

Masterarbeit

# Bilevel Programming to Optimize the Use of Traffic Control Gantries for Toll Enforcement

Stephan Schwartz

27. November 2013

Erstgutachter: Prof. Dr. Ralf Borndörfer

Zweitgutachter: Dr. Guillaume Sagnol





# Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Optimal Placement of Traffic Control Gantries</b>	<b>3</b>
2.1. First Approaches from Literature . . . . .	4
2.2. Placement of Gantries with a Fixed Budget . . . . .	6
2.2.1. The Cover Approach . . . . .	9
2.2.2. Computing an Optimal Placement . . . . .	13
2.3. Placement with an Indefinite Number of Control Gantries . . . . .	15
2.3.1. The $d\%$ -Cover . . . . .	16
2.3.2. Placement of Profitable Control Gantries . . . . .	17
<b>3. The Optimal Control Strategy</b>	<b>19</b>
3.1. The Game Theoretic Model . . . . .	20
3.2. A Probabilistic Cover . . . . .	23
3.2.1. The pCover Approach . . . . .	23
3.2.2. Computation of the Inspectors' Strategy . . . . .	26
3.3. pCover and Perception . . . . .	33
3.4. pCover vs. $\alpha$ -pCover . . . . .	37
3.4.1. Comparing the Objective Values . . . . .	37
3.4.2. The Perception Gap . . . . .	38
3.4.3. Influence of $\alpha$ . . . . .	41
<b>4. Application to German Motorways</b>	<b>43</b>
4.1. Placing Control Gantries on German Motorways . . . . .	43
4.2. Optimal Control Strategies on German Motorways . . . . .	49
<b>5. Some Further Problems</b>	<b>55</b>
5.1. Conducting Controls according to $q$ . . . . .	55
5.2. Combining Stationary and Mobile Controls . . . . .	59
<b>A. Appendix</b>	<b>61</b>
A.1. The Cut Problem is NP-hard for $k \geq 2$ . . . . .	61
A.2. Computing $R^i$ with Yen's Algorithm . . . . .	62
<b>Bibliography</b>	<b>64</b>



# 1. Introduction

In January 2005 a distance based toll for trucks on German motorways was introduced. Since then, trucks weighing twelve tonnes or more have to pay a toll which depends on the emission category, the number of axles and the distance travelled on the toll route. The German Federal Office for Goods Transport (BAG) enforces the toll, using 300 traffic control gantries for automatic stationary controls and 300 control vehicles for mobile inspections.

In a current project of the Zuse-Institut Berlin (ZIB) in cooperation with the BAG, the goal is to optimize the toll enforcement based on mathematical models. Previous work focused on optimizing the tours of the mobile units. Therefore, [BSS11], [BOSS12] and [BBSS12] present several approaches to optimize the use of the controllers and to model the behavior of the drivers.

This thesis aims at studying another element of the toll control system and is completely dedicated to the optimal use of traffic control gantries. This includes the optimal placement of the control gantries (Chapter 2), the optimization of the control strategy (Chapter 3) and the combination with a mobile control strategy (Chapter 5). Despite the well defined project to ensure the toll enforcement on German motorways, we will develop models that are applicable to arbitrary motorway networks.

The specialty of our problem is that we consider a security issue on a network, but that our control possibilities are restricted to certain points in the network, the locations of the control gantries. Additionally, users in this network are willing to avoid a control and may evade on paths where we have no control possibility, e.g. trunk roads. These constraints make our problem unique and force us to develop new mathematical models to optimize the toll control.

We start with the important question where to locate a certain number of control points in a network. Proceeding from the assumption that traffic control gantries are not yet installed, the optimal placement of the gantries is one of the most

fundamental questions when planning an automatic toll control system. We will study this problem in Chapter 2 and develop our main approach, the cover approach, which can be formulated as an integer program.

With regard to the present project our focus lies on the third chapter where we want to find an optimal control strategy. A foremost issue in this context is the behavior of the drivers. Similar to [BOSS12] and [BSS12] we develop a game theoretic model where the drivers are playing against the inspectors. While the inspectors try to maximize their revenues, the drivers want to minimize their expenses. At first we assume that the drivers perfectly observe the control strategy and act accordingly. We derive the pCover approach to describe this game between inspectors and drivers and formulate it as a linear program. Afterwards we consider the case where the drivers misjudge the probability to be controlled and thus decide suboptimally. We adapt an approach from literature to model the perception of the drivers and formulate the  $\alpha$ -pCover approach. We conclude the chapter with an analysis of the  $\alpha$ -pCover and a theoretical comparison of the two approaches.

In Chapter 4 we apply the main models to the real traffic network of German motorways. We test the cover approach to place the control gantries optimally and compare the result with the actual placement of the gantries. In the second part, we run the pCover and the  $\alpha$ -pCover on this instance and compute an optimal control strategy for German motorways.

Chapter 5 contains two important aspects that have not been considered yet. We will develop and analyze a possibility to conduct toll controls according to our control strategy using basic stochastics. We close our study of stationary toll control with an examination of the integrability of mobile and stationary controls. The goal is to find a combined strategy that leads to a coordinated and more efficient toll control.

## 2. Optimal Placement of Traffic Control Gantries

The automatic toll control on German motorways is a user-friendly way of toll enforcement. In classical toll enforcement systems the drivers have to pass toll stations which leads to significant breaks. The absence of toll stations at every exit is convenient and money-saving not only for the drivers but also for the enforcing agency. The installation of several traffic control gantries is much cheaper than the building, operation and maintenance of many toll stations.

If we are planning such an automatic toll control system without barriers, the first question is where to build the traffic control gantries. The placement of the gantries is crucial for an efficient and broad toll control.

It turns out that it is hard to find a suitable mathematical model to optimize the placement of traffic control gantries in a traffic network. Hence we dedicate a whole chapter to this important question and consider two scenarios. In the first one, we have a fixed budget that allows us to install a certain number  $m$  of control gantries. The second scenario is about determining an appropriate number of gantries and placing them optimally.

Our main result is the *cover approach* which we will develop from two well-known combinatorial optimization problems, the set cover and the directed multicut problem. After formalizing the traffic network, we will present these two problems and formulate a first approach, the *cut approach*. As this approach still ignores important issues of a good placement, we will extend it to the cover approach. In the final section of this chapter we will derive two models from the cover approach to optimize the placement of an indefinite number of traffic control gantries.

**Formalization of the Traffic Network** The motorway network is represented by a directed graph  $G = (V, E)$ , where the nodes represent the exits and the edges represent the sections of the motorway. Every edge  $e \in E$  has a rational length  $l_e > 0$ , which is the length of the corresponding section. Furthermore, we are given

a set of commodities  $\mathcal{K}$ , where each commodity  $k_i \in \mathcal{K}$  is a triple  $k_i = (s_i, t_i, x_i)$ . Here,  $s_i, t_i \in V$  are the source and sink, respectively, of commodity  $k_i$  while  $x_i \in \mathbb{N}$  is the demand of  $k_i$ , meaning that  $x_i$  drivers want to go from  $s_i$  to  $t_i$  in  $G$  per time unit.

Installing a control gantry at section  $e \in E$  means that  $e$  becomes a *control edge* since it does not matter where on  $e$  the gantry is placed. We assume that every traffic control gantry has an unbounded control capacity, which means that all vehicles using this edge can be controlled. Our goal is now to specify a suitable subset  $\mathcal{C} \subseteq E$  as control edges.

## 2.1. First Approaches from Literature

If we assume that for every commodity  $k_i$  the drivers take a certain path  $P^i$ , our goal is to *cut* this path by placing a control gantry at one of its edges. The problem of finding a smallest set of edges  $M \subseteq E$  that cuts all paths  $P^i$  can be formulated as a *set cover*. To see this we follow the notation of [KV06] and define

$$U := \bigcup_{i=1}^{|\mathcal{K}|} P^i \quad \text{and} \quad S_e := \{P^i \in U : P^i \cap e \neq \emptyset\}.$$

The set cover problem is now to find a smallest subset  $\mathcal{R}$  of  $\mathcal{S} := \bigcup_{e \in E} S_e$  such that  $U$  is the union of all sets in  $\mathcal{R}$ . Since  $S_e$  is the set of paths that are cut if we install a control gantry at  $e$ , this instance of set cover describes the above problem of cutting all paths  $P^i$ .

In a second attempt we could restrict ourselves to sets  $\mathcal{R} \subseteq \mathcal{S}$  with  $|\mathcal{R}| = m_0$  for a fixed  $m_0$  and find such an  $\mathcal{R}$  which cuts the most traffic. This can be formulated as a *weighted maximum coverage problem*, where we are given a finite set  $U$  and each element  $u \in U$  has a weight  $w(u) \geq 0$ . Furthermore, we are given a set  $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$  of subsets of  $U$  and a number  $k$ . The goal is to find a set  $\mathcal{R}^* \subseteq \mathcal{S}$ ,  $|\mathcal{R}^*| = k$  such that the weight of the set  $\bigcup_{S_i \in \mathcal{R}^*} S_i$  is maximal. In order to formulate our problem as a weighted maximum coverage problem, we choose  $U$  and  $\mathcal{S}$  as in the set cover. Then we define  $w(P_i) := x_i$  (the demand of  $k_i$ ) and set  $k = m_0$ . The optimal set  $\mathcal{R}^*$  for this weighted maximum coverage problem, or rather the respective edge set  $\{e \in E : S_e \in \mathcal{R}^*\}$ , is optimal for our problem of cutting the paths with the most traffic.

Both these problems, the set cover and the weighted maximum coverage problem,

are known to be hard in terms of complexity. The set cover problem belongs to the famous 21 problems that Richard Karp proved to be NP-complete in [Kar72]. As there is a straightforward reduction to the (weighted) maximum coverage problem in polynomial time, the latter is also NP-hard.

These introductory approaches show how we can transform the problem of placing control gantries on motorways into cutting certain paths in the traffic network. The assumption, however, that the drivers of commodity  $k_i$  may only take a single path in  $G$  is unrealistic. The next approach from literature fixes this and considers all possible paths in  $G$  [CLR05].

In the *directed multicut problem* we are given a digraph  $G = (V, E)$  with source-sink pairs  $(s_1, t_1), \dots, (s_k, t_k)$ . A multicut  $M \subseteq E$  is a set of edges, such that for all  $i$  there is no  $s_i - t_i$ -path in  $(V, E \setminus M)$ . The directed multicut problem is to find a multicut of minimal size in  $(G, (s_1, t_1), \dots, (s_k, t_k))$ .

For  $k = 1$  the famous maxFlow-minCut theorem by Ford and Fulkerson [FF62] states that the size of a minimum multicut equals the maximum number of disjoint  $s_1 - t_1$ -paths in  $G$  and hence can be computed in polynomial time.

#### **directed multicut problem**

Input:  $(G, (s_i, t_i)_{i=1, \dots, k}, n_0)$ .

Decision problem: Is there a directed multicut of size  $\leq n_0$ ?

However, for  $k \geq 2$  the respective decision problem becomes NP-hard [GVY94]. As the directed multicut problem lays the ground for our models, we should study further results for this problem. The approach via the duality of flows and cuts that solved the problem for  $k = 1$  cannot be used to find a good approximation algorithm for  $k \geq 2$ . Relaxation of the multicut problem leads to the fractional multicut problem, but the integrality gap can be as large as  $k$ , the number of commodities [SSZ04]. So far, all results on approximation algorithms for the directed multicut problem are unsatisfactory. [CLR05] gives an excellent survey on the topic and the meager results on approximation. It should be mentioned that there are better results for the multicut problem for undirected graphs. It is NP-hard only for  $k \geq 3$  [DJP<sup>+</sup>92] and an  $\mathcal{O}(\log k)$ -approximation algorithm is presented in [GVY93]. The directed multicut problem is a good starting point for our model, but still not optimal as we consider all possible paths for the drivers. The drivers of  $k_i$  may choose an  $s_i - t_i$ -path in  $G$ , but it is obvious that they will not take an arbitrary

one. Instead, we assume that they avoid long detours and that the length of those detours depends on the length of a shortest  $s_i - t_i$ -path. If not specified, a shortest path is always with respect to  $l$  and thus a path  $P$  has length  $l(P) := \sum_{e \in P} l_e$ . Now let  $l^i$  be the length of a shortest  $s_i - t_i$ -path, then we assume that the drivers will not make a detour longer than  $\varrho \cdot l^i$  for some  $\varrho > 0$ . Therefore, every driver of commodity  $k_i$  will choose a path from

$$R^i = \{s_i - t_i - \text{paths } P \text{ in } G : l(P) \leq (1 + \varrho) \cdot l^i\}.$$

The elements of  $R^i$  are called routes and are denoted by  $r_1^i, r_2^i, \dots, r_{n_i}^i$ . The number of routes for  $k_i$  and thus our whole model strongly depends on the choice of  $\varrho$ . If we overestimate the detour that the drivers are willing to take, we might use more control capacity than necessary. If we underestimate it, however, the drivers might evade our control even though we invested a lot of control capacity, which would be more severe.

The sets  $R^i$  are the keys for our formulation. Instead of cutting a single path  $P_i$  or all possible paths, we cut the routes of commodity  $k_i$ , i.e. all paths in  $R^i$ .

## 2.2. Placement of Gantries with a Fixed Budget

For our purposes we use a *cut approach* similar to the directed multicut. We suppose that we have a fixed budget for the installation of control gantries and that we can build a certain number  $m$  of control gantries with this budget. More formal, this means that we can define a set  $\mathcal{C} \subseteq E$  of control edges with  $|\mathcal{C}| = m$ . Then we say:

$$k_i \in \mathcal{K} \text{ is cut} : \iff \forall r_j^i \in R^i : r_j^i \cap \mathcal{C} \neq \emptyset.$$

This means that a commodity is *cut* if every route of it contains at least one control edge. Our goal is now to place  $m$  control gantries in such a way that we maximize the number of drivers who have to use a control edge. The cut problem is a combinatorial optimization problem and can be formulated as an integer program (IP).

$$\max_{y, \delta} \sum_{i=1}^{|\mathcal{K}|} x_i \cdot \delta_i \quad (2.1a)$$

$$\text{s.t.} \quad \sum_{e \in \mathcal{C}} y_e \leq m \quad (2.1b)$$

$$\delta_i \leq \sum_{e \in r_j^i} y_e \quad \forall r_j^i \in R^i, \quad \forall i \quad (2.1c)$$

$$y_e \in \{0, 1\} \quad \forall e \in E \quad (2.1d)$$

$$\delta_i \in \{0, 1\} \quad \forall i \in \{1, \dots, |\mathcal{K}|\} \quad (2.1e)$$

Obviously,  $y_e = 1$  means that we install a control gantry at  $e \in E$ , while  $\delta_i = 1$  means that  $k_i$  is cut. (2.1b) guarantees that at most  $m$  control gantries are built, and the objective function (2.1a) maximizes the amount of traffic that is cut. As we try to maximize  $\sum_i x_i \cdot \delta_i$ , we have to enforce  $\delta_i = 0$  whenever there is a route  $r_j^i \in R^i$  without control edge. In this case  $\sum_{e \in r_j^i} y_e = 0$  and hence  $\delta_i = 0$  is ensured by (2.1c). Furthermore, we have to ensure that  $\delta_i = 1$  is feasible, if  $k_i$  is cut. Then,  $\sum_{e \in r_j^i} y_e \geq 1$  for all  $r_j^i$  and hence  $\delta_i = 1$  is feasible. Finally, the integrality constraints (2.1d) and (2.1e) guarantee that there are no such things as partial control gantries or partial cuts.

With the results on the complexity of the directed multicut problem, we can easily achieve a complexity result for our cut problem.

### cut problem

Input:  $(G, l > 0, \varrho, (s_i, t_i, x_i)_{i=1, \dots, |\mathcal{K}|}, n_1, d)$ .

Decision problem: Is there an edge set of size  $n_1$  that cuts a total traffic of  $\geq d$ ?

For  $|\mathcal{K}| \geq 2$  this is NP-hard to decide, since otherwise we could efficiently decide the directed multicut problem by choosing  $\varrho$  sufficiently large and setting  $n_1 = n_0$  and  $d = \sum_i x_i$ . A formal reduction is given in Appendix A.1. However, the complexity is not the main problem of the cut approach. It is particularly inapplicable because it does not take central issues for a good placement into account. When modelling the placement of the control gantries, we should be oriented towards the following criteria:

- i) enable a broad control
- ii) prefer long sections for control gantries
- iii) build multiple control gantries on long routes

The first criterion means that the placement should not be focused on a few commodities, but allows the inspectors to control many commodities or, in our case, to control as much traffic as possible. ii) and iii) both point out a very important aspect that we have not considered yet. When a driver of commodity  $k_i$  chooses an  $s_i - t_i$ -path, we have to face the problem that in real life he might make a detour to avoid certain traffic control gantries. However, this detour is not necessarily restricted to the motorway, but can also include the use of trunk roads.

This particularly means that we have to consider possible trunk road detours in our traffic network. For the sake of simplicity we assume that every edge  $e \in E$  can be avoided by making a detour on a parallel trunk road edge  $e'$  and that the length of  $e'$  depends only on the length of  $e$ . More specifically, we consider a detour function  $det : \mathbb{Q}_{>0} \rightarrow \mathbb{Q}_{>0}$ , and assume  $l_{e'} = det(l_e)$ .

There are numerous ways to define a detour function in a proper way. An affine function could be a good choice, for example. Then we have fixed costs (expressed in kilometers) for the drivers to leave and reenter the highway and, additionally, we have increased travelling costs for the detour on the trunk road. This increase of costs originates from the fact that this trunk road might be much longer than the motorway section. Furthermore, the drivers have to drive slower and have a higher fuel consumption for they frequently have to adjust their speed.

Thus, compared to these increased travelling costs, the costs for leaving the highway should be rather small. To simplify our model, we ignore them and assume that the detour function is linear. More specifically, we set  $l_{e'} = u \cdot l_e$  where  $u > 1$  is a constant detour factor. Then we define  $E'$  to be the set of all detour edges and thus  $G' = (V, E \cup E')$  is the traffic network which contains the detour edges.

After adding detour edges to our network, we can reconsider the criteria ii) and iii) from above. If we install a control gantry on a rather short edge, the drivers could take the detour edge and avoid the control gantry with low additional costs, which shows that ii) is a reasonable condition. The argument for iii) is similar. The detour to avoid a single control gantry on a long route will hardly be significant compared to the total length of the route. Both criteria become further significant, when we evaluate the actual placement of control gantries on German motorways.

While the average length of a section with control gantry is 8.5 km, sections without control gantries are only half as long on average, namely 4.3 km. Furthermore, on long motorways ( $>250$  km) we have 1.3 control gantries per 100 km. Those numbers show that the avoidance of control gantries by leaving the motorway is an actual problem for the enforcing agency. Clearly, our next approach has to take this into account.

