

RALF BORNDÖRFER GUILLAUME SAGNOL
THOMAS SCHLECHTE ELMAR SWARAT

Optimal Duty Rostering for Toll Enforcement Inspectors

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

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

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

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

Optimal Duty Rostering for Toll Enforcement Inspectors *

Ralf Borndörfer[†] Guillaume Sagnol[†] Thomas Schlechte[†]
Elmar Swarat[†]

Revised Version, 03.12.2014

Abstract

We present the problem of planning mobile tours of inspectors on German motorways to enforce the payment of the toll for heavy good trucks. This is a special type of vehicle routing problem with the objective to conduct as good inspections as possible on the complete network. In addition, we developed a personalized crew rostering model, to schedule the crews of the tours. The planning of daily tours and the rostering are combined in a novel integrated approach and formulated as a complex and large scale Integer Program. The main focus of this paper extends our previous publications on how different requirements for the rostering can be modeled in detail. The second focus is on a bi-criterion analysis of the planning problem to find the balance between the control quality and the roster acceptance. Finally, computational results on real-world instances show the practicability of our method and how different input parameters influence the problem complexity.

1 Introduction

Crew Scheduling and *Crew Rostering* are very important planning problems arising in several industries and applications. In this paper we will combine it

²Zuse Institute Berlin (ZIB), Takustr. 7, 14195 Berlin-Dahlem, Germany, Email {borndorfer, sagnol, schlechte, swarat}@zib.de

*This work was funded by the Federal Office for Goods Transport (BAG).

with a Vehicle Routing Problem (VRP), which is one of the most important and basic planning problems in Combinatorial Optimization and Operations Research, see Toth and Vigo (2002) for an overview.

The background of the studied problem is that the increase of individual traffic requires from public authorities to spend much investments on extensions or on maintenance of the road network. Therefore in many countries tolls were introduced, especially on motorways, to finance the growing investments. We focus here on the case of Germany where a distance-based toll on motorways and on some federal roads was introduced in 2005. All trucks with more than 12 tonnes vehicle weight have to pay a toll depending on their route distance and their emission class.

The enforcement of the toll is the responsibility of the German Federal Office for Goods Transport (BAG). It is both done by 300 stationary *control gantries* and by tours of about 300 mobile control teams on the complete motorway network. The teams, also called *control groups*, consist mostly of two inspectors, but in some cases of only one. Each team can only control toll roads in its associated *control area* or *region*, close to their depot. Germany is subdivided into 21 of those control areas.

Our challenge is to compute an optimal toll enforcement which aims at controlling as much as possible trucks with the available personnel. This optimization problem is called *Toll Enforcement Problem (TEP)* and it was first introduced in Borndörfer et al (2012c). The TEP is a combination of Tour Planning and Duty Rostering. We will explain later in more detail why both problems must be integrated in this setting. In Borndörfer et al (2012b) a case study of the TEP is presented, that shows the benefit of using the TEP for the planning of toll enforcement. Since then the TEP gained acceptance in practice. Hence, it is worth to give a more detailed presentation of this approach, especially with regard to the modeling aspects, that also includes the enhancements to handle bounds on night and weekend-duties.

In this paper we extend the work of Borndörfer et al (2012b) into several directions. Section 2 presents a comprehensive discussion that situate the TEP in the wide research fields on Vehicle Routing Problems and Crew Rostering. Then, Section 3 recalls the planning problem in general with a short description of an Integer Programming (IP) formulation for the tour planning and a graph model for the rostering of the crews. In contrast to Borndörfer et al (2012c,b), where the rostering model was rudimentary introduced due to space limitations, we provide in Section 4 an in-depth analysis of how real-world requirements can be modeled. This is the main

focus of this paper.

Next, in Section 5 our integrated approach is motivated by demonstrating the limits of a sequential approach. Moreover, Section 6 discusses the bi-criterion nature of the integrated model in more detail by utilizing the classical concept of Pareto-optimality. This is to analyse the relation between control quality and roster acceptance. While in Borndörfer et al (2012b) it was mainly stated that our approach helps to compute more employee-friendly plans, we provide in Section 7 an extensive computational study on different model parameters for real-world instances given from the productive operation at BAG. This demonstrates the tractability and applicability of the mathematical solution approach. Section 8 concludes and indicates further directions for future research.

