v^s - y_u^s \leq w_e + \sigma_e q_e P, \quad \forall s \in S,\ \forall e \equiv (u,v) \in E; \tag{6b}$$

$$y_s^s = 0, \quad \forall s \in S; \tag{6c}$$

$$\boldsymbol{q} \in \mathcal{Q}. \tag{6d}$$

We point out that the dual variables of inequalities (6b) define a (multicommodity) flow, from which it is possible to construct the Nash strategy $\hat{p}_R^k$ of the network users.

Computation of a Stackelberg Equilibrium (with the Inspector as leader)

The computation of a Stackelberg equilibrium is a hard problem. It can be shown that in general, the problem of finding a SE of a network spot-checking game $\mathcal{G}$ is NP-hard. Therefore, there is no hope to find a LP formulation.

However, we next show that a SE can be computed by solving a so-called *mixed integer programming* (MIP) problem, that is, a LP in which some of the variables are constrained to be integer-valued.

To do this, we define the arc-flow strategies

$$p_e^k = \sum_{R\in\mathcal{R}_k} \hat{p}_R^k,$$

which lie in the polyhedron defined by the following flow conservation constraints:

$$\begin{aligned} \sum_{e'\in\delta^+(v)} p_{e'}^k - \sum_{e\in\delta^-(v)} p_e^k = \begin{cases} 1 & \text{if } v = s_k; \\ -1 & \text{if } v = d_k; \\ 0 & \text{otherwise.} \end{cases} \qquad & \forall v \in V,\ \forall k \in \mathcal{K} \\ p_e^k \geq 0, \qquad & \forall e \in E,\ \forall k \in \mathcal{K} \end{aligned} \tag{7}$$

It is easy to see that the payoffs can be rewritten using these arc-flow strategies only:

$$\begin{aligned} \text{Payoff}_k(\boldsymbol{p},\boldsymbol{q}) &= -\left(\sum_{e\in E} p_e^k w_e + \sum_{e\in E} p_e^k q_e \sigma_e P\right), \\ \text{Payoff}_0(\boldsymbol{p},\boldsymbol{q}) &= \sum_k x_k \sum_{e\in E} p_e^k (\alpha\sigma_e q_e P + \beta_e). \end{aligned} \tag{8}$$

By Proposition 1, we know that the flow of Player k must be concentrated on the edges that belong to a shortest path tree rooted in s_k in the graph $G = (V, E, \boldsymbol{c}(\boldsymbol{q}))$. This can be enforced by the use of *big-M constraints*. The next proposition shows how we can modify the Nash LP (6) into a general MIP for the computation of a SE.

Proposition 3. *Let $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\rho})$ be a solution of the following MIP:*

$$\max_{\boldsymbol{q},\boldsymbol{y},\boldsymbol{\mu},\boldsymbol{\rho}} \quad \sum_{k\in\mathcal{K}} \alpha x_k y^{s_k}_{d_k} + \sum_{s\in S}\sum_{e\in E} \rho^s_e(\beta_e - \alpha w_e) \tag{9a}$$

s. t.

$$0 \le w_e + \sigma_e q_e P - (y^s_v - y^s_u) \le M_e(1-\mu^s_e), \qquad \forall s\in S,\ \forall e\equiv(u,v)\in E; \tag{9b}$$

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

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

$$\sum_{e'\in\delta^+(v)} \rho^s_{e'} - \sum_{e\in\delta^-(v)} \rho^s_e = \begin{cases} \sum_{k\in\mathcal{K}_s} x_k & \text{if } s=v;\\ -x_{(s,v)} & \text{if } (s,v)\in\mathcal{K}_s;\\ 0 & \text{otherwise,}\end{cases} \qquad \forall s\in S,\ \forall v\in V; \tag{9e}$$

$$0 \le \rho^s_e \le M^s \mu^s_e, \qquad \forall s\in S,\ \forall e \in E; \tag{9f}$$

$$\mu^s_e \in \{0,1\}, \qquad \forall (s,e)\in S\times E. \tag{9g}$$

Then, $\boldsymbol{q}$ is a Stackelberg strategy of the inspector. Moreover, a pure stackelberg strategy of the users exists, and for all k it consists of a path $R\in\mathcal{R}_k$ of minimal length in $G=(V,E,\boldsymbol{c}(\boldsymbol{q}))$, and in case of a tie R must be of maximal length for the weights $c'_e(\boldsymbol{q}) := \alpha\sigma_e q_e P + \beta_e$.

Proof. First note that the second part of the proposition (pure Stackelberg strategies for the followers) is a direct consequence of Theorem 3 in the lecture on polymatrix games.

As in Problem (6), constraints (9b)-(9c) 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 (9d) 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 (9b) for a large constant M_e). Indeed, the first inequality in (9b) 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 (9e) forces $\boldsymbol{\rho^s}$ to be a single-source multi-sink flow rooted in s (for a large constant M^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 (9f) 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 Proposition 1. Finally, the objective function (9a) rewrites to the controller's payoff (2) when replacing $y^{s_k}_{d_k}$ and ρ^s_e by their values as a function of p^k_e:

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

□