### 2.2.1. The Cover Approach

Since we assume that the drivers of  $k_i$  are willing to take any route with length at most  $(1 + \varrho) \cdot l^i$ , we have to improve the cut approach in order to meet the criteria from above. Our solution is the *cover approach* and allows the drivers to avoid control gantries by using trunk roads. For  $\mathcal{C} \subseteq E$  with  $|\mathcal{C}| = m$ , a commodity  $k_i$  is called *covered* if every  $s_i - t_i$ -path  $P'$  in  $G'$  with  $l(P') \leq (1 + \varrho) \cdot l^i$  contains a control edge  $e \in \mathcal{C}$ . In other words,  $k_i$  is covered if the drivers would have to make a long detour to avoid all control gantries. This definition requires that we consider paths in  $G'$  and is essentially a cut approach where we extended  $R^i$  with paths in  $G'$ . However, extending  $R^i$  with paths in  $G'$  would blow up the size of  $R^i$  dramatically. Fortunately, it suffices to work with  $G$ , as the following proposition shows.

**Proposition 2.1** *For  $\mathcal{C} \subseteq E$  with  $|\mathcal{C}| = m$ , we have*

$$k_i \in \mathcal{K} \text{ is covered} \iff \forall r_j^i \in R^i : \sum_{e \in r_j^i} l_e \cdot \left(1 + (u - 1) \cdot \chi_{\mathcal{C}}(e)\right) > (1 + \varrho) \cdot l^i,$$

where  $\chi_{\mathcal{C}}$  is the indicator function.

*Proof.* “ $\implies$ ” Assume that  $k_i \in \mathcal{K}$  is covered and consider a path  $P \in R^i$ . We define a path  $P'$  in  $G'$  by replacing every control edge  $e \in P \cap \mathcal{C}$  by its respective detour edge  $e'$  and state

$$l(P') = \sum_{e \in P' \cap E} l_e + \sum_{e' \in P' \cap E'} l_{e'} = \sum_{e \in P} l_e \cdot \left(1 + (u - 1) \cdot \chi_{\mathcal{C}}(e)\right).$$

Since  $k_i$  is covered and  $P'$  contains no control edge, we know that  $l(P') > (1 + \varrho) \cdot l^i$ .

“ $\impliedby$ ” Suppose that  $k_i$  is not covered. Then there is an  $s_i - t_i$ -path  $P'$  in  $G'$  with  $l(P') \leq (1 + \varrho) \cdot l^i$  and without a control edge. Similar to above we define

a path  $P$  in  $G$  by replacing every detour edge  $e' \in P' \cap E'$  by its parallel edge  $e \in E$ . Now we have  $l(P) \leq l(P') \leq (1+\varrho) \cdot l^i$  and hence  $P \in R^i$ . Furthermore, as  $P'$  contains no control edge, we get

$$(1 + \varrho) \cdot l^i \geq l(P') \geq \sum_{e \in P} l_e \cdot \left(1 + (u - 1) \cdot \chi_{\mathcal{C}}(e)\right),$$

which completes the contraposition.  $\square$

With this proposition we could prevent a blow-up of the size of our traffic network when considering the cover problem. Our goal hereby is to maximize the covered traffic by placing  $m$  control gantries optimally. Just like in the cut approach we have a combinatorial optimization problem and formulate it as an IP.

$$\max_{y, \delta} \sum_{i=1}^{|\mathcal{K}|} x_i \cdot \delta_i \tag{2.2a}$$

$$\text{s.t.} \quad \sum_{e \in \mathcal{C}} y_e \leq m \tag{2.2b}$$

$$\sum_{e \in r_j^i} l_e \cdot \left(1 + (u - 1) \cdot y_e\right) \geq \delta_i \cdot (1 + \varrho + \varepsilon) \cdot l^i \quad \forall r_j^i \in R^i, \forall i \tag{2.2c}$$

$$y_e \in \{0, 1\} \quad \forall e \in E \tag{2.2d}$$

$$\delta_i \in \{0, 1\} \quad \forall i \in \{1, \dots, |\mathcal{K}|\} \tag{2.2e}$$

The objective of the cover-IP is to maximize the amount of covered traffic and thus  $\delta_i = 1$  means now that  $k_i$  is covered. The only difference of (2.2) from the cut-IP (2.1) is (2.2c), where we ensure that  $\delta_i = 1$  only if  $k_i$  is covered. Therefore,  $\varepsilon > 0$  should be chosen sufficiently small and then, if  $k_i$  is not covered, we have

$$\sum_{e \in r_j^i} l_e \cdot \left(1 + (u - 1) \cdot \chi_{\mathcal{C}}(e)\right) \stackrel{\text{Prop. 2.1}}{\leq} (1 + \varrho) \cdot l^i < (1 + \varrho + \varepsilon) \cdot l^i$$

for an  $r_j^i \in R^i$ . Because  $y_e = \chi_{\mathcal{C}}(e)$ , (2.2c) ensures  $\delta_i = 0$  if  $k_i$  is not covered. In addition,  $\delta_i = 1$  is feasible if  $k_i$  is covered, since then

$$\sum_{e \in r_j^i} l_e \cdot \left(1 + (u - 1) \cdot \chi_{\mathcal{C}}(e)\right) \stackrel{\text{Prop. 2.1}}{>} (1 + \varrho) \cdot l^i \quad \forall r_j^i \in R^i$$

and hence

$$\sum_{e \in r_j^i} l_e \cdot \left(1 + (u - 1) \cdot \chi_C(e)\right) \geq 1 \cdot (1 + \varrho + \varepsilon) \cdot l^i \quad \forall r_j^i \in R^i.$$

The cover approach is quite suitable for our purposes as it meets all of the above criteria i)-iii). Maximizing the number of drivers who cannot evade a control even if they made a short detour leads to an efficient and broad control possibility. The aspect that short detours would not prevent the drivers from evading, together with a shortage of the total control capacity<sup>1</sup>, implies that long sections are preferably chosen to be control edges. This effect can be strengthened by extending the objective function (2.2a) and trying to maximize

$$\left(\sum_i x_i \cdot \delta_i\right) + \mu \cdot \sum_{e \in E} l_e \cdot y_e \quad (2.3)$$

for a small  $\mu > 0$ . In cases where different sections can be chosen without changing the amount of covered traffic, the latter sum breaks the tie in favor of the longest sections.

The cover approach is the fundamental approach of this chapter and serves as the basis for all further approaches in the next section. We will also reconsider it in section 4.1 when we apply the optimization to the real data of the German motorway network and compare the result with the actual placement of the control gantries.

Let us now take a brief look at the objective values of (2.1) and (2.2) which give us the number of drivers that cannot escape a control. This means that there is no route of commodity  $k_i$  without a control edge (cut), or that there is no route of  $k_i$  such that the drivers can avoid all control gantries on it with little detour costs (cover). It is very reasonable to maximize these objective values, but the actual values, however, are not important for us. Nevertheless we can state that the objective value of the cut-IP (2.1) is always greater or equal to the one of the cover-IP (2.2). This is simply due to the following lemma.

**Lemma 2.2** *For every commodity  $k_i \in \mathcal{K}$ :*

$$k_i \text{ is covered} \implies k_i \text{ is cut.}$$

---

<sup>1</sup>This means that  $m$  is not big enough to cover all commodities.

*Proof.* Recall, that  $k_i$  is cut iff

$$\forall s_i - t_i - \text{paths } P \text{ in } G \text{ with } l(P) \leq (1 + \varrho) \cdot l^i : P \cap \mathcal{C} \neq \emptyset,$$

and that  $k_i$  is covered iff

$$\forall s_i - t_i - \text{paths } P \text{ in } G' \text{ with } l(P) \leq (1 + \varrho) \cdot l^i : P \cap \mathcal{C} \neq \emptyset.$$

With  $G \subseteq G'$  the above implication follows.  $\square$

We can also use this lemma to prove a complexity result for the cover problem. Recall that the cut problem is NP-hard for  $|\mathcal{K}| \geq 2$  and consider the cover problem as follows.

**cover problem**

Input:  $(G, l > 0, \varrho, (s_i, t_i, x_i)_{i=1, \dots, |\mathcal{K}|}, n_1, u, d)$ .

Decision problem: Is there an edge set of size  $n_1$  that covers a total traffic of  $\geq d$ ?

By reducing the cut problem to it, we show that the cover problem is NP-hard for  $|\mathcal{K}| \geq 2$  as well. Consider an instance  $(G, l > 0, \varrho, (s_i, t_i, x_i)_{i=1, \dots, |\mathcal{K}|}, n_1, d)$  of the cut problem. We compute  $l_{\min} = \min_e l_e > 0$  and  $l_{\text{sum}} = \sum_e l_e$  in polynomial time and create an instance of the cover problem by adding  $u = \frac{(1+\varrho) \cdot l_{\text{sum}}}{2l_{\min}}$  to the given instance. Now we have for every set of control edges  $M \subseteq E$  and for every  $i$ :

$$k_i \text{ is covered} \iff k_i \text{ is cut.}$$

While “ $\implies$ ” follows directly from Lemma 2.2, our choice of  $u$  allows the proof of the other direction. Therefore, consider a commodity  $k_i$  that is cut and an  $s_i - t_i$ -path  $P$  in  $G'$ . If there is some  $e' \in P \cap E'$  we consider the parallel edge  $e \in E$  and state

$$l(P) \geq l_{e'} = u \cdot l_e = (1 + \varrho) \cdot l_{\text{sum}} \cdot \frac{l_e}{2l_{\min}} \stackrel{l_e \geq l_{\min}}{>} (1 + \varrho) \cdot l_{\text{sum}} \cdot 1 \stackrel{l_{\text{sum}} \geq l^i}{\geq} (1 + \varrho) \cdot l^i.$$

Thus we have  $\forall s_i - t_i - \text{paths } P \text{ in } G' \text{ with } l(P) \leq (1 + \varrho) \cdot l^i : P \cap E' = \emptyset$  or equivalently,  $P$  in  $G$ . As  $k_i$  is cut, these paths contain at least one control edge and therefore  $k_i$  is covered.

If we could efficiently decide the cover problem, we could polynomially transform a given cut problem instance into an instance of the cover problem and hence decide the cut problem in polynomial time. This proves that the cover problem is NP-hard to decide at least for  $|\mathcal{K}| \geq 2$  and raises the question of how to compute a solution for the pCover.

### 2.2.2. Computing an Optimal Placement

With Proposition 2.1 we prevented a doubling in the number of edges and thus a dramatic increase in the number of routes. Our first approach is hence to compute all the  $R^i$ , i.e. all  $s_i - t_i$ -paths  $P$  with  $l(P) \leq (1 + \varrho) \cdot l^i$ , and solve the cover-IP. Determining  $R^i$  can be done with Yen's algorithm [Yen71] in a slightly modified version. Yen's algorithm computes the  $K$  shortest loopless  $s - t$ -paths in a digraph with non-negative edge costs. It finds the second shortest path by iteratively altering the shortest path and similarly the  $k^{\text{th}}$  shortest path by altering the  $k - 1^{\text{st}}$  shortest path. Our modified version of Yen's algorithm is described in Appendix A.2.

With a moderate choice of  $\varrho$  the number of routes  $|R^i|$  for  $k_i$  will be relatively small for most  $i$ . However, if the traffic network contains many cycles with short length,  $|R^i|$  may be huge for a few commodities  $k_i$ . In our case of the German motorway network the Ruhr Area particularly leads to an enormous number of paths for several commodities.

As we can see in Figure 2.1, the drivers have numerous possibilities to cross the Ruhr Area with short detours. Indeed, the application of the modified Yen's algorithm to the German motorway network leads to single commodities with thousands of possible routes, even if we set  $\varrho = 0.1$ , i.e. we allow detours of up to ten percent.

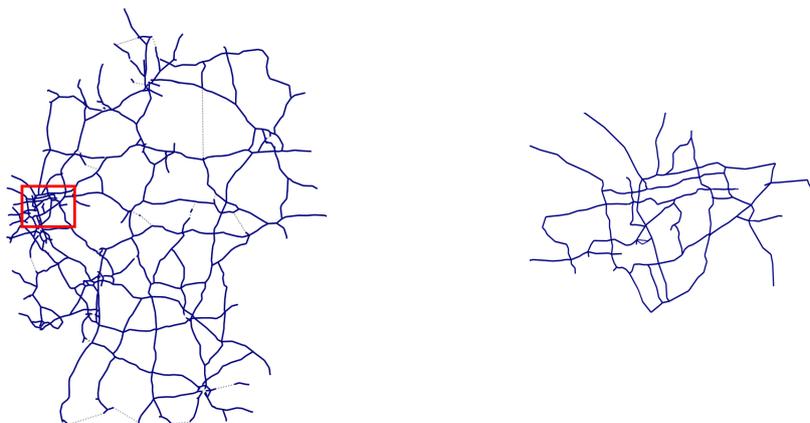


Figure 2.1.: The Ruhr Area (right) is a dense subnetwork of German motorways

Clearly, we need a different approach to solve the cover-IP (2.2) efficiently. Our solution is to handle the IP with row generation, i.e. to iteratively add constraints to the IP if necessary. The main idea is to consider a subset  $\overline{R}^i \subseteq R^i$  for all  $i$  and to solve the Restricted Master Problem  $\overline{P}$ .

$$\max_{y, \delta} \sum_{i=1}^{|\mathcal{K}|} x_i \cdot \delta_i \quad (2.4a)$$

$$\text{s.t.} \quad \sum_{e \in \mathcal{C}} y_e \leq m \quad (2.4b)$$

$$(\overline{P}) \quad \sum_{e \in r_j^i} l_e \cdot \left(1 + (u-1) \cdot y_e\right) \geq \delta_i \cdot (1 + \varrho + \varepsilon) \cdot l^i \quad \forall r_j^i \in \overline{R}^i, \forall i \quad (2.4c)$$

$$y_e \in \{0, 1\} \quad \forall e \in E \quad (2.4d)$$

$$\delta_i \in \{0, 1\} \quad \forall i \in \{1, \dots, |\mathcal{K}|\} \quad (2.4e)$$

Then we check in the so called pricing problem if the solution  $(y^*, \delta^*)$  of  $\overline{P}$  is also feasible for the cover-IP (2.2) or if there is a route  $r_j^i \in R^i \setminus \overline{R}^i$  such that

$$\sum_{e \in r_j^i} l_e \cdot \left(1 + (u-1) \cdot y_e^*\right) < \delta_i^* \cdot (1 + \varrho + \varepsilon) \cdot l^i. \quad (2.5)$$

If there is no such route, the solution of  $\overline{P}$  is optimal for (2.2), and if we find such a route  $r_j^i$ , we add it to  $\overline{R}^i$  and solve the Restricted Master Problem again.

**The Pricing Problem** Fortunately, the pricing problem turns out to be a shortest path problem. Note that (2.5) can only be true if  $\delta_i^* = 1$ , which means that  $k_i$  is thought to be covered. Now we define

$$l'_e := \begin{cases} l_e & \text{if } y_e^* = 0, \\ u \cdot l_e & \text{if } y_e^* = 1, \end{cases}$$

and compute a shortest path  $P^*$  in  $G$  w.r.t.  $l'$ . If

$$l'(P^*) := \sum_{e \in P^*} l'_e \leq (1 + \varrho) \cdot l^i < (1 + \varrho + \varepsilon) \cdot l^i,$$

we know that  $P^* \in R^i$ , since  $l(P^*) \leq l'(P^*)$ , and  $P^*$  violates (2.2c). Hence, we add  $P^*$  to  $\overline{R}^i$  and consider the next commodity.

If  $l'(P^*) > (1 + \varrho) \cdot l^i$ , we know that for every route  $r_j^i \in R^i$

$$\sum_{e \in r_j^i} l_e \cdot \left(1 + (u - 1) \cdot y_e^*\right) = l'(r_j^i) \geq l'(P^*) \geq (1 + \varrho + \varepsilon) \cdot l^i,$$

if  $\varepsilon$  is chosen small enough, and hence (2.2c) is satisfied in this case.

It is very useful that the pricing problem is in fact a shortest path problem. This allows for an efficient solution of the cover approach, if we do not have to solve  $\bar{P}$  too often. As mentioned above, we will apply this algorithm in Chapter 4 when we optimize the placement of the control gantries for German motorways.

We finish this section with an important note. The cover approach can also be used to decide where to install new, additional control gantries. Suppose therefore, that we already have a set of control edges  $\mathcal{C}_0 \subseteq E$  with  $|\mathcal{C}_0| = m_0$  and that we want to add  $m_1$  traffic control gantries. The cover-IP can easily be adapted by replacing  $m$  in (2.2b) by  $m_0 + m_1$  and by adding the constraints  $y_e = 1 \forall e \in \mathcal{C}_0$ .

## 2.3. Placement with an Indefinite Number of Control Gantries

In many cases it is not clear how many control gantries are appropriate, especially if we are planning to establish the automatic toll control. Of course one could look at specific traffic networks such as the German motorway network and compute the number of control gantries in relation to the total number of sections. However, the number of necessary control gantries crucially depends on the topology of the network and hence should be determined separately. In this chapter we will therefore use two strategies to determine this number. While at first we ask for the number of gantries necessary to control a certain amount of traffic, our second approach aims at placing control gantries if they are profitable.

Determining this number might be one of the most important issues for the implementation of an automatic toll control system. Both our models are interesting and useful but only simple developments of the cover approach. Therefore, we will present them just briefly to illustrate the respective approaches.

### 2.3.1. The $d\%$ -Cover

The central question of this approach is: How many traffic control gantries do we need in order to cover  $d\%$  of the traffic? A greedy approach where we successively cover the commodities with the highest demand  $x_i$  may lead to arbitrarily bad results. Fortunately, a little modification of the cover-IP (2.2) solves the problem.

$$\min_{y, \delta, m} m \quad (2.6a)$$

$$\text{s.t.} \quad \sum_{e \in \mathcal{C}} y_e \leq m \quad (2.6b)$$

$$\sum_{e \in r_j^i} l_e \cdot (1 + (u-1) \cdot y_e) \geq \delta_i \cdot (1 + \varrho + \varepsilon) \cdot l^i \quad \forall r_j^i \in R^i, \forall i \quad (2.6c)$$

$$\sum_i x_i \cdot (\delta_i - d\%) \geq 0 \quad (2.6d)$$

$$y_e \in \{0, 1\} \quad \forall e \in E \quad (2.6e)$$

$$\delta_i \in \{0, 1\} \quad \forall i \in \{1, \dots, |\mathcal{K}|\} \quad (2.6f)$$

$$m \in \mathbb{N} \quad (2.6g)$$

By declaring the number  $m$  of control gantries as a variable that we want to minimize and by adding constraint (2.6d), we modified the cover-IP to perfectly formulate our problem. Constraint (2.6d) ensures that we cover at least  $d\%$  of the total traffic, since

$$\sum_i x_i \cdot (\delta_i - d\%) \geq 0 \iff \sum_i x_i \cdot \delta_i \geq d\% \cdot \sum_i x_i.$$

In order to determine the “right” number of control gantries, we have to choose  $d$  though. There is an obvious relation between  $d$  and the objective value  $m^*(d)$  of (2.6), namely that  $m^*$  is a monotonically increasing function of  $d$ . In other words, the more traffic we want to cover, the more control gantries we need. A quantitative statement, however, is impossible since  $m^*$  heavily depends on the network, particularly on the commodities. In conclusion we can say that the choice of  $d$  for a specific network is a political decision. After computing various combinations  $(d_i, m^*(d_i))_{i \in I}$ , one has to decide which combination represents a good trade-off between a broad control and the installation costs.