2 Problem classification

To express it in simple words the TEP solves the following problem: On each day of the planning horizon we route control groups through the network to control trucks but with a very detailed view on the availability of crews that requires a detailed construction of a duty roster. Hence, we start with a comparison of our model to classical VRPs and to those with profits. Like in many other VRPs there is a limitation on the length of a tour, here according to daily working time limitations. Similar to the classical constraint that each customer must be served, there are constraints that each network part must be covered by a control from time to time. As in VRPs with Time Window the temporal aspect of the inspector routing is also very relevant since the amount of traffic changes a lot on different days and times of day.

A slight difference to many VRP models is that our routing graph is not complete since deadhead trips, i.e., driving on non-toll roads, are not permitted. But the main difference is the following: Most of VRP solutions consist of a set of tours, but availability of drivers is ignored. Hence, the feasibility of crew assignments is not part of the classical algorithms. But, in our application there are a lot of rostering rules to cope with to determine the set of available inspectors on each day. This detailed view on the crew availability leads to a relevant problem extension in comparison to classical VRPs.

We have chosen as our main objective that controls should be planned in order to maximize the number of controlled vehicles. Therefore a maxi-

mization problem with limited resources has to be solved. Since we assign a profit value to each motorway section to control, our problem relates to a *Team Orienteering Problem (TOP)* or a *Selective Vehicle Routing Problem*, see Boussier et al (2007); Archetti et al (2013b). In the case of only one vehicle this is known as the *Orienteering Problem*, a variant of the *TSP with profits*. For a recent publication on the Orienteering Problem we refer the reader to Vansteenwegen et al (2011) and for the more general case of the TSP with profit see Feillet et al (2005) for a literature survey. Archetti et al (2013a) studied another variant of a TSP with profits in case of several vehicles, namely the *Capacitated Team Orienteering Problem* and the *Capacitated Profitable Tour Problem*. There are two main differences to Team Orienteering problems, studied in general. First, our restriction of the available personnel can not simply be expressed by (travel) cost constraints. Second, the publications on TOPs ignore the fact that the problem has to be solved on each day over several weeks with the same crews.

Our rostering model is very similar to the approach of Cappanera and Gallo (2004). We also use a 0 – 1-multi-commodity flow model where each employee complies with a commodity. Equally, the rosters correspond to paths in a suitable planning graph with a layer for each day of the planning horizon. Our main difference and extension to this approach is that our objective is not to maximize the number of covered activities or duties. There are no uncovered activities in the TEP since the tours are chosen simultaneously to the assignment of inspectors to those tours. Therefore, each node in the rostering graph of the TEP corresponds to potential duties and not to fixed duties. This increases on the one hand the number of activities but on the other hand it eliminates some symmetry in the rostering problem.

Another related publication to our rostering approach is a paper by Kohl and Karisch (2004), also dealing with airline crew rostering. They provide an extensive description on different requirements for the crew rostering and how they are modeled. As a solution approach first rosters were generated and then an optimal subset of them is chosen in the master problem. This is different to our model where a flow based formulation in a roster graph is used. Furthermore, the TEP adopts some modeling techniques that are both used by Cappanera and Gallo and by Kohl and Karisch. Section 4 will present this in more detail.

Another application for rostering is nurse scheduling, see Burke et al (2004) for an overview on solution methods. There they present some approaches to mathematical programming not far from our method, but the

main difference to our problem is that in a hospital a certain staff coverage is mandatory on each day and time of day. An overview on general rostering applications and methods can be found in Ernst et al (2004). They provide a classification of solution methods where our attempt relates to the area of mathematical programming and in particular to network approaches. The authors claim that by using network approaches good quality solutions for difficult real-world applications are still not easy to compute. Therefore, our paper delivers a valuable impact in the field of rostering, since it presents a successful use of mathematical programming to a real-world rostering problem.

As an alternative approach, in a companion work we have proposed a game theoretic approach that takes into account the behaviour of drivers and fare evaders to distribute the controls in a more strategical way, see Borndörfer et al (2012a) and Borndörfer et al (2013). The output there is not a roster or a tour plan but a probability distribution of the inspectors force over all arcs of the network. We plan to integrate this planning strategy in our optimization model in a follow-up work.