The installation costs were considered implicitly in the  $d\%$ -cover approach as we tried to minimize the number of control gantries. The next model, however, includes the installation costs for a traffic control gantry explicitly.

### 2.3.2. Placement of Profitable Control Gantries

Building a control gantry is expensive and hence our goal is to build as few as possible. For every possible control gantry we should therefore weigh the costs and benefits of it. It is, however, unclear how to express the benefit of a control gantry in terms of money. One way could be to value the control of an additional driver with some  $v > 0$ . If we assume that the installation costs for a control gantry are a constant  $c$  (independent of the location), then we can say that an additional gantry is worth to be installed, if at least  $c/v$  additional drivers are covered with it. As we can see, we use the cover approach once again and hence modify the cover-IP (2.2) to determine the appropriate number of control gantries for this model.

$$\max_{y, \delta, m} \left( v \cdot \sum_{i=1}^{|\mathcal{K}|} x_i \cdot \delta_i \right) - c \cdot m \quad (2.7a)$$

$$\text{s.t.} \quad \sum_{e \in \mathcal{C}} y_e \leq m \quad (2.7b)$$

$$\sum_{e \in r_j^i} l_e \cdot (1 + (u-1) \cdot y_e) \geq \delta_i \cdot (1 + \varrho + \varepsilon) \cdot l^i \quad \forall r_j^i \in R^i, \forall i \quad (2.7c)$$

$$y_e \in \{0, 1\} \quad \forall e \in E \quad (2.7d)$$

$$\delta_i \in \{0, 1\} \quad \forall i \in \{1, \dots, |\mathcal{K}|\} \quad (2.7e)$$

$$m \in \mathbb{N} \quad (2.7f)$$

The constraints are known from the cover-IP with the little modification that  $m$  is a variable now. The objective function (2.7a) ensures that we maximize the amount of covered traffic with control gantries that are worth to be installed, as the following proposition shows.

**Proposition 2.3** *In every optimal solution of (2.7) we have  $y_e = 1$  only if a control gantry on  $e$  is worth to be installed in the above sense.*

*Proof.* Suppose that a control gantry on  $e$  covers  $x < \frac{c}{v}$  additional drivers and suppose that  $y_e = 1$  in an optimal solution of (2.7). Setting  $y_e = 0$  would save installation costs of  $c$  but lead to less revenue of  $-x \cdot v$ . Hence, the objective value would increase by  $-x \cdot v + c > -c + c = 0$ , which contradicts the optimality.  $\square$

The main problem of this approach is still to evaluate the benefit of a control gantry or to set  $v$ . The idea, however, to consider the number of control gantries as a variable, is of crucial importance for practical use. Both our models provide understandable approaches for this problem and can be a valuable help when placing traffic control gantries. As pointed out above, the good placement of the control gantries is vital for an effective automatic toll control. For this control we need a control strategy, when confronted with the additional constraint that we may only use a certain part of the control gantries at the same time. The placement of the gantries, however, should be independent from finding this optimal control strategy. As described before, it should enable a broad control which is costly to avoid. With such a fundamental infrastructure, we can consider different approaches for optimizing the control strategy. This will indeed be the subject of the following chapter where we extensively study two models to optimize the toll control.

### 3. The Optimal Control Strategy

Concerning the German motorway network, we are confronted with an existing infrastructure of about 300 traffic control gantries. Due to data capacity constraints and data protection, it is not possible to activate all control gantries at the same time. For this reason we assume in this chapter that  $m$  control gantries are already placed and consider the following task:

Find an optimal control strategy, which only uses the capacity of  $k$  control gantries!

The main idea hereby is to split the control capacity, meaning that a control gantry is active only a certain amount of time. The model that we develop should represent a game theoretic approach, since the drivers will clearly adapt to our control strategy. For instance, they would pay the toll instead of evading if the toll expenses are less than the expected penalty for evading. The basis for the game is a traffic network similar to the one in the previous chapter, and a set  $\mathcal{C}$  of control edges is given. The drivers of a commodity may choose between several paths but again, they will not take an arbitrary path. Our goal is now to distribute the total control capacity among all control edges “in a good way”. Even with this vague objective this problem is very specific and therefore, it is difficult to find a fitting approach in the literature. Indeed, the literature search was without success as all approaches from literature lack at least one important aspect.

However, we can use the cover approach from the previous chapter and extend it to our benefit. By relaxing the integer constraints of the  $y_e$  and  $\delta_i$  in the cover-IP, we get a linear program (LP) which represents a probabilistic cover. In this *pCover* every commodity is covered with a certain probability and the drivers have to pay a fine if they are caught evading. They now compare the expected penalty for evading on the best route to the toll costs for the respective commodity, and play accordingly.

The basics of our game theoretic model are described in the first section. After a slight modification of the traffic network we will specify the players, strategies and payoffs for our game.

In the second section we will present the pCover. This is a robust approach with the goal to maximize the total revenues for the controlling inspectors, consisting of toll revenues and penalties. As the number of constraints is immense in our LP-formulation of the pCover, we apply a cutting-plane method to solve it.

For the pCover we assumed that the drivers are perfectly aware of the control strategy. However, in real life one cannot tell whether a certain control gantry is activated or not. Hence, we take the flawed perception of the drivers into account and present the  $\alpha$ -pCover, an extension of the pCover, in the third section.

Finally, in the last section we will compare these two approaches and examine a gap in the respective objective values.

### 3.1. The Game Theoretic Model

We start with the network stated in the previous chapter, consisting of a digraph  $G = (V, E)$  with edge lengths  $l_e \in \mathbb{Q}_{>0}$  and commodities  $\mathcal{K}$ . Similar to Chapter 2, we add a detour edge  $e'$  parallel to every control edge  $e \in \mathcal{C}$  and set  $l_{e'} := u \cdot l_e$ , allowing the drivers to avoid  $e$  by making a detour.

For the game theoretic model we assume that all drivers are rational and intelligent, i.e. they want to maximize their expected payoff and thus play accordingly [Mye91]. This assumption implies that there are no honest drivers who always pay the toll, which is obviously wrong. However, the assumption becomes legitimate, if the fraction of honest drivers is the same on every route. In this case we could consider the remaining fraction of crafty drivers and obtain the same results [BOSS12].

For our purpose we consider a *Stackelberg game*, i.e. a bilevel game where the players are divided into leaders and followers. First, each leader commits to a strategy, then the followers choose a strategy after observing the leaders' strategies. Stackelberg games were introduced by Heinrich von Stackelberg in 1934 [vS34] and are particularly suitable for handling security problems. They have been used in [JTP<sup>+</sup>10] for airport security in Los Angeles, in [YJJ<sup>+</sup>12] for fare inspection at LA Metro and in [BOSS12] for mobile toll control on German motorways. All these applications differ from our problem in significant issues. A main difference is that we may control only at certain points of the network. In contrast, fare inspectors of LA Metro and mobile toll control units may control network-wide.

In the following, we will specify the players and strategies of our game theoretic approach.

**Players** The inspectors are aggregated as one player  $insp$ , the leader. They will initially commit to a control strategy, which the drivers should learn and adapt to over time. Hence, for every commodity  $k_i \in \mathcal{K}$ , the drivers of  $k_i$  are represented by one player  $P_i$  and  $P_i$  is considered as a follower. The set of players can then be written as  $(insp, P_1, \dots, P_{|\mathcal{K}|})$ .

**Strategies** Let us first consider the set of strategies  $\Sigma_{insp}$  for the inspectors. The total control capacity of  $k$  control gantries can be assigned to the set  $\mathcal{C}$  of control edges. Thus,

$$\Sigma_{insp} = \left\{ (q_e) \in [0, 1]^{\mathcal{C}} : \sum_{e \in \mathcal{C}} q_e = k \right\}.$$

$0 \leq q_e \leq 1$  means that the control gantry at  $e \in \mathcal{C}$  is active with probability  $q_e$ . Note that we could also have used pure strategies  $(0, \dots, 0, k, 0, \dots, 0)$ . Then  $\Sigma_{insp}$  is the set of mixed strategies and we have to add the additional constraint  $q_e \leq 1 \forall e \in \mathcal{C}$  since we cannot control more than 100% on a control edge.

Let us now consider the drivers' strategies. Player  $P_i$  of commodity  $k_i$  can decide between paying the toll and evading. If he decides to pay, we assume that he takes a shortest  $s_i - t_i$ -path in  $G$  and pays a toll  $t_i = l^i \cdot t$  proportional to the path length  $l^i$ . If he decides to evade, he may choose an  $s_i - t_i$ -path in addition, and has to pay a penalty  $p \gg t_i$  if he gets caught. Since the drivers' wages as well as the fuel consumption are not uniform, we would like not to take any travelling costs into account. Instead, we will adapt the approach from Chapter 2 and forbid  $P_i$  to make a long detour. For a fixed  $\varrho > 0$  let

$$R^i = \{s_i - t_i - \text{paths } P \text{ in } G : l(P) \leq (1 + \varrho) \cdot l^i\}.$$

Note that a route in  $R^i$  can now contain detour edges which means that every driver on this route will avoid the associated control edges. Therefore, we denote a route in  $R^i$  by  $r_{j,B}^i$ , where  $r_j^i$  is the underlying main route<sup>1</sup> and  $B = \mathcal{C} \cap r_{j,B}^i$  is the set of control edges that are used in  $r_{j,B}^i$ .

An evading player  $P_i$  may choose a route from  $R^i$  and hence, the set of strategies  $\Sigma_i$  for  $P_i$  is given by

$$\Sigma_i = \{\text{pay toll}\} \cup (\{\text{evade}\} \times R^i).$$

<sup>1</sup>If we replace every detour edge  $e'$  on  $r_{j,B}^i$  by its associated control edge  $e$  we get  $r_j^i$ .

**Payoffs** The inspectors' payoff  $\pi_{insp}$  is given by the total revenue achieved by toll revenues and penalties. Since we do not consider any travelling costs, the reward for the inspectors equals the total expenses of the drivers and thus, our game is zero-sum.

The choice not to consider travelling costs for  $P_i$  but to focus on  $R^i$  represents no restriction if we choose  $\varrho$  big enough. In this case every route that would be optimal for  $P_i$  if we considered travelling costs, will also be in  $R^i$ , since a route  $r \notin R^i$  would lead to exorbitantly high travelling costs. However, we would lose the zero-sum property of the game, if we included travelling costs.

Let  $\pi_{insp,i}(q, \sigma_i)$  be the payoff the inspectors get from player  $P_i$  if he plays according to  $\sigma_i \in \Sigma_i$  while the inspectors' control strategy is  $q$ . Then

$$\pi_{insp}(q, \sigma_1, \dots, \sigma_{|\mathcal{K}|}) = \sum_{i=1}^{|\mathcal{K}|} \pi_{insp,i}(q, \sigma_i).$$

Similarly,  $\pi_i(q, \sigma_i)$  denotes the payoff for a driver of commodity  $k_i$ . Note that it does not depend on the behavior of the drivers of other commodities. With the assumption of the unbounded control capacity of the traffic control gantries we do not have to consider the amount of traffic on a control edge per time unit and can handle the commodities individually. While

$$\pi_{insp,i}(q, \text{pay toll}) = t_i = -\pi_i(q, \text{pay toll})$$

is obvious, the evading strategy needs some explanation. If a control edge is activated, every evader on it will get caught. Again, this is due to the unbounded control capacity. If it is activated with probability  $q_e$ , every evader on it will get caught with that probability. If player  $P_i$  plays according to  $(\text{evade}, r_{j,B}^i)$ , he might get caught exactly at the control edges  $e \in B$ . The probability not to be controlled on  $e \in B$  is  $1 - q_e$  and as these events are independent, the probability for an evader not to get caught on  $r_{j,B}^i$  is  $\prod_{e \in B} (1 - q_e)$ . For small  $q_e$  it is reasonable to approximate the probability to get caught with

$$1 - \prod_{e \in B} (1 - q_e) \approx \sum_{e \in B} q_e.$$

The right hand side equals the expected number of controls for a driver on  $r_{j,B}^i$ . The approximation becomes perfectly legitimate if the drivers had to pay the penalty

$p$  twice if they got caught twice. For the sake of simplicity we will assume this throughout this thesis and work with the expected penalty for  $P_i$  as follows.

If  $P_i$  evades and takes route  $r_{j,B}^i$  the expected penalty for him is

$$-\pi_i(q, (\text{evade}, r_{j,B}^i)) = p \cdot \sum_{e \in B} q_e,$$

if the inspectors' control strategy is  $q \in \Sigma_{insp}$ .

## 3.2. A Probabilistic Cover

The aim of the cover problem from Chapter 2 was to ensure the control of the traffic on certain commodities. However, it could be sufficient to control the traffic with a certain probability. In this section we will closely examine this probabilistic cover approach. At first we will introduce the pCover and study the equilibria of this game. Later on, we focus on the computation of an optimal control strategy with cutting planes.

### 3.2.1. The pCover Approach

As mentioned earlier, the pCover is an extension of the cover problem and a robust approach to optimize the automatic toll control. In our game theoretic setting, the inspectors commit to a control strategy which is perfectly observed by the drivers. Based on their observations, the drivers decide either to pay the toll or to evade on a chosen route. Our goal is to maximize the total revenue for the inspectors consisting of toll revenues and penalty payments.

The robustness is due to the drivers' knowledge about the control strategy and the zero-sum property of our game. Every player  $P_i$  chooses the worst-case route for us if he decides to evade, or he pays the toll if this is cheaper. Thus, the total revenue computed with the pCover approach is a lower bound on the actual revenue with the same control strategy when the drivers might choose more advantageously for us.

The next issue, how the drivers will respond to the control strategy of the inspectors, is particularly important for an equilibrium. As we consider a Stackelberg game, we are interested in a Stackelberg equilibrium, which we will define in a moment.

Given the inspectors' strategy  $q$ , a best response for  $P_i$  is given by the *best response function*  $r_i$ .

**Definition 3.1**  $r_i : \Sigma_{insp} \rightarrow \Sigma_i$  is a best response function for player  $P_i$  if

- i)  $\forall q \in \Sigma_{insp} : \pi_i(q, r_i(q)) \geq \pi_i(q, \sigma_i) \quad \forall \sigma_i \in \Sigma_i,$
- ii)  $\pi_i(q, r_i(q)) = \pi_i(q, \sigma_i) \implies \pi_{insp}(q, r_i(q)) \geq \pi_{insp}(q, \sigma_i).$

The first condition assures that there is no strategy for  $P_i$  better than  $r_i(q)$  if the inspectors' strategy is  $q$ . Equivalently, we have  $r_i(q) \in BR_i(q)$  where  $BR_i(q)$  is the set of best responses for  $P_i$  to the control strategy  $q$ . The second condition states that the followers will break a tie in favor of the leader. This means that if they are indifferent between two options, they will choose the one which is better for the leader. With this assumption the existence of a Stackelberg equilibrium is guaranteed [CS06]. However, ii) is particularly important in non-zero-sum games, since in our zero-sum game  $\pi_i(q, r_i(q)) = \pi_i(q, \sigma_i)$  implies

$$\pi_{insp}(q, r_i(q)) = -\pi_i(q, r_i(q)) = -\pi_i(q, \sigma_i) = \pi_{insp}(q, \sigma_i).$$

We will abuse notation and denote by  $r_i(q)$  not only a best strategy for  $P_i$ , but also an optimal route, if  $r_i(q) \neq \text{pay toll}$ . With our best response function we are now prepared for defining a Stackelberg equilibrium.

**Definition 3.2** A strategy profile  $(q, \sigma_1, \dots, \sigma_{|\mathcal{K}|}) \in \Sigma_{insp} \times \Sigma_1 \times \dots \times \Sigma_{|\mathcal{K}|}$  is a Stackelberg equilibrium if

- i) for all best response functions  $r_1, \dots, r_{|\mathcal{K}|} :$   
 $\pi_{insp}(q, r_1(q), \dots, r_{|\mathcal{K}|}(q)) \geq \pi_{insp}(q', r_1(q'), \dots, r_{|\mathcal{K}|}(q')) \quad \forall q' \in \Sigma_{insp},$
- ii) for all  $i \in \{1, \dots, \mathcal{K}\} : \quad \sigma_i \in BR_i(q).$

While i) assures that there is no better control strategy for the inspectors, ii) ensures that every player responds optimally to the control strategy. In both definitions we considered only pure strategies for the players. The following proposition shows that we may restrict ourselves to these.

**Proposition 3.3** Let  $s$  be a mixed strategy for  $P_i$  and let  $\pi_i(q, s)$  be his expected payoff with  $q \in \Sigma_{insp}$  fixed.

Then there is a pure strategy  $\tau \in \Sigma_i$  such that  $\pi_i(q, \tau) \geq \pi_i(q, s).$

*Proof.* Let  $s_{\sigma_i}$  denote the probability that  $P_i$  plays  $\sigma_i \in \Sigma_i$  (according to  $s$ ) and consider  $\tau = r_i(q)$ . Then  $\tau \in \Sigma_i$  and

$$\pi_i(q, s) := \sum_{\sigma_i \in \Sigma_i} s_{\sigma_i} \cdot \pi_i(q, \sigma_i) \stackrel{3.1 \text{ i)}}{\leq} \sum_{\sigma_i \in \Sigma_i} s_{\sigma_i} \cdot \pi_i(q, r_i(q)) = \pi_i(q, \tau) \cdot \sum_{\sigma_i \in \Sigma_i} s_{\sigma_i} = \pi_i(q, \tau).$$

□

With this proposition we can assume that all drivers of  $k_i$  take the same route and still guarantee the existence of a Stackelberg equilibrium in the pCover.