To the best knowledge of the authors there is no optimization approach to inspection scheduling that includes rostering in the literature yet. There are recent approaches on optimizing the schedule of Federal Air Marshals Tsai et al (2009), on randomized patrol planning for the LA airport police Pita et al (2008) and on scheduling randomized patrols for fare inspection at LA Metro Yin et al (2012). All of them apply game theoretic methods. Other related publications deal with scheduling of highway patrols Jiann-Sheng and Tze-Chiang (2010), or with the scheduling of security teams in mass transit railway networks Lau and Gunawan (2012). The application of Lau and Gunawan is not far from our approach, but they restrict their model to a one-day planning horizon. For a single day a detailed schedule is determined, but the sequence of several duties of the same patrol team over several days is not taken into account. There is no crew rostering in their model. Another similar approach Thorlacius and Clausen (2010) is about the scheduling of ticket inspectors in local trains in Copenhagen, Denmark. There they optimize a temporal scheduling model of the inspectors but in contrast to our approach neither an exact route determination nor a crew rostering is considered.

The following Section 3 gives a partly informal description of the TEP and afterwards in Section 4 a more in-depth analysis of legal and organisational requirements for the rostering part of the TEP is presented.

3 The Toll Enforcement problem

In the TEP, the challenge is to integrate and to optimize the vehicle routing and the personalized crew rostering. Therefore, we developed a novel integrated approach that solves both subproblems. In this report, mainly a separate view on the constraints of each (sub-)problem is used, but we will examine the bi-criterion character of the objective function later in more detail. The coupling constraints, connecting both problems in the final IP formulation, will be presented at the end of this section. A typical problem instance is to produce a monthly schedule for one control area with given personal data and resources, e.g., inspectors with working time accounts, feasible routes, vacations, and so on.

We point out, that in the toll control setting it is not possible to ignore the availability of crews. Each control tour can only be conducted by a few amount of inspectors. Namely, the sections of the tour can only be controlled by a group that has its home depot not too far away, since each tour must start and end at the home depot of its associated group. Therefore, a sequential planning strategy (compute control tours first and then crew rosters) will tend to fail. We will discuss this in more detail in Section 5.

The assignment of a crew to a tour on a certain day must fit within a feasible *crew roster*, which defines the duties for several weeks and respects several restrictions regarding the sequence of duties. Minimum rest times, daily working times, vacations or maximal amounts of consecutive working days are examples of important requirements for the planning of rosters. Hence, we use a personalized *crew-* or *duty roster planning* in our model.

We see our approach as an integrated model, since it simultaneously solves tasks that are usually separated and solved sequentially. Traditionally, in railway Ernst et al (2001) or airline Cappanera and Gallo (2004) crew rostering problems, a timetable or flight plan is generated first. In a second step the rolling stock or the aircrafts and thereafter duties or pairings are scheduled. In the toll enforcement case these two scheduling problems are neglected since a vehicle is permanently assigned to an inspector team and all duties are uniform according to their duration, to their length and to the numbers of breaks. In a third step for (individual) crew members rosters over several weeks are generated that must cover all pairings or duties. In those models duties or pairings (or in case of integrated duty scheduling and rostering tasks) are an input of the model. But in the TEP tasks and duties are generated at the same time when the personalized roster is generated.

The coupling constraints guarantee that crew members are only assigned to tours carried out in their local area.

The following subsections discuss first the Tour Planning and second the Crew Rostering problem each providing an IP formulation for the corresponding subproblems. Finally, the coupling constraints that lead to an integrated formulation are presented.

3.1 The Tour Planning Problem

In the TEP, the daily tours should guarantee a network-wide control that takes given spatial and time dependent traffic distributions into account. To say it in one sentence: We model the subproblem of finding daily control tours by using a space-time network and formulate a corresponding optimization problem as an Integer Program (IP). Feasible tours correspond to paths in the network. We call this subproblem the *Tour Planning Problem (TPP)*. In the following we will describe some aspects that extend our brief explanations in previous publications but the model remains the same.

We divide the network into *sections*. A section is a subpart of the network with length of approx. 50-70 kilometers, where a team can be assigned to for a control during a certain time interval, e.g., four hours. The (overlapping) partition of the network into sections is mandatory for planning tours that map requirements from real-world. Namely, the exact course of a control can not be anticipated since catching an evader leads to initiating an administrative offense procedure. The duration of this procedure varies greatly. After that or even after a short control on a parking area it might be essential to change the direction on the motorway. Therefore, the inspectors need some degrees of freedom in space and time.

The problem to solve is to determine a sequence of sections and a time frame, when the tour takes place. This clearly is a problem separated from rostering since the route planning itself only considers the time horizon of one day. During a tour sections can follow in a sequence if they have at least one motorway junction in common. This guarantees that there are no deadhead trips between parts of the control tour. The time frame of a tour is important, since the profit values assigned to sections vary during different points of time. For example, it might be useful to control some sections in the morning and others in the evening or even later. Therefore, in our setting profit values are guided by the historic number of trucks that pass through a section during some time interval, called Δ . This rewards a

control of highly utilized sections more than of sections with low traffic. The background graphic in Figure 1 gives an example for such an estimation of the number of trucks on a section during hourly time intervals.

Each duty, and respectively each tour, has a duration of eight hours, plus setup and post-processing times, and a break in the middle. According to the practical requirements discussed at the beginning of this subsection, only values of two or four hours are appropriate for Δ , and respective for the duration of a section control. For example in daily operation $\Delta = 4h$ is used. This corresponds to the control of one section, a break, and a second control section.

Since the set of admissible sections is different among the control groups, a single commodity flow model does not suffice to solve the problem. Therefore, the TPP correlates to a Multi-Commodity Flow Problem. Let P be the set of all paths, that represent feasible control tours and $P_{f,j} \subset P$ the set of all paths that are feasible for group $f \in F$ and start at day $j \in J$. In addition for a section $s \in S$, the set of all paths $p \in P$ that visit section s is denoted by P_s . Let further T be the set of all time intervals with length Δ during a day. Then $P_{j,s,t}$ corresponds to all paths controlling s at day j during interval $t \in T$. By κ_s the minimum control quota, i.e., the minimum number of control visits (with duration Δ) on section s during the planning horizon, is indicated. Then let $w_{s,t_i} \in \mathbb{R}^+$ be the profit of controlling s during time interval $t_i \in T$. The profit of a tour is defined as the sum of all its section controls $w_p := \sum_{s_i,t_i} w_{s_i,t_i}$. We introduce binary variables z_p , $p \in P$, to decide that a tour is chosen or not. Then the following IP solves the TPP:

$$\max \sum_{p \in P} w_p z_p \quad (1)$$

$$\sum_{p \in P_{f,j}} z_p \leq 1, \quad \forall (f, j) \in F \times J \quad (2)$$

$$\sum_{p \in P_s} z_p \geq \kappa_s, \quad \forall s \in S \quad (3)$$

$$\sum_{p \in P_{s,j,t}} z_p \leq 1 \quad \forall (s, j, t) \in S \times J \times T \quad (4)$$

$$z_p \in \{0, 1\}, \quad \forall p \in P. \quad (5)$$

In the objective function (1) the profit of the selected tours is maximized.

Constraints (2) guarantee that each group performs at most one tour per day. Constraints (3) require that at least κ_s paths, that traverse section s , are chosen in a feasible solution. The single control constraints (4) make sure that different groups do not control a section at the same time. Finally constraints (5) demand the path variables being binary.

The local control restriction also explains why it is still possible to generate all paths by a simple enumeration. Furthermore, if $\Delta = 4$, the number of tour variables (which is normally between 15000 und 250000) is less than the number of roster sequence variables, which will be presented in the next section. We prefer the path based formulation since it allows to define the profit of a control tour independently of Δ and the length of the tour does not have to be modeled via constraints. Both would be the case if we use a formulation based on arcs variables.

3.2 Duty Roster Planning for Inspectors

The second task in the TEP is the planning of the rosters, called the *Inspector Rostering Problem (IRP)*. There, the objective is to penalize unsuitable sequences of duties. We will discuss later what this means exactly. We formulate the IRP again as a Multi-Commodity flow problem in a directed graph $\tilde{D} = (\tilde{V} = (\hat{V} \cup \{s, t\}), \tilde{A})$ with two artificial start and end nodes s, t . The nodes $\hat{v} \in \hat{V}$ represent duties as a pair of day and time period. The arcs $(u, \tilde{v}) \in \tilde{A} \subseteq \tilde{V} \times \tilde{V}$ model a feasible sequence of two duties according to legal rules. Hence, we call the arcs in the model *Duty Sequence Arcs*. By \tilde{A}_m we name all arcs representing duty arcs of inspector $m \in M$. We call the graph \tilde{D} *Inspector Roster Graph*. Therefore for each inspector its roster corresponds to exactly one s - t path in \tilde{D} , called *roster path*.