Similar to the Nash equilibrium in a zero-sum game, we compute a Stackelberg equilibrium for the pCover with a linear program (LP). We are particularly interested in an optimal strategy for the leader and receive it by solving the following LP.

$$\max_{q, \lambda} \sum_{i=1}^{|\mathcal{K}|} x_i \cdot \lambda_i \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{e \in \mathcal{C}} q_e \leq k \tag{3.1b}$$

$$\lambda_i \leq p \cdot \sum_{e \in B} q_e \quad \forall r_{j,B}^i \in R^i, \quad \forall i \tag{3.1c}$$

$$\lambda_i \leq t_i \quad \forall i \tag{3.1d}$$

$$q_e \geq 0 \quad \forall e \in \mathcal{C} \tag{3.1e}$$

$$\lambda_i \geq 0 \quad \forall i \in \{1, \dots, |\mathcal{K}|\} \tag{3.1f}$$

$\lambda_i$  are the expenses for every driver of commodity  $k_i$  and thus the inspectors want to maximize  $\sum_i x_i \cdot \lambda_i$ , the total revenue. (3.1b) ensures that we do not exceed the control capacity  $k$ , while (3.1c) and (3.1d) assure that the truck drivers take the worst-case strategy for the inspectors. In (3.1e) we relaxed the integrality constraints and replaced  $y_e \in \{0, 1\}$  with  $q_e \geq 0$ . Furthermore, the following holds:

**Lemma 3.4** *There is always an optimal solution of (3.1) where  $q \in \Sigma_{insp}$ .*

*Proof.* Recall  $\Sigma_{insp} = \{(q_e) \in [0, 1]^{\mathcal{C}} : \sum_{e \in \mathcal{C}} q_e = k\}$  and let  $(q, \lambda)$  be an optimal solution for the pCover. If  $\sum_{e \in \mathcal{C}} q_e < k$ , we can increase one  $q_e < 1$  by the difference  $k - \sum q_e$  and the objective value will not decrease. Now suppose  $q_{\tilde{e}} > 1$  for some  $\tilde{e} \in \mathcal{C}$ . If we set  $q_{\tilde{e}} = 1$ , the solution is still feasible and the objective value remains

the same, since

$$\lambda_i \stackrel{(3.1d)}{\leq} t_i < p \cdot q_{\tilde{e}} \leq p \cdot \sum_{e \in B} q_e \quad \forall r_{j,B}^i \in R^i \text{ with } \tilde{e} \in r_{j,B}^i.$$

By successively applying these two procedures we get an optimal solution with a feasible control strategy.  $\square$

As seen in the proof,  $q_e > 1$  is a waste of control capacity, as well as  $\sum_{e \in \mathcal{C}} q_e < k$ . Therefore, if  $k$  is small enough<sup>2</sup>,  $q \in \Sigma_{insp}$  for all optimal solutions of the pCover.

### 3.2.2. Computation of the Inspectors' Strategy

We have seen that the optimal control strategy can be computed with an LP. However, the transformation of the pCover problem into the LP is not in polynomial time. As in the previous chapter,  $R^i$  may contain exponentially many  $s_i - t_i$ -paths that we have to consider in (3.1c). For this reason, we adapt the solution concept from there, and use row generation to determine the optimal control strategy.

In this cutting-plane approach we consider a subset  $\overline{R}^i \subseteq R^i$  for all  $i$  and find an optimal control strategy  $q^*$  if  $P_i$  has to choose a route in  $\overline{R}^i$ . Again, this is done in the Restricted Master Problem  $\overline{P}$ :

$$\max_{q, \lambda} \sum_{i=1}^{|\mathcal{K}|} x_i \cdot \lambda_i \tag{3.2a}$$

$$\text{s.t.} \quad \sum_{e \in \mathcal{C}} q_e \leq k \tag{3.2b}$$

$$\lambda_i \leq p \cdot \sum_{e \in B} q_e \quad \forall r_{j,B}^i \in \overline{R}^i, \quad \forall i \tag{3.2c}$$

$$\lambda_i \leq t_i \quad \forall i \tag{3.2d}$$

$$q_e \geq 0 \quad \forall e \in \mathcal{C} \tag{3.2e}$$

$$\lambda_i \geq 0 \quad \forall i \in \{1, \dots, |\mathcal{K}|\} \tag{3.2f}$$

Afterwards, we solve the pricing problem where we check if there is a better strategy for  $P_i$  in  $R^i \setminus \overline{R}^i$  and add this route to  $\overline{R}^i$ . More specifically, we compute a best response  $r_i(q^*)$  for all  $P_i$  and check whether  $r_i(q^*) \in \overline{R}^i$ . If  $r_i(q^*) \in \overline{R}^i \forall i$ , the computed control strategy is optimal.

<sup>2</sup>In practice,  $k$  is always “small enough”.

The cutting-plane algorithm can then be stated as follows.

---

**Algorithm 1:** cutting-plane algorithm

---

```

1 Initialize  $\overline{R}^i$ ;
2 repeat
3   Solve  $\overline{P}$ ;
4   for  $i \in \{1, \dots, |\mathcal{K}|\}$  do
5     Determine best response  $r_i(q)$  for  $P_i$ ;
6     if  $r_i(q) \notin \overline{R}^i$  and  $r_i(q) \neq \{\text{pay toll}\}$  then
7       Add  $r_i(q)$  to  $\overline{R}^i$ ;
8     end
9   end
10 until no path has been added to an  $\overline{R}^i$ ;

```

---

For our purposes it suffices to initialize  $\overline{R}^i$  as a shortest  $s_i - t_i$ -path in  $G$ . Since  $\overline{P}$  has a relatively small number of constraints, the computation of the solution is fast. However, the pricing problem turns out to be much harder.

**The Pricing Problem** After solving  $\overline{P}$  to optimality, we want to find a cheapest evading route for the drivers and check if they would have paid less than  $\lambda_i$ . Therefore, we extend the optimal control strategy to  $(q_e)_{e \in E}$  by setting  $q_e = 0 \quad \forall e \notin \mathcal{C}$  and compute a shortest  $s_i - t_i$ -path  $r^* \in R^i$  w.r.t.  $q$ . A shortest  $s_i - t_i$ -path w.r.t.  $q$  is a path in  $G$  where the drivers have the lowest probability to get controlled among all  $s_i - t_i$ -paths. The constraint  $r^* \in R^i$  ensures that we consider only feasible evading routes.

Unfortunately, with this additional constraint the pricing problem becomes a *constrained shortest path problem* (CSP) which is difficult to handle. For our purpose we consider the CSP where only one additional resource is considered and formulate the problem similar to [DB01]. Here, we are given a digraph  $G = (V, E)$  where each edge  $e \in E$  has a weight  $w_e \geq 0$  and a cost  $c_e \geq 0$ . Furthermore, we consider  $s, t \in V$  and a number  $W$  and ask for an  $s - t$ -path  $P$  in  $G$  with  $\sum_{e \in P} w_e \leq W$  that minimizes  $\sum_{e \in P} c_e$ . The CSP is closely related to the Knapsack problem and hence the corresponding decision problem is NP-complete [HZ80].

Obviously, in our case we have  $w_e = l_e$  and  $c_e = q_e$  while  $W = (1 + \varrho) l^i$  depends on the considered commodity. We want to take advantage of the specialty of our problem where we have to solve a CSP for every commodity and compute multiple constrained shortest paths at the same time. Therefore, we will regard the CSP as a

bicriteria optimization problem and introduce the most important definition in this field.

**Definition 3.5** *Let  $P_0, P_1$  be paths in  $G$  and let  $l(P)$  denote the length of a path  $P$  and  $pr(P)$  denote the probability for a control on  $P$ . We write  $P_0 \succeq P_1$  if*

$$l(P_0) \leq l(P_1) \text{ and } pr(P_0) \leq pr(P_1).$$

*Furthermore, we say that  $P_0$  dominates  $P_1$  ( $P_0 \succ P_1$ ) if at least one of the inequalities is strict.*

*Finally, an  $s - v$ -path  $P$  is called pareto-optimal if there is no  $s - v$ -path  $P'$  such that  $P' \succ P$ .*

Our algorithm for solving the CSP can be described as a pareto-optimal version of the Bellman-Ford algorithm [Bel58], [For56]. As we have a huge number of commodities and want to find a constrained shortest path for every  $k_i$ , we develop an algorithm that considers multiple commodities at the same time. More specifically, we solve the problem simultaneously for all commodities with the same start node  $s$ . Therefore, we explore the network starting at  $s$ , and store in  $reach[v]$  the pareto-optimal  $s - v$ -paths. As stated in the definition above, an  $s - v$ -path  $P$  is pareto-optimal if no other  $s - v$ -path is shorter than  $P$  and has a lower probability for a control.

In our single-source multiple-sink constrained shortest path algorithm the nodes are labeled and scanned. In each iteration every labeled node is considered once and scanned afterwards. When a node  $u$  is considered, we check for every pareto-optimal  $s - u$ -path that we have detected so far, if an outgoing arc  $(u, v)$  will lead to a new pareto-optimal  $s - v$ -path. If this is the case, we store the new path in  $reach[v]$  and label  $v$ . If  $v$  is already scanned, i.e.  $v$  was already considered in this iteration, we add  $v$  to the set  $next$  instead of labeling it and consider it in the next iteration.

---

**Algorithm 2:** single-source multiple-sink constrained shortest path algorithm
 

---

**Input:**  $(G, l_e > 0, q_e \geq 0), s \in V, T \subseteq V, \varrho > 0$ 
**Output:** Shortest  $s - t$ -paths  $P_t$  for  $t \in T$  w.r.t.  $q$  and  $l(P_t) \leq (1 + \varrho) \cdot l^{s,t}$  <sup>3</sup>

```

1 reach[v] := ∅  ∀v ;
2 reach[s] := {s};
3 Unlabel all nodes;
4 next = {s};
5 while next ≠ ∅ do
6   Label all nodes in next and set next := ∅;
7   Unscan all nodes;
8   while ∃ labeled node do
9     Choose labeled node u and scan u;
10    for s - u-paths P ∈ reach[u] do
11      for successors v of u do
12        if P ∪ v is not dominated by any s - v-path in reach[v] then
13          Add P ∪ v to reach[v];
14          Label v or add v to next if v is scanned;
15        end
16        Remove all P' from reach[v] with (P ∪ v) ≻ P';
17      end
18    end
19  end
20 end
21 for t ∈ T do
22   Sort reach[t] w.r.t. the pathlengths l(P) in decreasing order;
23   Find the first path P in reach[t] respecting the length constraint;
24   P_t := P;
25 end
26 return (P_1, …, P_{|T|}) ;

```

---

<sup>3</sup> $l^{s,t}$  is the length of a shortest  $s - t$ -path.

The following lemma is the key to the correctness proof for the multiple-sink constrained shortest path algorithm.

**Lemma 3.6** *Let  $P_v$  be a pareto-optimal  $s - v$ -path with  $k$  edges. Then, after  $k$  iterations<sup>4</sup>,  $reach[v]$  contains a path  $P_v^*$  with  $P_v^* \succeq P_v$ .*

*Proof.* We proof the statement by induction. For  $k = 1$  it is obviously true. In the first iteration we consider every neighbor  $v$  of  $s$  and store  $\{s, v\}$  as pareto-optimal path in  $reach[v]$ .

Now suppose that the statement is correct for some  $k$  and consider a pareto-optimal path  $P_{k+1} = s, v_1, \dots, v_k, v_{k+1}$  with  $k + 1$  edges. Obviously,  $P_k = s, v_1, \dots, v_k$  is a pareto-optimal  $s - v_k$ -path, since otherwise, if some  $P$  would dominate  $P_k$ , then  $P \cup v_{k+1} \succ P_{k+1}$ . Furthermore, by induction, after  $k$  iterations  $reach[v_k]$  contains a path  $P'_k$  with  $P'_k \succeq P_k$ . Now we have two possibilities:

- i) After the  $k^{th}$  iteration  $reach[v_{k+1}]$  already contains a path  $P_0$  s.t.  $P_0 \succeq P_{k+1}$ . Then, in the  $k + 1^{st}$  iteration, this path can only be replaced by a dominating path.
- ii) In the other case,  $P_k$  was added to  $reach[v_k]$  in the  $k^{th}$  iteration and  $v_k$  was not considered since. Hence,  $v_k$  is in *next* after the  $k^{th}$  iteration and in the next iteration we consider  $P_{k+1}$  and add it to  $reach[v_{k+1}]$  if not  $P'_{k+1} \succeq P_{k+1}$  for some  $P'_{k+1}$  in  $reach[v_{k+1}]$ .

□

This result is similar to the well-known and fundamental lemma for the correctness proof of the Bellman-Ford algorithm. In accordance with that, we are now prepared for the following theorem.

**Theorem 3.7** *Algorithm 2 is correct.*

*Proof.* Since  $l_e > 0, q_e \geq 0 \quad \forall e \in E$ , we know that  $G$  contains no negative cycles and even no cycles of length  $(0, 0)$ . Therefore, every pareto-optimal  $s - v$ -path has at most  $n - 1$  edges. According to Lemma 3.6 we need at most  $n - 1$  iterations in Algorithm 2 to find all pareto-optimal paths in  $G$ . As every node is labeled at most once per iteration, the algorithm terminates.

---

<sup>4</sup>The number of iterations is w.r.t. line 5 of the algorithm.

Now we have to prove that the algorithm does not terminate before every pareto-optimal path has been found. Suppose that Algorithm 2 terminates but there is a pareto-optimal path  $P_v = s, v_1, \dots, v_k, v$  such that  $P_v \succ P \forall P \in reach[v]$  at the end. Let  $j$  be maximal such that  $P_0 \succeq P_{v_j} := s, v_1, \dots, v_j$  for some  $P_0 \in reach[v_j]$ , i.e.  $s, v_1, \dots, v_j, v_{j+1} \succ P' \forall P' \in reach[v_{j+1}]$ <sup>5</sup>, and suppose that  $P_0$  was added to  $reach[v_j]$  in the  $i^{th}$  iteration.

If  $v_j$  would have been labeled afterwards, the algorithm would have considered  $v_j$  and the outgoing edge  $(v_j, v_{j+1})$  later on in the same iteration. Then it would have detected  $P_0 \cup v_{j+1}$  as a pareto-optimal  $s - v_{j+1}$ -path and stored the new path in  $reach[v_{j+1}]$ , which is a contradiction to the maximality of  $j$ .

Otherwise,  $v_j$  would have been added to  $next$  and  $P_0 \cup v_{j+1}$  would have been detected in the next iteration, which leads to the same contradiction.  $\square$

Now that we have completed the correctness proof for the single-source multiple-sink constrained shortest path algorithm, we have to worry about the runtime. As the CSP is NP-hard to decide, there is likely no algorithm to find a constrained shortest path in polynomial time (unless  $P=NP$ ). In fact, our algorithm not only finds the shortest path which respects the length constraint, but computes all pareto-optimal paths from one node  $s$ . There can be exponentially<sup>6</sup> many pareto-optimal paths, even if we consider only one commodity.

**Example** Consider the following digraph  $G$ :

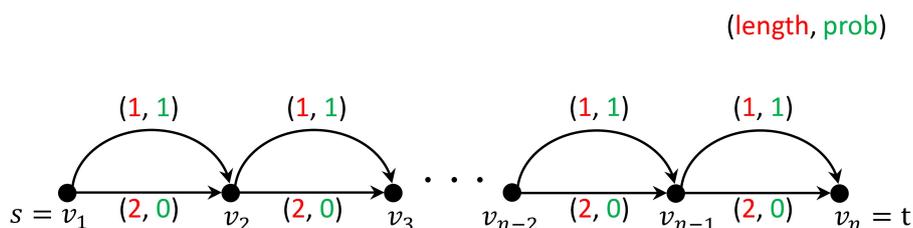


Figure 3.1.: Digraph with exponentially many pareto-optimal  $s - t$ -paths

Note that every  $s - t$ -path in  $G$  is pareto-optimal and hence there are  $2^{n-1}$  pareto-optimal  $s - t$ -paths. If we apply Algorithm 2 to it, we label every node only once, but when we consider  $v_i$ ,  $reach[v_i]$  contains  $2^{i-1}$  paths and we expand each of those paths in two ways. Therefore, we consider each edge  $(v_k, v_{k+1})$   $2^k$  times and hence have a total runtime of  $\mathcal{O}(2^n)$ .

<sup>5</sup>For the case  $j = k$  we set  $v_{k+1} := v$ .

<sup>6</sup>This means exponential in the number of nodes or edges.

We conclude the algorithmic part with a few remarks on possible improvements of the algorithms stated above.

- i) In Algorithm 2 we can compute  $lmax = (1 + \varrho) \cdot \max_{t \in T} l^{s,t}$  at the beginning. Note that every path  $P$  with length  $l(P) > lmax$  does not meet the length constraint. Thus, we check after line 11 if  $l(P \cup v) > lmax$  and consider  $(P \cup v)$  only if this is false. The correctness of the algorithm is not affected thereby. We might not find every pareto-optimal path, but we will still find every pareto-optimal path that fullfills the length constraint.
- ii) Since we compute all pareto-optimal paths for  $k_i$  we can as well add multiple paths to  $\overline{R}^i$  in line 7 of the cutting-plane algorithm. If we do, we potentially have to solve our pricing problem less often.
- iii) In a preprocessing of Algorithm 1 we delete those commodities where a shortest path contains no control edges. As the players of the respective commodities find a feasible evading route in it where they cannot be caught, we no longer have to consider these commodities. Applied to our data for the German motorways this leads to a reduction in the number of commodities of about a third.

These modifications can improve the performance of our algorithms dramatically. However, the solution of the pCover, or more specifically an optimal control strategy for the inspectors, may be very sparse. This means that many control gantries may never be activated according to the optimal strategy and are therefore unnecessary. If we aim for an optimal control strategy when the gantries are already built, a sparse solution is difficult to explain in political terms. Furthermore, one should not overrate the mathematical optimization here and underrate the psychological impact of an adjusted optimized control strategy, since our optimization crucially depends on the assumptions we made.

Our solution for this problem is effective and yet simple. We assign a small *basic probability*  $q_0 > 0$  to every traffic control gantry and assure thereby that every control gantry may be activated. The rest of the total control capacity, i.e.  $k - |\mathcal{C}| \cdot q_0$ , is distributed optimally. Algorithmically this is done by replacing  $q_e \geq 0 \forall e \in \mathcal{C}$  with  $q_e \geq q_0 \forall e \in \mathcal{C}$  in constraint (3.1e) of the pCover.

What is a good choice for the basic probability  $q_0$ ? If we set  $q_0 = 0.01$ , say, every evading driver would risk to get caught with a probability of at least 1% for every control edge he uses. However, it is inappropriate to assign a constant value to

$q_0$ . The choice of  $q_0$  should rather depend on the total control capacity and the number of gantries, since otherwise we could violate the control capacity constraint if  $|\mathcal{C}| \cdot q_0 > k$ . Therefore, we choose a proportion of the total control capacity and distribute it uniformly among all control gantries. For our purposes we take 5% of the total control capacity to initialize the basic probability, i.e.  $q_0 = 5\% \cdot k / |\mathcal{C}|$ , while 95% of the control capacity are distributed optimally.

Again, the evading drivers can be caught at every control edge and have to take all detour edges instead, in order to ensure that they are not caught. Due to a phenomenon in decision making, called the *zero-risk bias*, this is an important aspect. The zero-risk bias states that people are willing to pay very high costs to obtain a certainty of 100%. This has been demonstrated in numerous surveys such as [BGK93] and [VMH87] and contradicts our assumption that the drivers act rationally. Thus, with the initialization of basic probabilities, our approach seems to be reasonable in both ways, mathematically and psychologically.

The introduction of basic probabilities was a first tribute to the psychology of decision making in our optimization problem. In the next section we will focus on another phenomenon of real life, namely the perception of the drivers.

### 3.3. pCover and Perception

In the pCover approach we assumed that the drivers are perfectly aware of the inspectors' control strategy. However, in real life the drivers cannot check whether a certain control gantry is activated at the moment. Therefore, we have to take the flawed perception of the drivers into account, when we evaluate a control strategy. The integration of bounded rationality and limited observations to Stackelberg games has been studied in [PJT<sup>+</sup>10]. We will adapt the approach stated there to our problem of finding an optimal control strategy for toll enforcement. With the flawed perception of the drivers we lose the robustness of the approach. Since they misjudge the probability to get caught on a certain route, they might not choose a good response to the inspectors' control strategy.

In order to restore the robustness, we integrate the bounded rationality of the drivers. Therefore, we assume that the drivers will not necessarily select a route which they take for optimal, but choose one which is worst for us. More specifically, we assume that among all routes where the drivers think that evading is better than paying, they take the worst-case route for us. Identically to the pCover, we assume that

they pay the toll and take a shortest path, if this is even cheaper than the expected penalty on this worst-case evading route. This approach to restore the robustness is different from the one in [PJT<sup>+</sup>10], where the drivers would choose the worst-case path in an  $\varepsilon$ -region around their expected optimum. However, our approach is much easier to integrate into the pCover and we will analyze the impact of this choice later on.

With regard to the drivers' limited observations we are looking for a suitable *perception function*  $perc : \Sigma_{insp} \rightarrow \Sigma_{insp}$ . For every control strategy  $q$ , this function gives the perceived control strategy  $q' := perc(q)$ . Note that we demand that the perceived strategy  $q'$  is still a feasible control strategy. While  $0 \leq q'_e \leq 1$  is an obvious requirement, one can argue about the need of  $\sum_{e \in \mathcal{C}} q'_e = k$ . With the assumption that the drivers know the control capacity  $k$ , our claim  $q' \in \Sigma_{insp}$  is justified. There are several reasons to assume the drivers' knowledge of  $k$ . It is unlikely that the drivers would underestimate  $k$  since we assume that  $k$  is relatively small. If, however, they overestimate  $k$ , the payoff for the inspectors cannot decrease as the drivers would misjudge their expected penalties even more. Another reason is that the control capacity could be leaked and in that case it would be crucial to allow  $\sum_{e \in \mathcal{C}} q'_e > k$ .

Now that we have established the foundations of the drivers' limited observations, we are ready for a short trip into the psychology of decision making. A lot of work has been done in the field of decision research in the last decades, mainly by psychologists. A considerable part of it concerns our question of how people judge likelihoods. A well-known phenomenon is that individuals severely misjudge small probabilities such as winning the lottery. Similarly, large probabilities are often underestimated as the birthday paradox shows.

The work of Fox and Rottenstreich [FR03] yields similar results. The first result is that individuals' judgements "are biased toward an ignorance-prior probability that assigns equal credence to each mutually exclusive event considered by the judge". This means that under uncertainty, we tend to assume a uniform distribution and explains why individuals overestimate small probabilities while they underestimate large ones.

The second result is that the observations only have a limited impact on the ignorance prior. With regard to the drivers this means that initially they assume a uniform control strategy and that their observations only lead to slight deviations of that assumption. More specifically, we adopt the approach from [FR03]

and [PJT<sup>+</sup>10] and assume that the perceived strategy  $q'$  is given by

$$q'_e = \alpha \cdot \frac{k}{|\mathcal{C}|} + (1 - \alpha) \cdot q_e \quad (e \in \mathcal{C}),$$

where  $0 \leq \alpha \leq 1$  describes the impact of the observations on the initial uniform distribution. The following proposition shows that  $q'$  is a feasible control strategy.

**Proposition 3.8** *For every  $\alpha \in [0, 1]$  the transformation  $q'_e = \alpha \cdot \frac{k}{|\mathcal{C}|} + (1 - \alpha) \cdot q_e$  defines a perception function.*

*Proof.* We have to ensure that  $q' \in \Sigma_{insp}$ . Let  $q \in \Sigma_{insp}$ . Obviously,  $q'_e \geq 0 \forall e \in \mathcal{C}$  as  $q_e \geq 0$  and similarly,  $q'_e \leq 1 \forall e \in \mathcal{C}$  for every  $0 \leq \alpha \leq 1$ . Finally,

$$\begin{aligned} \sum_{e \in \mathcal{C}} q'_e &= \sum_{e \in \mathcal{C}} \left( \alpha \cdot \frac{k}{|\mathcal{C}|} + (1 - \alpha) \cdot q_e \right) \\ &= \alpha \cdot \sum_{e \in \mathcal{C}} \frac{k}{|\mathcal{C}|} + (1 - \alpha) \cdot \sum_{e \in \mathcal{C}} q_e \\ &= \alpha \cdot k + (1 - \alpha) \cdot k = k. \quad \square \end{aligned}$$

The transformation from  $q$  to  $q'$  is not only justified but can also be computed easily, since it is affine in  $q$ . However, it implies that the drivers will perceive positive probabilities for traffic control gantries that are never activated, i.e.  $q'_e > 0$  while  $q_e = 0$ . This property is questionable as we can assume that the drivers will notice over time that a control gantry is never activated. We can easily solve this problem by assigning a positive basic probability to every gantry, as pointed out at the end of the last section. Indeed, we will demand throughout this section that  $q_e \geq q_0$  for a basic probability  $q_0 > 0$ .

The formulation of the  $\alpha$ -pCover is similar to the pCover. While the pCover problem could be formulated as an LP with potentially many constraints, we transform the  $\alpha$ -pCover into a mixed integer program (MIP) with even more constraints. The main problem thereby is that the players choose their routes according to the perceived probabilities but pay penalties according to the actual probabilities. Thus we assume for the  $\alpha$ -pCover that player  $P_i$  either pays the toll or chooses the worst-case route for us from

$$R_\alpha^i := \{r_{j,B}^i \in R^i : t_i > p \cdot \sum_{e \in B} q'_e\}.$$

To include this to the MIP we add a binary variable  $\mu_{j,B}^i$  for every route  $r_{j,B}^i$  of  $R^i$  which states if player  $P_i$  thinks that evading on this route is better than paying the toll for the commodity. More specifically,  $\mu_{j,B}^i = 1$  whenever  $r_{j,B}^i \in R_\alpha^i$ . For every  $\alpha \in [0, 1]$  and with  $q'_e = \alpha \cdot \frac{k}{|\mathcal{C}|} + (1 - \alpha) \cdot q_e$ , the  $\alpha$ -pCover can be stated as follows.

$$\max_{q, \mu, \lambda'} \sum_{i=1}^{|\mathcal{K}|} x_i \cdot \lambda'_i \quad (3.3a)$$

$$\text{s.t.} \quad \sum_{e \in \mathcal{C}} q_e \leq k \quad (3.3b)$$

$$(1 - \mu_{j,B}^i) \cdot t_i \leq p \cdot \sum_{e \in B} q'_e \quad \forall r_{j,B}^i \in R^i, \quad \forall i \quad (3.3c)$$

$$\lambda'_i \leq t_i \quad \forall i \quad (3.3d)$$

$$\lambda'_i \leq p \cdot (1 - \mu_{j,B}^i + \sum_{e \in B} q_e) \quad \forall r_{j,B}^i \in R^i, \quad \forall i \quad (3.3e)$$

$$q_e \geq q_0 \quad \forall e \in \mathcal{C} \quad (3.3f)$$

$$\lambda'_i \geq 0 \quad \forall i \in \{1, \dots, |\mathcal{K}|\} \quad (3.3g)$$

$$\mu_{j,B}^i \in \{0, 1\} \quad \forall r_{j,B}^i \in R^i \quad (3.3h)$$

The objective function (3.3a) maximizes the total revenue for the inspectors, while (3.3b) assures that the control capacity is not exceeded. As we try to maximize  $\lambda'_i$ ,  $\mu_{j,B}^i$  tends to be 0, since with  $\mu_{j,B}^i = 1$ , we would strengthen the upper bound for  $\lambda'_i$  in (3.3e). Therefore, (3.3c) ensures  $r_{j,B}^i \in R_\alpha^i \implies \mu_{j,B}^i = 1$  and that  $\mu_{j,B}^i = 0$  is feasible for  $r_{j,B}^i \notin R_\alpha^i$ . Furthermore, (3.3e) states that the expected penalty for  $r_{j,B}^i$  is an upper bound for  $\lambda'_i$  whenever  $\mu_{j,B}^i = 1$ , i.e. if  $r_{j,B}^i \in R_\alpha^i$ . According to (3.3d), the drivers will pay the toll if this is better than choosing a route from  $R_\alpha^i$ . This is a debatable constraint and we will discuss this aspect in the next section. Finally, we assign a basic probability to every control edge (3.3f).

The computation of the inspectors' strategy in the  $\alpha$ -pCover is nearly identical to the pCover and uses a cutting-plane approach. The pricing problem is still a constrained shortest path problem, but this time we look for a shortest  $s_i - t_i$ -path w.r.t.  $q'$  instead of  $q$ . In particular, we can apply the same methods to solve the  $\alpha$ -pCover as for the pCover. An interesting aspect, though, is a comparison between the solutions of the two approaches.

### 3.4. pCover vs. $\alpha$ -pCover

The similarities between these two models are obvious, but the differences may have a big impact on the solutions. As the drivers act optimally in a zero-sum game, the pCover is a robust approach, while we had to recover the robustness in the  $\alpha$ -pCover, since the drivers made suboptimal decisions. As the  $\alpha$ -pCover is still zero-sum, it would be dangerous to assume that the drivers will play according to their perceived optimum. Instead, we introduced the set  $R_\alpha^i$  of potential evading routes for the drivers of  $k_i$ , from which they choose a worst-case for us. This can still lead to suboptimal decisions and in this section we will study the possible difference in the respective objective values.

#### 3.4.1. Comparing the Objective Values

Let  $f(q)$  be the objective value of the pCover-LP (3.1), if the inspectors play according to  $q \in \Sigma_{insp}$ , i.e.

$$f(q) = \sum x_i \lambda_i(q) \text{ with } \lambda_i(q) = \begin{cases} t_i & \text{if } t_i < p \cdot \sum_{e \in B} q_e \quad \forall r_{j,B}^i \in R^i, \\ \min_{r_{j,B}^i \in R^i} \left( p \cdot \sum_{e \in B} q_e \right) & \text{else.} \end{cases}$$

Analogously, for a fixed  $\alpha$ , let  $g(q)$  be the objective value of the  $\alpha$ -pCover-MIP (3.3), if the inspectors play according to  $q \in \Sigma_{insp}$ , i.e.

$$g(q) = \sum x_i \lambda'_i(q) \text{ with } \lambda'_i(q) = \begin{cases} t_i & \text{if } t_i < p \cdot \sum_{e \in B} q'_e \quad \forall r_{j,B}^i \in R^i, \\ \min \left( t_i, \min_{r_{j,B}^i \in R_\alpha^i} p \cdot \sum_{e \in B} q_e \right) & \text{else.} \end{cases}$$

Recall that  $R_\alpha^i = \{r_{j,B}^i \in R^i : t_i > p \cdot \sum_{e \in B} q'_e\}$  is the set of routes for which the drivers think that evading is better than paying the toll. As the drivers misjudge the probabilities of being caught, their response to the inspectors' control strategy  $q$  should be worse than their response to  $q$  under correct judgement.

**Lemma 3.9** *Let  $q \in \Sigma_{insp}$ . Then  $\lambda_i(q) \leq \lambda'_i(q)$  for all  $i$  with  $k_i \in \mathcal{K}$ .*

*Proof.* Suppose  $t_i < p \cdot \sum_{e \in B} q_e \quad \forall r_{j,B}^i \in R^i$ , thus  $\lambda_i(q) = t_i$  and we have  $\lambda'_i(q) \geq t_i$  immediately as  $R_\alpha^i \subseteq R^i$ .

Now suppose  $\lambda_i(q) = \min_{r_{j,B}^i \in R^i} \left( p \cdot \sum_{e \in B} q_e \right)$  and  $\lambda_i(q) \leq t_i$ . Then we could have  $\lambda'_i(q) = t_i \geq \lambda_i(q)$  or we had

$$\lambda'_i(q) = \min_{r_{j,B}^i \in R_\alpha^i} \left( p \cdot \sum_{e \in B} q_e \right) \stackrel{R_\alpha^i \subseteq R^i}{\geq} \min_{r_{j,B}^i \in R^i} \left( p \cdot \sum_{e \in B} q_e \right) = \lambda_i(q).$$

□

This result implies that the objective value of the  $\alpha$ -pCover will always be greater or equal to the objective value of the pCover for the same instance.

**Theorem 3.10** *Let  $q^* \in \arg \max_{q \in \Sigma_{insp}} f(q)$  and  $p^* \in \arg \max_{p \in \Sigma_{insp}} g(p)$  considering the same instance. Then*

$$f(q^*) \leq g(p^*).$$

*Proof.* The proof is straightforward.

$$f(q^*) = \sum_i x_i \lambda_i(q^*) \stackrel{\text{Lemma 3.9}}{\leq} \sum_i x_i \lambda'_i(q^*) = g(q^*) \leq g(p^*).$$

□

What is the reason for this *perception gap*? The problem originates from the set  $R_\alpha^i$  and from the fact that a route which is actually optimal, i.e.  $t_i > p \cdot \sum q_e$ , is not perceived as good enough, i.e.  $t_i \leq p \cdot \sum q'_e$ , and hence not added to  $R_\alpha^i$ . Let us take a closer look at the set  $R_\alpha^i$  and the decision of the drivers in the  $\alpha$ -pCover.

### 3.4.2. The Perception Gap

When we set  $\alpha = 0$  the drivers perceive the actual control strategy and we have the same situation as in the pCover. Note that paying the toll is optimal here iff  $R_0^i = \emptyset$ , since then  $t_i \leq p \cdot \sum q_e \forall r_{j,B}^i \in R^i$ . If  $R_0^i$  is not empty, it contains the routes of commodity  $k_i$  that are cheaper for the drivers than paying the toll. For the pCover we assumed that  $P_i$  chooses a worst-case route out of  $R_0^i$  if  $R_0^i \neq \emptyset$ .

For the  $\alpha$ -pCover, however, we have to take the flawed perception of the drivers into account. A route  $r_{j,B}^i \in R_\alpha^i \setminus R_0^i$  cannot be chosen by  $P_i$  since the robustness constraint states that the drivers will always pay the toll if this is the best response

for them. With  $r_{j,B}^i \notin R_0^i$ , paying the toll is better than evading on  $r_{j,B}^i$ , even if the drivers are convinced of the opposite. One could argue that the drivers should pay for their suboptimal choice and should be fined on a route where they underestimated the expected penalty. However, this could lead to an optimal control strategy where we try to trick the drivers into evading when it is suboptimal. With so many parameters that we have to set correctly, this can be fatal and may destroy the robustness of our approach. We do not want a control strategy that encourages the drivers to evade, but a robust approach for modelling their behavior.

As the robustness constraint ensures that underestimating the expected penalty has no impact on the objective, this does not contribute to the perception gap. With an overestimation of the expected penalty, however,  $P_i$  might wrongly ignore a route  $r_{j,B}^i$ . Hence, if  $r_{j,B}^i \in R_0^i \setminus R_\alpha^i$ , evading on this route is not perceived as better than paying the toll and therefore,  $P_i$  will not choose it. The drivers of the respective commodity may have to pay more than they had to pay when they chose  $r_{j,B}^i$ , and this contributes to the perception gap. Moreover, this is the only possibility for the perception gap to develop. So, for a route  $r_{j,B}^i \in R^i$  an overestimation of the expected penalty is necessary for a contribution to the perception gap. Equivalently, if  $\alpha > 0$ ,

$$\begin{aligned}
& p \cdot \sum_{e \in B} q_e < p \cdot \sum_{e \in B} q'_e \\
\iff & \sum_{e \in B} q_e < \sum_{e \in B} \left( \alpha \frac{k}{|\mathcal{C}|} + (1 - \alpha) \cdot q_e \right) \\
\iff & \sum_{e \in B} q_e < \alpha \cdot \frac{k \cdot |B|}{|\mathcal{C}|} + (1 - \alpha) \cdot \sum_{e \in B} q_e \\
\iff & \alpha \cdot \sum_{e \in B} q_e < \alpha \cdot \frac{k \cdot |B|}{|\mathcal{C}|} \\
\stackrel{\alpha > 0}{\iff} & \sum_{e \in B} q_e < \frac{k \cdot |B|}{|\mathcal{C}|}.
\end{aligned}$$

Whenever the control intensity on  $r_{j,B}^i$  is less than with a uniform distribution, the probability to get caught is overestimated by the drivers. Considering a single control edge, the probability for a control on it is overestimated iff it is below average. We will use this observation to design an example where the pCover and the  $\alpha$ -pCover lead to different revenues for the inspectors.

Before we consider this example, we conclude our new results. Player  $P_i$  will choose an evading route  $r_{j,B}^i$  only if he thinks that it is better than paying the toll ( $r_{j,B}^i \in R_\alpha^i$ ) and if it really is ( $r_{j,B}^i \in R_0^i$ ). Thus we conclude that in the  $\alpha$ -pCover  $P_i$  chooses a worst-case route from  $R_\alpha^i \cap R_0^i$  if this is not empty, and pays the toll otherwise.

Now that we have analyzed what leads to the perception gap, we ask how large it can get. The following example demonstrates the difference of the pCover and the  $\alpha$ -pCover and shows that the perception gap can be arbitrarily large.

**Example** We consider the following digraph  $G = (V, E)$  with three disjoint commodities where  $x_1 = x_2 = x_3 = n \in \mathbb{N}$ . Furthermore  $t_1 = 160$ ,  $t_2 = t_3 = 40$  along with  $p = 200$  and finally,  $\mathcal{C} = E$ ,  $k = 1$  and  $\alpha = 3/7$ .

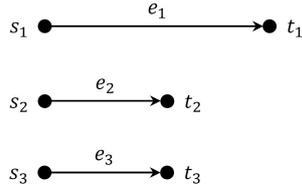


Figure 3.2.: Network where the perception gap can get arbitrarily large

First note that for  $P_1$  evading is better than paying the toll iff  $q_1 < 0.8$ , where  $q_i := q_{e_i}$ , since then  $p \cdot q_1 < t_1$ . Analogously, for  $P_2$  and  $P_3$  paying the toll is a best response iff  $q_2 \geq 0.2$  or  $q_3 \geq 0.2$ , respectively.