Figure 1 shows the roster graph of an inspector for one week. Three different time periods (E[arly], D[ay] and L[ate]) for the duties can be chosen and the other nodes indicate days-off or holidays. The solid thick path, starting from s to Monday 12am, then to Tuesday 8am, traversing to the same start times during the next days until Friday and finally two days-off at the weekend, exemplary shows a potential roster path for this inspector. The traffic distribution chart in the background indicates that in the integrated formulation all duties correspond to tours on sections that were planned according to the traffic distribution, as explained in the previous Section 3.1.

This problem is modeled by an IP, but based on arc variables. For an arc $(u, \tilde{v}) \in \tilde{A}$ define the costs $c_{u, \tilde{v}}$ of using this arc in a roster. The costs

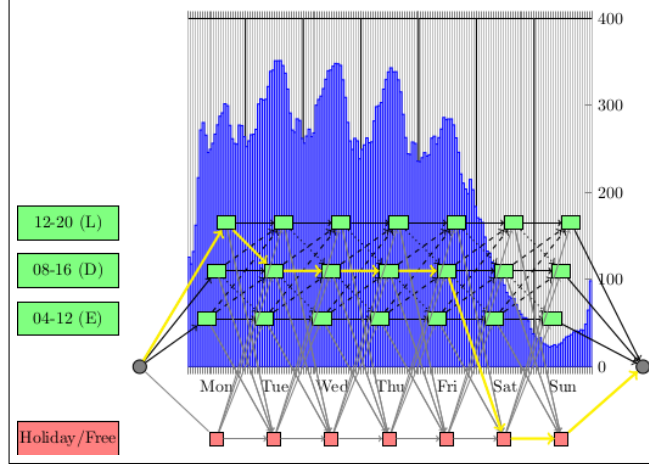


Figure 1: Excerpt of an Inspector Roster Graph of one inspector during one week. The solid thick path gives an exemplary roster. Three time periods for duties are allowed. In the background the traffic volume of one section during a week is shown. The traffic intensity is indicated by the numbers on the right y-axis.

represent penalties on the duty sequence arcs. Let further R be a matrix where each entry $r_{i,j}$ indicates the consumption of a resource i by arc j . A vector of upper bounds on resource consumption is given by r . The resource constraints used will be described in detail in Section 4.1. Then a variable $x_{u,v}^m$ is introduced for each arc $(u,v) \in \tilde{A}$ and inspector m . This leads to the following IP formulation:

$$\min \sum_{m \in M} \sum_{(u,v) \in \tilde{A}} c_{u,v} x_{u,v}^m \quad (6)$$

$$\sum_v x_{s,v}^m = 1, \quad \forall m \in M, \quad (7)$$

$$\sum_k x_{v,k}^m - \sum_u x_{u,v}^m = 0, \quad \forall v \in \hat{V}, m \in M, \quad (8)$$

$$Rx \leq r, \quad (9)$$

$$x_{u,v}^m \in \{0, 1\}, \quad \forall (u,v) \in \tilde{A}, m \in M. \quad (10)$$

The objective function (6) minimizes the costs of the rosters. The next constraints (7) and (8) represent the flow value and the flow conservation for the inspectors. Constraints (9) represent resource consumption of duties or sequence arcs, that can not be modeled locally in the graph. And last but not least the integrality constraints for the flow variables (10) are given.

Finally, the coupling constraints are presented. To this purpose, we denote by $P_{f,u}$ the set of all paths feasible for group f and duty $u \in \hat{V}$. The parameter n_f gives the number of inspectors in group f and by $m \in f$ it is meant, that the inspector m belongs to group f (which is a fixed assignment). This leads to the following constraint.

$$\sum_{p \in P_{f,u}} n_f z_p - \sum_{m \in f} \sum_v x_{u,v}^m = 0 \quad \forall f \in F, u \in \hat{V}$$

In the next Section 4 we will describe how some exemplary chosen requirements on crew rostering for toll inspectors can be modeled. Local modeling in the graph is one key issue of our approach, since it helps to reduce the number of (resource) constraints in the IP. In case where a local modeling is not possible we will describe the corresponding resource constraints.