**pCover** We claim that  $q^* = (0.8, 0.2, 0)$  is an optimal control strategy for the pCover. Obviously,  $q_1 > 0.8$  and  $q_2 > 0.2$  are a waste of control capacity. On the other hand, for  $a \in [0, 0.8]$  and  $b \in [0, 0.2]$  the inspectors' payoff with a control strategy  $q = ((0.8 - a), (0.2 - b), (a + b))$  and best responses  $r_i$  for  $P_i$  is

$$\begin{aligned} \pi_{insp}(q, r_1(q), r_2(q), r_3(q)) &\leq n \cdot p \cdot (0.8 - a) + n \cdot p \cdot (0.2 - b) + n \cdot p \cdot (a + b) \\ &= n \cdot p \\ &= n \cdot (t_1 + t_2) \\ &= \pi_{insp}(q^*, r_1(q^*), r_2(q^*), r_3(q^*)) = f(q^*). \end{aligned}$$

The inequality in the first line is strict when  $a + b > 0.2$  while for  $a + b \leq 0.2$  we

have equality, since in the former case, i.e.  $a + b > 0.2$ , we had

$$\pi_{insp,3}(q, r_3(q)) = n \cdot t_3 < n \cdot p \cdot (a + b).$$

**$\alpha$ -pCover** For the  $\alpha$ -pCover  $p^* = (0.8, 0.1, 0.1)$  is optimal. The drivers of  $k_1$  will pay the toll, since  $R_0^1 = \emptyset$ . The perceived strategy of the drivers is  $p' = (0.6, 0.2, 0.2)$ , since

$$p'_2 = \frac{3}{7} \cdot \frac{1}{3} + \frac{4}{7} \cdot p_2 = \frac{10}{70} + \frac{4}{70} = \frac{1}{5}.$$

Therefore,  $P_2$  and  $P_3$  both decide to pay the toll as we have  $e_2 \in R_\alpha^2$ ,  $e_3 \in R_\alpha^3$ , and the payoff for the inspectors is  $g(p^*) = n \cdot (t_1 + t_2 + t_3)$ . The difference between the two objective values is  $n \cdot t_3$  and gets arbitrarily large if we increase  $n$ .

In this example the difference increased only when the objective value itself increased. While the difference  $g(p^*) - f(q^*)$  can be unbounded, the quotient may not be. More specifically, is there a number  $N \in \mathbb{N}$  such that  $\frac{g(p^*)}{f(q^*)} \leq N$ ? In our example we have

$$\frac{g(p^*)}{f(q^*)} = \frac{n \cdot 240}{n \cdot 200} = \frac{6}{5} = 1.2.$$

It seems hopeless to find an example where the quotient is arbitrarily large, but without a proof we can only conjecture the existence of an upper bound  $N$ .

### 3.4.3. Influence of $\alpha$

In the following we will analyze the impact of  $\alpha$  when modelling the perception of the drivers. Lemma 3.9 stated that the drivers would not pay less with a flawed perception than under correct judgement. We can derive a conjecture from this lemma: The better the judgement of the drivers, the less they have to pay. Essentially, this means that the optimal objective value of the  $\alpha$ -pCover increases when  $\alpha$  is increased.

Therefore, we consider an arbitrary instance of the  $\alpha$ -pCover and fix everything except of  $\alpha$ . Now let  $g^*(\alpha)$  denote the optimal payoff for the inspectors<sup>7</sup> in the  $\alpha$ -pCover for  $\alpha \in [0, 1]$ . Then we confirm the conjecture with the following theorem.

**Theorem 3.11** *Let  $\alpha_1, \alpha_2 \in [0, 1]$ . Then*

$$\alpha_1 < \alpha_2 \implies g^*(\alpha_1) \leq g^*(\alpha_2).$$

<sup>7</sup>This is the optimal objective value of the  $\alpha$ -pCover-MIP.

*Proof.* Suppose  $0 \leq \alpha_1 < \alpha_2$  and  $g^*(\alpha_1) > g^*(\alpha_2)$ . Recall that  $P_i$  pays the toll in the  $\alpha$ -pCover if  $R_\alpha^i \cap R_0^i = \emptyset$  and takes a worst-case route from  $R_\alpha^i \cap R_0^i$  otherwise. Thus,  $g^*(\alpha_1) > g^*(\alpha_2)$  implies that there is a route  $r_{j,B}^i \in R_{\alpha_2}^i \cap R_0^i$  with  $r_{j,B}^i \notin R_{\alpha_1}^i$  for some  $i$ . We derive a contradiction and show  $r_{j,B}^i \in R_{\alpha_1}^i$ .

At first,  $r_{j,B}^i \in R_{\alpha_2}^i$  gives us

$$\begin{aligned} t_i &> p \cdot \sum_{e \in B} \left( \alpha_2 \frac{k}{|C|} + (1 - \alpha_2) q_e \right) \\ \stackrel{q_2 > 0}{\iff} \frac{\alpha_1}{\alpha_2} t_i &> \frac{\alpha_1}{\alpha_2} p \cdot \sum_{e \in B} \left( \alpha_2 \frac{k}{|C|} + (1 - \alpha_2) q_e \right) \\ \iff \frac{\alpha_1}{\alpha_2} t_i &> \alpha_1 p \cdot \frac{|B| k}{|C|} + \frac{\alpha_1(1 - \alpha_2)}{\alpha_2} p \cdot \sum_{e \in B} q_e \end{aligned} \quad (*)$$

and from  $r_{j,B}^i \in R_0^i$  we get

$$\begin{aligned} t_i &> p \cdot \sum_{e \in B} q_e \\ \stackrel{\alpha_2 > \alpha_1 \geq 0}{\iff} \frac{\alpha_2 - \alpha_1}{\alpha_2} t_i &> \frac{\alpha_2 - \alpha_1}{\alpha_2} p \cdot \sum_{e \in B} q_e. \end{aligned} \quad (**)$$

Combining (\*) and (\*\*) leads to

$$\begin{aligned} \frac{\alpha_1}{\alpha_2} t_i + \frac{\alpha_2 - \alpha_1}{\alpha_2} t_i &> \alpha_1 p \cdot \frac{|B| k}{|C|} + \frac{\alpha_1(1 - \alpha_2) + \alpha_2 - \alpha_1}{\alpha_2} p \cdot \sum_{e \in B} q_e \\ \iff t_i &> \alpha_1 p \cdot \frac{|B| k}{|C|} + (1 - \alpha_1) p \cdot \sum_{e \in B} q_e \\ \iff t_i &> p \cdot \sum_{e \in B} \left( \alpha_1 \frac{k}{|C|} + (1 - \alpha_1) q_e \right) \end{aligned}$$

and thus  $r_{j,B}^i \in R_{\alpha_1}^i$ . □

For a given instance of the  $\alpha$ -pCover the optimal objective value increases monotonically with  $\alpha$ . As  $\alpha = 0$  describes the pCover, we obtain Theorem 3.10 as a corollary of Theorem 3.11.

This concludes our comparison between the pCover and the  $\alpha$ -pCover as well as our presentation of the models for optimizing the automatic toll enforcement. In the next chapter, we will apply our models to the German motorway network and determine an optimal toll control strategy.

## 4. Application to German Motorways

In this chapter we will apply the main approaches of the last two chapters to the traffic network of German motorways. Here we have over 2000 exits and 5000 sections with a total length of about 13 000 km. Truck drivers have to pay an average toll of 0.176 € per km and 302 control gantries are installed to detect toll evading truck drivers. If caught, these drivers normally have to pay a penalty which lies between 200 € and 400 €, but it can increase up to 20 000 € in case of recurrence. Our first issue is to apply the cover approach to the German motorway network in order to place 302 traffic control gantries. Afterwards, we compare this result with the actual placement of the control gantries.

In the second section we test the pCover and the  $\alpha$ -pCover approach and apply it to the real data of German motorways. We evaluate the respective results and compare them with the control strategy of uniform distribution.

The given data consists of the exits and sections of the German motorway network, together with real traffic data which is averaged over time. Based on this data we build our traffic network  $(G, l, \mathcal{K})$  which also contains a few trunk roads (dotted) and is shown in figure 4.1.

The implementations were done in the Python programming language and all computations were made on a PC with 8 processors at 3 GHz using the CPLEX solver.

### 4.1. Placing Control Gantries on German Motorways

In this section we will employ the cover approach to optimize the placement of 302 traffic control gantries. For the performance of our algorithms, it is of great help to simplify the traffic network in a preprocessing.

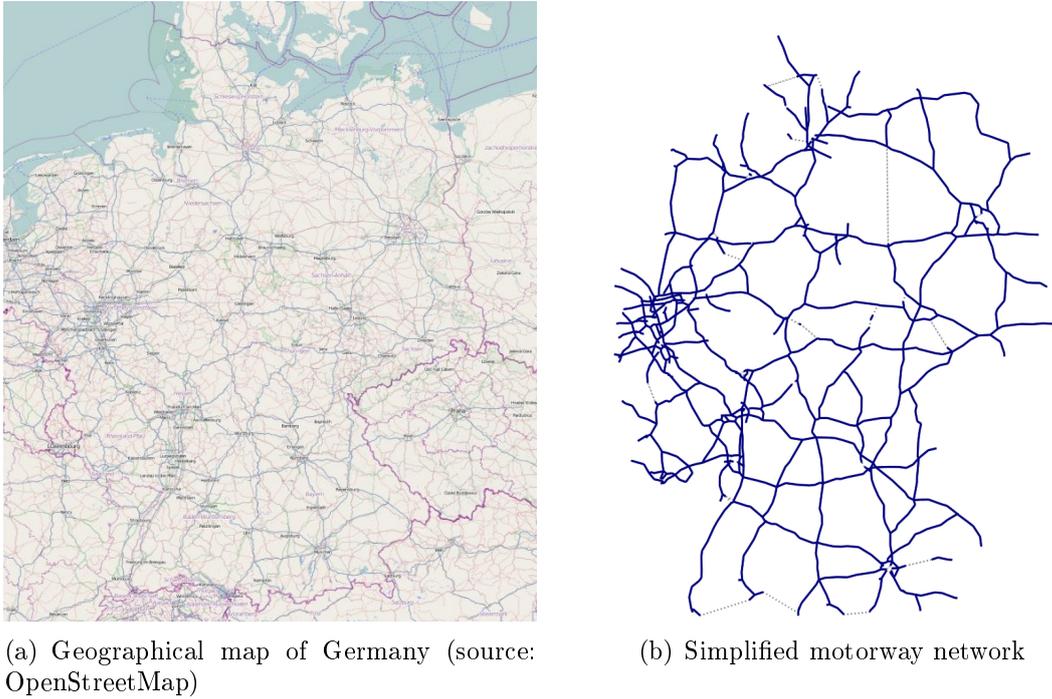


Figure 4.1.: Simplification of the German motorway network

**Simplification of the Network** The initial network contains too much detailed information that should be aggregated in order to reduce the solution time of the IP. Therefore, we simplify the traffic network in three steps. The first issue concerns the given traffic data. There are  $|V| \cdot (|V| - 1)$  possible commodities and by merging close nodes, we can reduce this number dramatically. If we halve the number of nodes, only a quarter of the initial commodities remains. Therefore, we identify short edges in the digraph and combine the ingoing and, respectively, the outgoing traffic of the endnodes. The combined traffic is assigned to the endnode which contributed more traffic. For our purposes we merged the traffic of two nodes, if their distance was less than 2 km, and thereby reduced the number of commodities by 60%.

In the next step we replace certain paths in  $G$  with a single edge. More specifically, a path  $P = (v_0, v_1, \dots, v_k)$  gets replaced with  $(v_0, v_k)$  if  $P$  is an induced subgraph of  $G$  and if no  $v_i$ ,  $i \in \{1, \dots, k-1\}$  is source or sink of any commodity. Furthermore, we set  $l_{(v_0, v_k)} = l(P)$  and define  $w_{(v_0, v_k)} := \max_{e \in P} l_e$  and  $wmax((v_0, v_k)) := e^*$  for an  $e^* \in P$  with  $w_{(v_0, v_k)} = l_{e^*}$ . With this simplification of the network we impose the additional constraint that on these paths at most one control gantry can be built. However, as the numerous sources and sinks of our commodities remain unchanged, this represents no restriction in practice.

Table 4.1.: Effects of the simplification of the traffic network for the cover approach

	# nodes	# edges	# commodities
initial	2 682	5 637	645 166
after Step 1	2 682	5 637	253 407
after Step 2	1 248	2 769	253 407
after Step 3	1 248	2 769	2 968

If we decide to build a traffic control gantry at  $(v_0, v_k) \in E$ , we actually build it at  $wmax((v_0, v_k))$  concerning the initial network. In conclusion we can say that the simplifications of Step 2 have few disadvantages but a big impact on the size of the traffic network (see Table 4.1).

When we take a closer look at the traffic data, we see that there are a lot of commodities  $k_i = (s_i, t_i, x_i)$  where  $x_i$  is very small. Without loss of generality, we can assume that the commodities are sorted in decreasing order w.r.t. the demand  $x_i$ . Figure 4.2 shows that it suffices to consider a decent number of commodities to cover a high amount of traffic. For this reason we reduce the number of commodities drastically in Step 3 by considering only the top commodities which represent a certain fraction  $frac$  of the total traffic  $\sum_i x_i$ . For the optimal placement of the control gantries we set  $frac = 1/3$  and reduce the number of commodities by 98.8%. This third step completes the preprocessing to simplify the traffic network. Table 4.1 summarizes the effects of the single steps and shows the success of our methods.

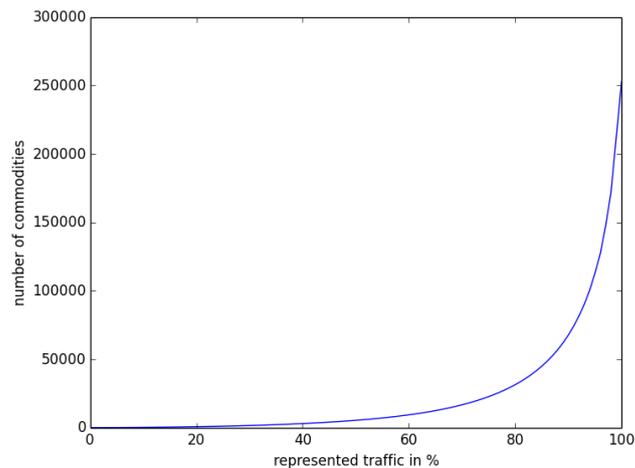


Figure 4.2.: Representation of traffic

**Solving the Cover-IP** For the application of the cover approach to the German motorway network, we have to set several parameters. Since we want to compare it with the actual placement of the traffic control gantries, we aim at placing  $m = 302$  control gantries optimally. In addition, we have to specify two parameters concerning the possible detour of the truck drivers.

The detour factor has been discussed in Chapter 2 (page 8) where we concluded that the additional costs for avoiding a control edge by taking a trunk road are significant. Here, we set  $u = 2$  which means that taking a trunk road is twice as expensive as taking the according motorway section.

We also mentioned in Chapter 2 (page 6) that we should rather overestimate the length of the detour that the drivers are willing to take, instead of underestimating it. Therefore, we will set  $\varrho$  to 0.1 and allow the drivers to make a detour of ten percent in relation to the shortest path.

In summary, we solve the cover approach with the following parameters:

$\varrho$	(possible detour)	:	10%
$u$	(detour factor)	:	2
$frac$	(considered traffic)	:	$\frac{1}{3}$
$m$	(control gantries)	:	302

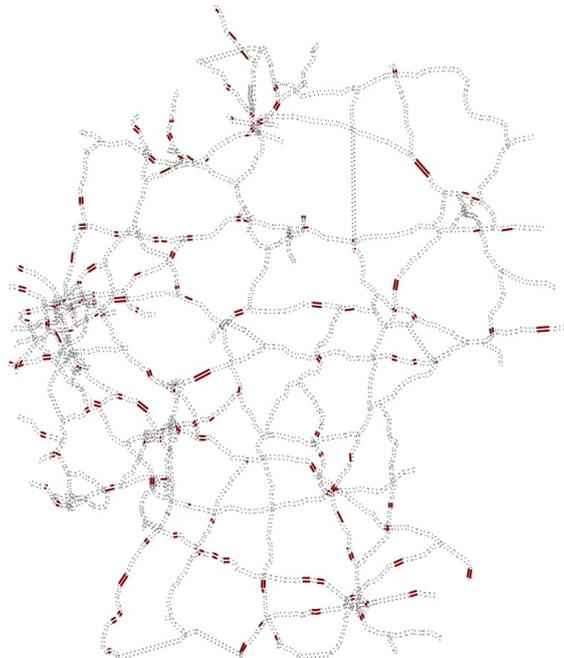
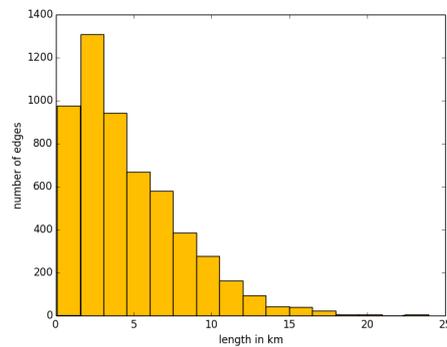
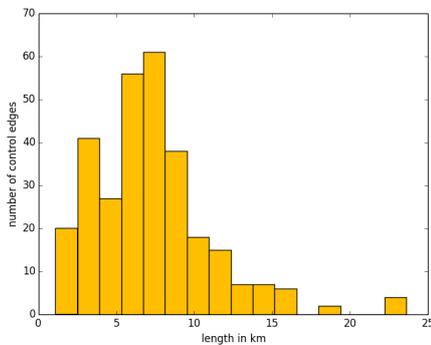


Figure 4.3.: Optimal Placement of traffic control gantries for the above instance

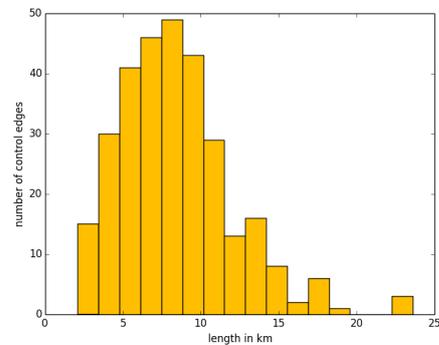
Furthermore, we extend the objective function as pointed out in (2.3) on page 11 and maximize  $(\sum_i x_i \cdot \delta_i) + \mu \cdot \sum_{e \in E} l_e \cdot y_e$  in order to support long sections. Finally,  $\mu$  and  $\varepsilon$  were both chosen sufficiently small.

The results of the optimization are presented in Figure 4.3 where parallel edges correspond to different directions on the motorway. With the exception of control gantries, all edges are represented by dotted lines, while the control gantries are marked red. The computation took 192 seconds and the restricted IP had to be solved 3 times.

As one can see, the control gantries are spread throughout the whole network and are placed preferably at long sections. Indeed, the latter effect can be confirmed by considering Figure 4.4 where the distribution of the edge lengths is illustrated. 4.4(a) shows that most of the edges in  $G$  are shorter than 5 km, while the average length of an edge in  $G$  is 4.7 km. The average length of a control edge in our solution (7.3 km) and the average length of an actual control edge (8.5 km) lie well above this value. In 4.4(b) and 4.4(c) the distribution of these lengths is demonstrated.

(a) all edges in  $G$ 

(b) control edges from our solution



(c) actual control edges

Figure 4.4.: Distribution of edge lengths

The results of the application of the cover approach to the German motorway network show that this approach can lead to a broad control possibility with multiple control gantries on long routes that are costly to avoid. For this instance, all of the criteria from Chapter 2 (page 8) are thus satisfied after the optimization.

With 66.8%, most of the considered traffic is covered regarding the optimal placement. In contrast, the actual placement leads to a cover rate of only 32.0% which means that two out of three drivers can evade a control if they take a detour of at most 10%. Of course, these values are bound to our model where we set  $u = 2$  and  $frac = 1/3$ . However, in this setting the effect of the optimization is obvious. With the optimized placement of the control gantries an evading driver is forced to take a detour of 15.4% on average to avoid a control. Considering the actual placement, this rate falls to 6.7% compared to a shortest path.

On the other hand, we should treat our results with caution. The objective of the cover-IP (2.2) is to maximize the covered traffic, but does not differentiate between traffic of long and short commodities<sup>1</sup>. However, it can be more advantageous to cover a long commodity instead of a short commodity, even if the latter has a higher demand. If we thereby force the drivers of the commodity to pay the toll, the total revenue for the inspectors can be much higher if we decide to cover long commodities. Therefore, we can easily adapt the objective (2.2a) and maximize

$$\sum_i \delta_i \cdot (x_i + a \cdot x_i \cdot l^i)$$

where  $a > 0$  determines how much we prefer long commodities with many drivers. Indeed, our solution could benefit from such an extension, as the average length of a covered commodity is 119 km.

Furthermore, we need to increase the fraction of considered traffic to achieve more reliable results. Currently, the computation collapses already for  $frac = 0.4$  as the memory capacity (16 041 MB) is not sufficient for solving the IP. Since our goal should be to apply the algorithm to traffic networks similar to the German motorway network and with  $frac \geq 0.8$ , this leaves room for improvement in the future.

A possible solution could be a branch and price approach where we make use of branching techniques in combination with column generation to solve the cover-IP in a memory-saving way. However, this method is very complex and difficult to implement and thus, we opted for solving our restricted IP using CPLEX.

---

<sup>1</sup>Long and short commodities are w.r.t.  $l^i$ .

## 4.2. Optimal Control Strategies on German Motorways

In this section we will compute optimal control strategies for the automatic toll control on German motorways using the pCover and the  $\alpha$ -pCover approach. To conceal the actual placement of the traffic control gantries we employ the solution from the previous section and use the control gantries that were placed with the cover approach. The obtained solution is also compared with an optimal control strategy concerning the actual placement of the control gantries. Similar to the preprocessing for the cover approach, we start with simplifying the traffic network to reduce the computation time.

**Simplification of the Network** The optimal placement of traffic control gantries crucially depends on the topology of the network. Thus, we wanted to leave the network similar to the original and merged close nodes only if they were very close together ( $< 2\text{km}$ ). For the control strategy, however, we already identified the important control edges and hence tend to weaken this condition. Therefore, we merge close nodes if their distance is less than 10 km and if they are not endnodes of a control edge. Clearly, we want to retain all control edges in our network and have to insert detour edges parallel to them in order to enable the drivers to avoid a control gantry on a trunk road. Again, we consider only a small subset of all commodities and set  $frac = 1/3$ . We will combine these steps as Phase 1 of the preprocessing and illustrate the effect in Table 4.2.

Phase 2 of the preprocessing contains only one step and that is to identify the commodities where a shortest path contains no control edge. The drivers of these commodities can take the shortest path and evade the toll without being caught. Thus, their total expenses in our game theoretic approach will be 0 and we delete these commodities to achieve another reduction of about a third in the number of commodities.

Table 4.2.: Effects of the simplification of the traffic network for the pCover approach

	# nodes	# edges	# commodities
initial	2 682	5 637	645 166
after Phase 1	911	2 085	1 475
after Phase 2	911	2 085	969

The biggest reduction was certainly achieved with the decision to consider only a certain amount of traffic. As we have seen in Figure 4.2 the number of commodities grows exponentially in the amount of traffic that we want to consider. Let us take a closer look at the consequences of this restriction.

Let  $N = \sum_i x_i$  be the total number of drivers and suppose that we consider only the top commodities which represent  $x\%$  of the traffic. If  $R$  is the optimal objective value of the pCover when we consider only the top commodities, the average costs for every driver are  $\frac{R}{x\% \cdot N}$ .

Now assume that the average costs per driver are the same if we optimize with  $frac = 1$ . Normally, this should be an upper bound for the average costs as we have a shortage of control capacity and with  $frac = x\%$  we can focus on a few commodities while with  $frac = 1$  many commodities will be ignored. However, the expected revenue for the inspectors would then be

$$N \cdot \frac{R}{x\% \cdot N} = R \cdot \frac{100}{x}. \quad (4.1)$$

If we optimized with  $frac = x\%$ , the total revenue for the inspectors would be higher than the optimal objective value of the respective pCover-LP, since the control strategy also affects some of the unconsidered drivers. In particular, we assume that these drivers pay half as much as the drivers of the considered commodities. The expected revenue for the inspectors if we optimized with  $frac = x\%$  would then be

$$R + (1 - x\%)N \cdot \frac{1}{2} \cdot \frac{R}{x\% \cdot N} = R + \frac{R}{2x\%} - \frac{R}{2} = R \cdot \frac{100 + x}{2x}. \quad (4.2)$$

With these assumptions we can conclude that optimizing with  $frac = x\%$  leads to a total revenue for the inspectors which represents  $\frac{100+x}{2}\%$  of the inspectors' total revenue when optimizing with  $frac = 1$ . For the choice  $frac = 1/3$  the inspectors gain  $2/3$  as much as they would earn with  $frac = 1$ . Compared to the effects on the size of the network and the associated reduction in the computation time, this is an acceptable loss of revenue.

**Optimizing the Control Strategy** In order to test the pCover and the  $\alpha$ -pCover approach on the German motorway network, we choose the same parameters as in the previous section. We have a total control capacity of  $k = 15$  and want to distribute it optimally. In accordance with Chapter 3 (page 33) we introduce a basic probability that we assign to every traffic control gantry. Therefore, we uniformly

distribute 5% of the total control capacity among all control gantries and optimize the distribution of the remaining 95%.

Furthermore, we assume that the average toll price per km is 0.176 € and that truck drivers have to pay a fine of 300 € if they are caught evading. Finally, we have to specify  $\alpha$  to describe the impact of the observations on the perceived strategy of the drivers in the  $\alpha$ -pCover. With  $\alpha = 0$  we get the pCover while  $\alpha = 1$  leads to the uniform distribution as perceived control strategy. Due to Theorem 3.11 an overestimation of  $\alpha$  is more severe than an underestimation and hence, we choose  $\alpha = 0.25$ .

Taken together, we compute an optimal control strategy with the following parameters:

$\varrho$	(possible detour)	:	10%
$u$	(detour factor)	:	2
$frac$	(considered traffic)	:	$\frac{1}{3}$
$k$	(control capacity)	:	15
$q_0$	(basic probability)	:	$\frac{5\% \cdot 15}{302}$
$t$	(toll costs per km)	:	0.176 €
$p$	(penalty)	:	300 €
$\alpha$	(perception bias)	:	0.25

The solution of the pCover-LP is demonstrated in Figure 4.5. The displayed control strategy differs substantially from the actual control. It presents the optimum of the pCover considering the placement of the control gantries from the last section and with our parameters. We see that in our solution, the control capacity is well distributed over the control gantries. Figure 4.6 illustrates this distribution more clearly. There are many control gantries with a small probability to be activated, but however, there are only 32 control gantries that were assigned the basic probability. Interestingly, the optimal solution of the  $\alpha$ -pCover contains 63 control gantries with basic probability. This coincides with our results on the perception gap in Section 3.4 where we concluded that the increase in the objective value of the  $\alpha$ -pCover originates from routes where the control intensity is below average<sup>2</sup>. It also proves the importance of introducing a basic probability, since otherwise we might get an optimal solution where certain control gantries are never used. However, our basic probability  $q_0 \approx 0.0025$  is very small and even with  $\alpha = 0.25$  the drivers severely

<sup>2</sup>This means that the control intensity is less than with a uniform distribution of the control capacity.

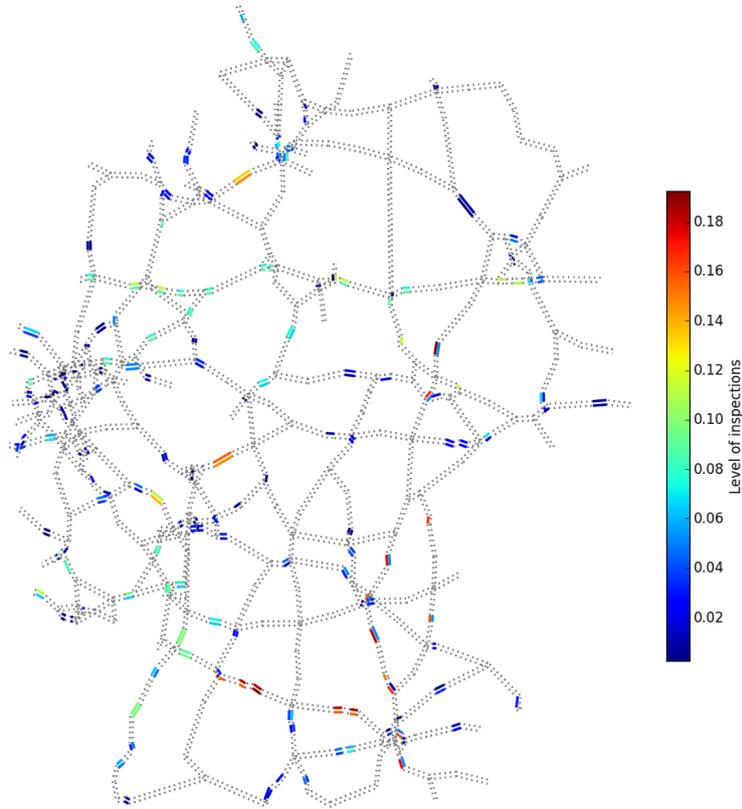


Figure 4.5.: Optimal control strategy for the pCover with fictitious control gantries

overestimate it ( $q'_0 \approx 0.014$ ). Increasing  $q_0$  reduces the misjudgement and may result in a more practical basic probability.

The impressive results of the optimization are shown in Table 4.3. The first two columns contain the optimal objective values of the pCover and the  $\alpha$ -pCover, respectively. The third column shows the revenues for the inspectors if their control strategy was a uniform distribution of the total control capacity. The computations were done for the optimized placement from the previous section and for the actual placement of the control gantries. In accordance with Theorem 3.10 we see that the optimal objective value of the  $\alpha$ -pCover is greater than the one of the pCover. Furthermore, our optimization led to a revenue increase of 29.4% (optimized placement) and 72% (actual placement), respectively, compared to a uniform distribution. We can also observe that the optimized placement of the control gantries leads to higher revenues for the inspectors than the actual placement. This is obviously due to the fact that our models for optimizing the control strategy are closely related to the cover approach, which we used to obtain the placement of the gantries.

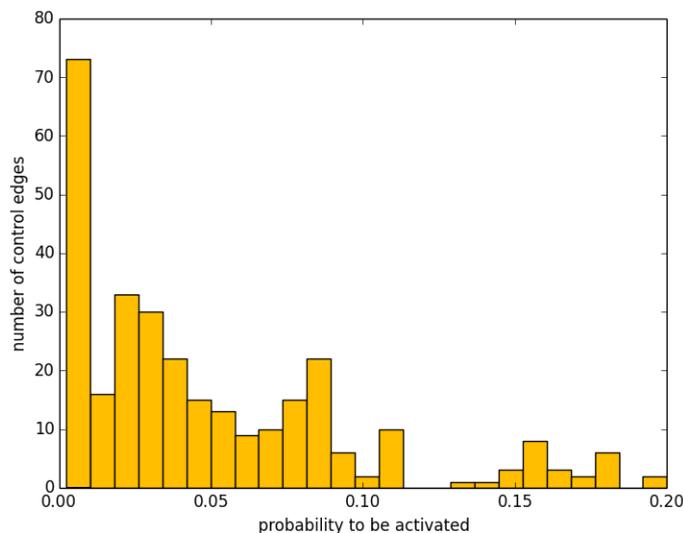


Figure 4.6.: Distribution of the control capacity

Again, we have to see that our optimization was done with  $frac = \frac{1}{3}$  and that it is necessary to increase this parameter. This time the limitation is not due to the disk space but due to the runtime of our algorithm. Solving the pCover with optimally placed control gantries took 13 iterations and 19 hours. The solution of the  $\alpha$ -pCover took even longer, namely 16 iterations and 23 hours. Most of the computation time is needed to solve the pricing problem, which is solved to optimality in every iteration. This means that we find a best response for every player to the current control strategy by solving a constrained shortest path problem for every  $k_i$ .

However, it could be sufficient to find a response better than the current one, i.e. a path  $P \in R^i \setminus \overline{R^i}$  which is better for  $P_i$  than the best path in  $\overline{R^i}$ . Therefore, we can use heuristics for finding a constrained shortest path. [Wei07] and [GBL03] describe a Lagrangian relaxation of the length constraint  $P \in R^i$  in order to find a short path w.r.t.  $l$  and  $q$ . Thus, a Lagrange multiplier is added to the objective function and the length constraint  $l(P) \leq (1 + \varrho) \cdot l^i$  is omitted. The violation of this constraint, however, leads to a decrease in the objective value and thus, in most cases the

Table 4.3.: Comparison of the optimal objective values

	pCover	$\alpha$ -pCover	uniform distr.
with optimized placement	10 782 822	10 929 693	8 334 972
with actual placement	9 005 410	9 110 717	5 234 455

optimal path  $P$  will satisfy our constraint. If  $P \in R^i \setminus \overline{R^i}$  we add this path to  $\overline{R^i}$  and iterate. Only if no path was added to any  $\overline{R^i}$ , we solve the constrained shortest path problem to optimality. This approach should reduce the computation time of the pCover drastically and allow us to test it with an appropriate value for  $frac$ .

However, the results for smaller networks such as the German subregion of North Rhine-Westphalia are much better. This federal state of Germany contains the Ruhr Area and thus represents a complex subnetwork of German motorways. A simplification similar to the above leads to a traffic network with 94 nodes, 227 edges and 623 commodities. Considering this instance, the pCover-LP with the above parameters and  $k = 1$ ,  $frac = 0.9$  is solved in 8.9 seconds and 4 iterations. This is a much better result which shows that our solution concept is perfectly suitable for smaller motorway networks. However, due to the complexity of the complete German motorway network, the integration of a heuristic for the constrained shortest path problem is an important issue for the future.

## 5. Some Further Problems

So far we have developed powerful models to optimize the placement of the control gantries as well as the control strategy. We employed cutting-plane methods to solve the cover and the pCover approach and presented a constrained shortest path algorithm to solve the pricing problem for the pCover. We also included the interesting aspect of the drivers' flawed perception of the actual control strategy and their resulting suboptimal decisions. Our models describe the respective problems in a proper way. This is also demonstrated by our applications of these models to German motorways where our approaches led to considerable results. We see that we have achieved a variety of valuable results.

In this final chapter of the thesis we want to address two problems of great importance in practice. The first one concerns our control strategy  $q$  for the stationary control. We need to specify how to conduct controls according to  $q$  since our capacity constraint restricts us to activate only  $k$  control gantries at the same time. We will develop and study an elegant response to this important issue.

The second problem is to develop an integrated approach for stationary and mobile controls. We will explain the arising problems and point out a possible solution in the second section.

### 5.1. Conducting Controls according to $q$

Assume that we have a control strategy  $q \in \Sigma_{insp}$  for a given traffic network and want to conduct controls according to  $q$ . We cannot activate a control gantry partially and on the other hand, we have to respect the capacity constraint which restricts us to activate at most  $k$  gantries. Therefore, we define the set of *controls* by

$$\Gamma = \left\{ (\gamma_e) \in \{0, 1\}^{\mathcal{C}} : \sum_{e \in \mathcal{C}} \gamma_e = k \right\}.$$

These are essentially the pure strategies for the inspectors and we should have chosen a mixed strategy  $(p_\gamma)_{\gamma \in \Gamma}$  instead of  $(q_e)_{e \in \mathcal{C}}$ . However,  $|\Gamma| = \binom{|\mathcal{C}|}{k}$  and therefore our

LP-formulation for the pCover would have been ridiculously complex and impossible to handle. Instead, we have to find a way such that the active time for the control gantries corresponds to  $q$  over a long period of time. When doing this, we might only switch between different strategies in  $\Gamma$  over time. Our approaches make use of basic probability features that can be found in [Geo08] for example.

The first idea is to commit to a control  $X_1 = \gamma^1 \in \Gamma$  for a certain amount of time, say one hour, and then choose  $X_2 = \gamma^2 \in \Gamma$  for the next hour, independent from previous controls. If we could derive a probability distribution  $p$  over  $\Gamma$  with  $\sum_{\gamma} \gamma \cdot p_{\gamma} = q$ , we could select  $\gamma$  according to  $p$  every time we want to change the control. Then,  $X_1, X_2, X_3, \dots$  is a sequence of independent and identically distributed (i.i.d.) random variables and we have

$$\mathbb{E}(X_i) = \sum_{\gamma} \gamma \cdot p_{\gamma} = q.$$

Furthermore, we have  $\mathbb{E}(|X_i|) = \mathbb{E}(X_i) < \infty$  and thus the law of large numbers gives us

$$\frac{1}{n} \sum_{i=1}^n X_i \xrightarrow{\text{a.s.}} \mathbb{E}(X_i) = q \quad \text{for } n \rightarrow \infty.$$

This shows that the resulting sequence of controls corresponds to our control strategy  $q$  over a long period of time and thus, this method may be suitable to implement our control strategy. The problem of this approach is that we have to determine a suitable distribution  $p$  and therefore have to solve a system of linear equations with  $\binom{C}{k}$  variables. In addition, we may be confronted with rounding errors as the variables may attain values very close to each other.

Fortunately, we can formulate a different approach to avoid this problem in an elegant and simple way. Our solution is to choose a control edge  $e \in \mathcal{C}$  according to a uniform distribution and to activate this gantry for an amount of time proportional to  $q_e$ . We can assume w.l.o.g. that  $0 < q_e < 1 \forall e \in \mathcal{C}$ , since otherwise if  $q_e = 0$  we ignore  $e$ , and if  $q_e = 1$  we activate  $e$  all the time and proceed with control capacity  $k - 1$ .

We define a *control process* to be a triple  $(e, t_0, t_1) \in \mathcal{C} \times [0, \infty) \times (t_0, \infty)$  where the control gantry  $e$  is activated from time  $t_0$  until  $t_1$ . We want to execute  $k$  control processes simultaneously at all times, but concurrent control processes must not involve the same control gantries. This can be interpreted as filling  $k$  slots  $S_1, \dots, S_k$  with control processes respecting the concurrence constraint. For  $t = 0$  and for every

slot or whenever a control process ends at time  $t$ , we invoke Algorithm 3 in order to fill an empty slot. Therefore, we need to determine the control processes at time  $t$ , i.e.  $(e^i, t_0^i, t_1^i)_{i=1, \dots, l}$ , where  $l \leq k$ ,  $t_0^i \leq t < t_1^i \forall i$ ,  $e^i \neq e^j \forall i \neq j$ .