4 How to model duty sequencing rules

First, we consider requirements that can locally be modeled in the roster graph. Our first requirement to discuss is that duties must not start at arbitrary points of time during the day. Therefore, we define a *Duty Type* by its start and end time and only duties corresponding to one of the pre-defined duty types may be scheduled. We remind the reader from Section 3.1 that all control duties are uniform according to their length and therefore one can classify them according to their start (and end) time. Duty types can easily be modeled since each potential duty corresponds to a node in the roster graph. We explained above that a node is defined by a specific day and a time period. Hence, only nodes whose time period corresponds to a feasible duty type are generated.

In classical planning approaches, like in public transport Weider (2007), duty types are part of duty scheduling. There duties can be generated independently for each duty type in the pricing step. Then only tasks that

are usable by the current duty type are present in the pricing network. Afterwards, in duty rostering the beginning and end of a duty is not changed anymore.

The second issue is how to take annual leave days or weeks into account. Suppose an inspector m has a leave day on a certain day, e.g., on a Wednesday. At this day inspector m must not be assigned to a duty. To this purpose for the respective day a “holiday”-node is generated. All arcs of \tilde{A}_m , which start or end at a node belonging to this Wednesday, are incident to this node. Hence, it is not possible to define a path in \tilde{D} with any other than the holiday node on this Wednesday. Further in a preprocessing step holiday nodes can be contracted to better model the sequence between the last duty before the holiday and the following duty after the holiday. Pre-assigned duties, e.g., staff meetings, can be modeled analogously.

A very important requirement for a duty roster is to respect minimum rest times between two subsequent duties. The German *Working Hours Act* (*Arbeitszeitgesetz*) lays down 11 hours as minimum rest times. This is a local decision in our model since there is a conflict between two duties if the precedent duty ends less than 11 hours earlier than the second starts. A graph based approach, as it is used here, is very suitable to resolve this conflict. It can simply be modeled there by only setting arcs between two duty nodes, if the head node duty starts at the earliest 11 hours after the tail node duty has ended. See for example the early duty on Tuesday 4am-12am in Figure 1. There is an arc from the early duty on Monday to this duty since the rest time equals 16 hours. But there is no arc between the late duty on Monday 12am-20pm and the early duty on Tuesday, since the rest time is only eight hours. According to this modeling of rest times it is a simple observation that \tilde{D} is acyclic and almost no arcs exist between duty nodes belonging to the same day.

This modeling has the advantage that we do not need any rest time constraints in our model. It is the main algorithmic aspect and contribution in this paper that we try to model as many constraints as possible as local decisions in our graph model. This is a key issue to reduce the high complexity of the integrated problem to get a final optimization problem to solve that is as small as possible. On the one hand this modeling technique is commonly used in network flow approaches. But our goal is not to introduce this technique itself. On the contrary, we want to show that it supports a strong graph formulation, reducing the IP complexity. It leads to a compact integrated formulation for a large-scale optimization problem that combines

two different planning steps.

A similar regulation is valid for the rest time when there are days off in between. In case of a time-off there should be at least two days off to ensure a sufficient rest time for the inspectors. This is also modeled by duty sequence arcs. Therefore, a one-day-duty-off would be represented by sequence arcs between nodes on day j and nodes on day $j + 2$. Then a two-days-off is accordingly modeled by arcs between nodes of day j and day $j + 3$. To prevent “short weekends”, i.e., only one day-off, no arcs between days j and $j + 2$ are generated. As a consequence if an inspector has a duty on day j , then his next duty cannot be on day $j + 2$.

Next, we show how to model (employee) preferences that do not correspond to statutory rules but should be considered by the model as best as possible. These rules are motivated by the human biorhythms and play a crucial role in rostering due to the monthly time horizon. In particular in planning of safety relevant tasks, e.g., in the airline and railway industry, the fatigue levels depending on the human biorhythms of the personnel must be incorporated. A common approach is to handle this by the objective function.

In our application, the preference criterion relates to the sequence of duties, namely to changes of the duty starting time on two subsequent days. This is called a *rotation*. If a duty on Thursday starts later than on Wednesday (the day before), this is an example of a *forward rotation*. According to this definition the case that a duty on a subsequent day starts earlier than on the day before is called *backward rotation*. Backward rotations can only occur, if they do not violate the minimum rest time between the corresponding duties. The solid thick path in Figure 1 contains a backward rotation, since after the Monday 12am duty the following duty on Tuesday yet starts at 8am. Even though rotations are legally feasible, they should be avoided. It is particularly known for backward rotations that they alter the human biorhythms and affect the sleep Knauth and Hornberger (2003). Hence, we are able to tackle rotation costs directly by definition of the arcs.

4.1 Resource constraints

In the following, two important types of resource constraints, used in our model, are presented. First, the observance of working hours of the inspectors is discussed. It is very important that their average working time is approximately kept. At the beginning of the planning horizon the current

account value a_m of each inspector $m \in M$ is given. At the end of the planning horizon a feasible interval for the working time account is given with bounds ℓ_m and o_m . Each duty u on the roster path of m consumes some working time ω_u and therefore the constraints bound the length of the roster paths. This leads to the following constraints of the resource “working time”.

$$\sum_{(u,v) \in \tilde{A}_m} \omega_u x_{u,v}^m + a_m \leq o_m, \forall m \in M, \quad (11)$$

$$\sum_{(u,v) \in \tilde{A}_m} \omega_u x_{u,v}^m + a_m \geq \ell_m, \forall m \in M. \quad (12)$$

Kohl and Karisch (2004) call this kind of rule a *horizontal rule* according to the fact, that in many rotas the roster of an inspector is displayed in a single row and only one roster is involved by this rule. The working time-limits ℓ and o are not the same for all inspectors since part-time workers, regular employees and officials have different daily working times.

Second, we describe a new type of resource constraints in our model. We will also analyse the effect of these constraints in the computational results presented afterwards. It is about *unsocial working hours*. This term is hard to define, but in this paper we use the following definition: A duty consumes resources of unsocial working hours, if it includes times in the late evening, during the night, at the weekend or at bank holidays. For health reasons and to ensure the reconciliation of work and family life, the number of unsocial hours to work is restricted. For inspector m let γ_m be the maximum number of unsocial working hours in the current planning horizon. Each duty u has a unsocial work part p_u with $0 \leq p_u \leq \omega_u$. This leads to the following resource constraint for unsocial working hours.

$$\sum_{(u,v) \in \tilde{A}_m} p_u x_{u,v}^m \leq \gamma_m, \forall m \in M. \quad (13)$$

We indicate that those constraints do also involve only one roster and therefore correspond to a horizontal rule.

In contrast to horizontal rules Kohl and Karisch name constraints as *vertical rules* if they affect several rosters. An example for a vertical rule, that

could be used in the TEP, are so-called *Duty Mix*-Constraints. They guarantee a pre-defined mix of the duty types providing lower and upper bounds on how many of all duties may belong to the current type. An important purpose of them is to give upper bounds on certain duty types, e.g., overnight duties. But since resource constraints were used to bound unsocial working hours, duty mix constraints became obsolete in our model. Therefore, they are not used in our computations presented later in this report. Another work rule used in the TEP is that inspectors must have at maximum six working days in a row. This is simply modeled by constraints prohibiting sequences of duties on more than six days without a duty-off (or vacations) in between. They could be seen as *rolling* resource constraints since they involve only a subset of a roster, i.e., for each set of seven consecutive days one constraint is used. Thus, in case of n days, $n - 6$ additional resource constraints are defined involving days j until $j + 6$ for $j \in \{1, 2, \dots, n - 6\}$ with $n > 6$. Requirements representing other modeling techniques than the presented ones are currently not used in the TEP.

Before we want to look in more detail at the two different objective functions, the size of the roster graph is shortly discussed. The number of nodes depends on the length of the planning horizon and on the number of duty types. The number of arcs then clearly depends on the number of nodes, since the graph is connected and also on the number of inspectors. Especially in case of increasing duty types there is a significant increase in the number of arcs. Furthermore it decreases if there are a lot of fixed duties or holidays during the planning horizon. In Section 7 some numerical examples will be presented.