---

**Algorithm 3:** algorithm for determining a control process

---

**Input:**  $t$ ,  $(e^i, t_0^i, t_1^i)_{i=1, \dots, l}$  control processes at time  $t$  with  $l < k$

**Output:**  $(e^i, t_0^i, t_1^i)_{i=1, \dots, l+1}$  control processes at time  $t$

```

1  $e^{l+1} := \emptyset$ ;
2 while  $e^{l+1} = \emptyset$  do
3   Choose  $e \in \mathcal{C}$  according to a uniform distribution;
4    $e^{l+1} := e$ ;
5   for  $i \in \{1, \dots, l\}$  do
6     if  $e = e^i$  then
7        $t_1^i := t_1^i + q_e$ ;
8        $e^{l+1} := \emptyset$ ;
9     end
10  end
11 end
12  $(e^{l+1}, t_0^{l+1}, t_1^{l+1}) := (e, t, t + q_e)$ ;
13 return  $(e^i, t_0^i, t_1^i)_{i=1, \dots, l+1}$ ;

```

---

In order to fill an empty slot we choose a control edge  $e \in \mathcal{C}$  uniformly at random and check if  $e$  is part of a current control process. If this is not the case, we assign the control process  $(e, t, t + q_e)$  to the empty slot. In the other case we extend the current control process involving  $e$  by the time  $q_e$  and iterate.

**Proposition 5.1** *Algorithm 3 terminates almost surely.*

*Proof.* The algorithm does not terminate only if our choice in line 3 leads to  $e \in \{e^1, \dots, e^l\}$  over and over again. As our choices are independent and as

$$Pr(e \in \{e^1, \dots, e^l\}) = \frac{l}{|\mathcal{C}|} < 1,$$

the algorithm does not terminate with probability 0. □

Let us reformulate the above process in a less formal way. We choose a control edge  $e$  uniformly at random and activate this control gantry for  $q_e$  hours, say. Thus, if  $q_e = 1/3$ , the control gantry is active for 20 minutes. If during these 20 minutes the

same control edge is chosen again in order to fill a different slot, the control gantry cannot be activated twice. Hence, we extend the current control process involving  $e$  by another 20 minutes and iterate. Now we have to prove two things:

- i) A single control process is not extended over and over again.
- ii) The resulting control corresponds to our control strategy  $q$  over a long period of time.

Whenever  $e \in \mathcal{C}$  is chosen uniformly at random, we ensure that the control gantry at  $e$  is activated for the time  $q_e$ . Thus, we number all of these choices and define the random variable

$$X_i := \begin{cases} q_e & \text{if } e \text{ was chosen in the } i^{\text{th}} \text{ choice,} \\ 0 & \text{else.} \end{cases}$$

Note that  $\sum_{i=1}^n X_i$  denotes a vector of dimension  $|\mathcal{C}|$  and the entries correspond to the total active time of the control gantries after  $n$  choices. Again,  $X_1, X_2, X_3, \dots$  are i.i.d. and

$$\mathbb{E}(|X_i|) = \mathbb{E}(X_i) = \sum_{e \in \mathcal{C}} \frac{1}{|\mathcal{C}|} (0, \dots, 0, q_e, 0, \dots, 0)^T = \frac{1}{|\mathcal{C}|} q.$$

With the law of large numbers we obtain  $\frac{1}{n} \sum_{i=1}^n X_i \xrightarrow{a.s.} \frac{1}{|\mathcal{C}|} q$  for  $n \rightarrow \infty$ . In other words, the resulting control corresponds to our control strategy over a long period of time and we have our second statement.

The first statement ensures that no slot is being overfilled considering a long period of time. Therefore, we consider a control edge  $e \in \mathcal{C}$  and set  $\mathbf{1}^T := (1, \dots, 1)$ . Then, after  $n$  choices, the total time that has been distributed over all  $k$  slots is  $\sum_{i=1}^n \mathbf{1}^T \cdot X_i$  and  $e$  was active for the time  $\sum_{i=1}^n (X_i)_e$ . We have to prove

$$Pr\left(\sum_{i=1}^n (X_i)_e \geq \frac{1}{k} \sum_{i=1}^n \mathbf{1}^T \cdot X_i\right) \rightarrow 0 \quad \text{for } n \rightarrow \infty, \quad (5.1)$$

i.e. the scheduled active time for  $e$  does not exceed the time window per slot. In particular, this implies that a single control process will not be extended over and over again. We prove (5.1) by considering the random variables  $Y_i^e = (X_i)_e - \frac{1}{k} \mathbf{1}^T \cdot X_i$

which are i.i.d. with expectation

$$\mathbb{E}(Y_i^e) = \frac{1}{|\mathcal{C}|} q_e - \frac{1}{k} \sum_{f \in \mathcal{C}} \frac{1}{|\mathcal{C}|} q_f = \frac{1}{|\mathcal{C}|} (q_e - 1).$$

Moreover,  $\mathbb{E}(|Y_i^e|) \leq \frac{1}{|\mathcal{C}|} (q_e + 1) < \infty$  and thus we can apply the law of large numbers to obtain

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n Y_i^e &\xrightarrow{a.s.} \frac{1}{|\mathcal{C}|} (q_e - 1) \quad \text{for } n \rightarrow \infty \\ \Leftrightarrow \frac{1}{n} \sum_{i=1}^n (X_i)_e - \frac{1}{n \cdot k} \sum_{i=1}^n \mathbf{1}^T X_i &\xrightarrow{a.s.} \frac{1}{|\mathcal{C}|} (q_e - 1) \quad \text{for } n \rightarrow \infty. \end{aligned}$$

Since  $\frac{1}{|\mathcal{C}|} (q_e - 1) < 0$ , this implies (5.1) and completes the proof of i).

We have found a simple way of conducting controls according to our control strategy. Our elegant solution allows us to ignore the set  $\Gamma$  when optimizing the use of traffic control gantries for toll enforcement. The following section considers another aspect of effective toll enforcement and is concerned with an integrated approach for optimizing mobile and stationary toll control.

## 5.2. Combining Stationary and Mobile Controls

In this thesis we extensively analyzed important issues for a stationary toll control using traffic control gantries. Concerning German motorways the truck toll is also enforced by inspection tours of mobile units. Several works such as [BOSS12] and [BBSS12] focused on the optimization of the mobile controls. An independent optimization of the stationary and the mobile control strategy can lead to inefficient results where the drivers of some commodities are caught multiple times, by a control gantry and by a mobile unit. This is obviously not what we aim for and thus we have to find a way to combine these approaches.

Optimizing the mobile and the stationary controls at the same time would probably be too complex and thus, we only have two possibilities. We could optimize the mobile control strategy and then take this strategy as additional input for optimizing the use of traffic control gantries. However, the possibilities of expanding a mobile control strategy with stationary controls are limited and thus it would be unsuitable to start with the mobile control strategy.

On the other hand, we could determine an optimal control strategy for stationary controls and then input it to a model for the optimization of mobile controls. Since mobile controls can be conducted network-wide, they are perfectly suited to cover still existing evading routes and to complete the coordinated control strategy.

Let us consider the model in [BBSS12] to optimize the inspectors' control tours and integrate our stationary control strategy. The model includes travelling costs for the drivers and can thus ignore the detour constraint. An artificial toll edge from  $s_i$  to  $t_i$  is inserted for every commodity to represent the tour of the toll paying drivers and thus, the weight of this edge is the travelling costs for the shortest  $s_i - t_i$ -path increased by the toll costs  $t_i$ . Every other edge consists of a basic weight, namely the travelling costs on it, and a penalty weight which is essentially the expected penalty for the drivers on this edge. The latter can be increased on every edge by assigning control capacity to it, i.e. by assigning a mobile control unit to it with certain probability.

In order to integrate the stationary control we can increase the basic weight of the control edges by  $q_e \cdot p$ . Thus, the expected penalty for getting caught by the respective control gantry is already included in the edge weight. This modification does not change the structure of the problem but it does affect the drivers in the intended way. The optimal strategy for this problem, i.e. the control strategy for the mobile units, now respects our control strategy for stationary controls. This represents the coordinated control strategy to enforce the toll more effectively.

At the very end of this thesis we should take another look at the issues that remained open. We developed powerful models to describe important aspects of stationary toll control. However, as mentioned in Chapter 4, we need to improve the performance of our algorithms to obtain more reliable results for the optimization. The integration of heuristics as described on page 53 may particularly contribute to this improvement. The combination of stationary and mobile controls is another important aspect for the future and may be solved as pointed out above. Such a combined control strategy may be the economically most effective way of toll enforcement on motorways.

# A. Appendix

## A.1. The Cut Problem is NP-hard for $k \geq 2$

Recall that the directed multicut problem and the cut problem are stated as follows:

**directed multicut problem**

Input:  $(G, (s_i, t_i)_{i=1, \dots, k}, m)$ .

Decision problem: Is there a directed multicut of size  $\leq m$ ?

**cut problem**

Input:  $(G, l > 0, \varrho, (s_i, t_i, x_i)_{i=1, \dots, |\mathcal{K}|}, n, d)$ .

Decision problem: Is there an edge set of size  $n$  that cuts a total traffic of  $\geq d$ ?

and that the directed multicut problem is NP-hard for  $k \geq 2$ . In order to prove the same result for the cut problem, let  $(G, (s_i, t_i)_{i=1, \dots, k}, m)$  be an instance of the directed multicut problem with  $k \geq 2$ . We create an instance of the cut problem by considering the same graph  $G$  with  $l_e = 1 \forall e \in E$  and the same end points of the commodities. Furthermore, we set  $x_i = 1 \forall i$ ,  $n = m$ ,  $d = k$ , and  $\varrho = |E|$ . A shortest  $s_i - t_i$ -path has length  $l^i \geq 1$  and a longest path in  $G$  has at most length  $|E|$ . Thus, we have  $l(P) \leq (1 + \varrho) \cdot l^i$  for every  $s_i - t_i$ -path  $P$  in  $G$  and hence,  $k_i$  is cut only if every  $s_i - t_i$ -path contains at least one control edge. With our choice  $d = k$  we ask for a cut of size  $n$  that cuts all the traffic, and thus we have

$$\begin{aligned}
 & \text{There is a cut of size } n \text{ that cuts a total traffic of } \geq d \\
 \iff & \text{There is a cut of size } n, \text{ such that all commodities are cut} \\
 \iff & \exists \mathcal{C} \subseteq E, |\mathcal{C}| = n = m : \forall i \forall s_i - t_i \text{ - paths } P : P \cap \mathcal{C} \neq \emptyset \\
 \iff & \text{There is a directed multicut of size } \leq m.
 \end{aligned}$$

Since the reduction from above is in polynomial time, we could efficiently decide the directed multicut problem, if we could decide the cut problem in polynomial time.

## A.2. Computing $R^i$ with Yen's Algorithm

The set of routes  $R^i$  for  $k_i$  contains all  $s_i - t_i$ -paths  $P$  in  $G$  with  $l(P) \leq (1 + \varrho) \cdot l^i$ . These paths can be computed with a modified version of Yen's algorithm for finding the  $K$  shortest loopless  $s - t$ -paths in a digraph with non-negative edge lengths [Yen71]. We start with a presentation of Yen's algorithm.

During the algorithm, the set  $A$  contains the shortest paths to be returned and the set  $B$  contains the candidates to be added to  $A$ . In the first part of the algorithm we find a shortest  $s - t$ -path  $A^1$ , for example using the Dijkstra algorithm [Dij59] and add  $A^1$  to  $A$ . We illustrate the main idea of the second part by determining the second shortest path. Suppose

$$A^1 = v_1^1, v_2^1, \dots, v_n^1, t,$$

then the second shortest path  $A^2$  has to coincide with  $A^1$  in at least one node, namely  $s$ . Suppose that it coincides with  $A^1$  in  $v_1^1, \dots, v_i^1$  and  $(v_i^1, v_{i+1}^1) \notin A^2$ . The path  $v_1^1, \dots, v_i^1$  is called the root  $R_i^2$  of  $A^2$  and a shortest  $v_i^1 - t$ -path in  $G \setminus (v_i^1, v_{i+1}^1)$  which is called the spur  $S_i^2$ , completes  $A^2$ .

In order to find  $A^2$  we compute for every  $i \in \{1, \dots, n\}$  a shortest  $v_i^1 - t$ -path in  $G \setminus (v_i^1, v_{i+1}^1)$  and store the resulting path  $A_i^2 = R_i^2 + S_i^2$  in  $B$ . Afterwards, we find a shortest path in  $B$  and add it to  $A$  as  $A^2$ .

This can be generalized to find the  $k^{\text{th}}$  shortest path  $A^k$ . There are already several candidates for  $A^k$  in  $B$  but we have to add the new candidates that arise when taking a root from  $A^{k-1}$ . The procedure is similar to the one for  $k = 2$ . For every root  $R_i^k = v_1^{k-1}, \dots, v_i^{k-1}$  we check for every path  $A^j \in A$  if  $R_i^k \subseteq A^j$  and delete the edge  $(v_i^j, v_{i+1}^j)$  from  $G$  if this is true. After deleting all these edges the graph  $G'$  remains. Then we compute a shortest  $v_i^{k-1} - t$ -path  $S_i^k$  in  $G'$  and add  $R_i^k + S_i^k$  to  $B$ . In the end we determine the shortest path in  $B$  and add it to  $A$ .

**Algorithm 4:** Yen's algorithm for  $K$  shortest loopless paths

---

**Input:**  $(G, l_e \geq 0, s, t \in V, K)$   
**Output:**  $K$  shortest  $s - t$ -paths in  $G$  w.r.t.  $l$

```

1  $A^1 :=$  shortest  $s - t$ -path in  $G$ ;
2  $A := A^1$ ;
3  $B := \emptyset$ ;
4 for  $k \in \{2, \dots, K\}$  do
5   for  $i \in \{1, \dots, |A^{k-1}|\}$  do
6      $R_i^k := v_1^{k-1}, \dots, v_i^{k-1}$ ;
7     for  $j \in \{1, \dots, k-1\}$  do
8       if  $R_i^k = R_i^j$  then
9         | Remove  $(v_i^j, v_{i+1}^j)$  from  $G$ ;
10        end
11      end
12      if  $\exists v_i^{k-1} - t$ -path in  $G$  then
13        |  $S_i^k :=$  shortest  $v_i^{k-1} - t$ -path in  $G$ ;
14        | Add  $R_i^k + S_i^k$  to  $B$ ;
15      end
16    end
17    if  $B = \emptyset$  then
18      | return  $A$ ;
19    end
20    Add shortest path in  $B$  to  $A$ ;
21    Restore edges to  $G$ ;
22 end
23 return  $A$  ;

```

---

In order to compute  $R^i$  with Yen's algorithm, we adapt it and extend the input by  $lmax > 0$ . Now we modify line 14 and add  $P := R_i^k + S_i^k$  to  $B$  only if  $l(P) \leq lmax$ . When we choose  $K$  big enough this modified version of Algorithm 4 determines all  $s - t$ -paths  $P$  in  $G$  with  $l(P) \leq lmax$  and we can use it to compute  $R^i$ .

# Bibliography

- [BBSS12] R. Borndörfer, J. Buwaya, G. Sagnol, and E. Swarat. Optimizing toll enforcement in transportation networks: a game-theoretic approach. Technical Report 12-47, ZIB, 2012.
- [Bel58] R. E. Bellman. On a Routing Problem. *Quarterly of Applied Mathematics*, 16:87–90, 1958.
- [BGK93] J. Baron, R. Gowda, and H. Kunreuther. Attitudes toward managing hazardous waste: What should be cleaned up and who should pay for it? *Risk Analysis*, 13(2):183–192, 1993.
- [BOSS12] R. Borndörfer, B. Omont, G. Sagnol, and E. Swarat. A stackelberg game to optimize the distribution of controls in transportation networks. Technical Report 12–15, ZIB, 2012.
- [BSS11] R. Borndörfer, G. Sagnol, and E. Swarat. An ip approach to toll enforcement optimization on german motorways. Technical Report 11-42, ZIB, 2011.
- [CLR05] M.-C. Costa, L. Létocart, and F. Roupin. Minimal multicut and maximal integer multiflow: a survey. *European Journal of Operational Research*, 162:55–69, 2005.
- [CS06] V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90. ACM, 2006.
- [DB01] I. Dumitrescu and N. Boland. Algorithms for the weight constrained shortest path problem. *International Transactions in Operational Research*, 8(1):15–29, 2001.
- [Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

- [DJP<sup>+</sup>92] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiway cuts. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 241–251. ACM, 1992.
- [FF62] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton university press, 1962.
- [For56] L. R. Ford. Network flow theory. Technical Report P-923, The Rand Corp., 1956.
- [FR03] C. R. Fox and Y. Rottenstreich. Partition priming in judgment under uncertainty. *Psychological Science*, 14(3):195–200, 2003.
- [GBL03] M. Grötschel, R. Borndörfer, and A. Löbel. Duty scheduling in public transit, 2003.
- [Geo08] H.O. Georgii. *Stochastics: Introduction to Probability and Statistics*. Walter De Gruyter, 2008.
- [GVY93] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing*, 25:698–707, 1993.
- [GVY94] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in directed and node weighted graphs. In *Automata, Languages and Programming*, pages 487–498. Springer, 1994.
- [HZ80] G. Y. Handler and I Zang. A dual algorithm for the constrained shortest path problem. *Networks*, 10(4):293–309, 1980.
- [JTP<sup>+</sup>10] M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, M. Tambe, and F. Ordóñez. Software assistants for randomized patrol planning for the lax airport police and the federal air marshal service. *Interfaces*, 40:267–290, 2010.
- [Kar72] R. M. Karp. Reducibility Among Combinatorial Problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

- 
- [KV06] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. Algorithms and Combinatorics. Springer, 2006.
- [Mye91] R.B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.
- [PJT<sup>+</sup>10] J. Pita, M. Jain, M. Tambe, F. Ordóñez, and S. Kraus. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15):1142–1171, 2010.
- [SSZ04] M. Saks, A. Samorodnitsky, and L. Zosin. A lower bound on the integrality gap for minimum multicut in directed networks. *Combinatorica*, 24:525–530, 2004.
- [VMH87] W. K. Viscusi, W. A. Magat, and J. Huber. An investigation of the rationality of consumer valuations of multiple health risks. *The RAND journal of economics*, 18(4):465–479, 1987.
- [vS34] H. von Stackelberg. *Marktform und Gleichgewicht*. J. Springer, 1934.
- [Wei07] S. Weider. *Integration of Vehicle and Duty Scheduling in Public Transport*. PhD thesis, 2007.
- [Yen71] J. Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.
- [YJJ<sup>+</sup>12] Z. Yin, A. X. Jiang, M. P. Johnson, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, M. Tambe, and J. P. Sullivan. Trusts: Scheduling randomized patrols for fare inspection in transit systems. In *IAAI*, 2012.

## **Selbstständigkeitserklärung**

Ich versichere, dass ich die vorliegende Masterarbeit selbstständig und nur unter Zuhilfenahme der angegebenen Quellen angefertigt habe.

Berlin, 27. November 2013