5 A sequential solving approach

In this Section problems are discussed that arise when a straight-forward sequential solving approach for the TEP is used. This will prove the benefit and need of using an integrated model. In a sequential approach we first solve a tour planning problem independently of the crew assignment and the rosters are computed in a subsequent step.

To that end, in a first step the tour planning problem (1)-(5) is solved for each day of the planning horizon. We assume that tours can be arbitrary distributed along the sections of the network, since a tour is not assigned anymore to a certain crew that may only conduct controls on parts of the

Duty Type	Sections
06:00	8-8
06:00	9-7
06:00	1-4
06:00	3-3
06:00	2-1
06:00	4-2
08:00	16-9
09:00	9-4
13:00	8-8
13:00	2-2
13:00	1-3
13:00	3-1
20:00	2-2

Table 1: Tour suggestions for the first day of the planning period, a Wednesday

Group	Tours
1	none (vacation)
2	1-4,3-3,2-1,4-2,2-2,1-3,3-1,2-2
3	none (vacation)
4	3-3,1-3,3-1
5	1-4,3-3,2-1,4-2,2-2,1-3,3-1,2-2
6	
7	none (vacation)
8	8-8,9-7,3-3,16-9,9-4,8-8
9	none (vacation)
10	9-7,16-9
11	
12	1-4,3-3,2-1,4-2,2-2,1-3,3-1,2-2
13	8-8,8-8

Table 2: List of feasible tours for each control group on the first day, a Wednesday

current control area. Therefore, constraints (2) are replaced as follows. Let P_j be the set of paths starting on day $j \in J$ and γ the number of control groups in the current control area. The following constraints are part of the pure tour planning model:

$$\sum_{p \in P_j} z_p \leq \gamma, \forall j \in J.$$

Since each group executes at most one tour per day γ is a reasonable upper bound on the number of tours during a day. To test the sequential approach we used an exemplary instance from a control area, called r_2 , with discretization $\Delta = 4h$. There are 24 sections (numbered from 1 to 24) to control, 13 groups and 7 duty types. The pure tour planning approach could be solved within seconds and next some results are discussed in detail.

Table 1 lists the tours suggested for the first day, a Wednesday. Before the rosters are generated the tours must be assigned to control groups. Table 2 shows for each group the set of feasible tours on this day. It is obvious that no tour can be performed by groups 6 and 11. Then, one can either give a day-off to those groups or go back to the tour planning part and compute the best profit tours for groups 6 and 11. There only tours feasible for one of both groups are considered. Assigning a day-off is inappropriate since either

a single day-off is not desired or in case of several days where this group can not be allocated the working time consumption is too low. But an individual tour generation for a single team is in fact a kind of integration of tour planning and rostering or a tour planning approach where crew availability is a very important issue.

Next, a subset of the tours is distributed to the groups depending on crew feasibility and profit. But this team assignment can not be done independently for each day to prevent infeasible rosters. Assume for example group 8 is assigned on Wednesday to tour 8-8 starting at 1pm, but on Thursday to tour 9-7 starting at 6am. This is infeasible since the minimum rest time equals only eight hours. Hence, the final decision on the tours must be taken in the rostering optimization. But the resulting model is then quite close to our integrated model described in Section 3. We summarise that a sequential approach would either lead to a lot of re-computation steps of several tours to prevent infeasible duty rosters and would lose sight of the optimization goal. The alternative would be to use a model with a lot of integration aspects similar to the TEP.

6 Analysing the objective function

It is obvious that the TEP has two objectives: on the one hand the rewards of the sections to control and on the other hand the costs for the rotations. In our last publication Borndörfer et al (2012b) the focus was on the influence of rotation penalties on the number of rotations in an optimal solution. It was a very limited view on the problem since we only looked at two different not exactly specified levels of the rotation penalties and concluded that an increasing rotation factor leads to solutions with less rotations.

Therefore, we like to have a more sufficient and detailed look on the two different objectives. With two objectives, the TEP relates to a *bi-criteria optimization problem*, a problem in the important field of *Multi-criteria Optimization*, see Ehrgott (2005) for an overview. We analyse the interplay of the objectives with the well-known concept of *Pareto optimality* Ralphs et al (2006). First, we shortly recall the definition. Let $\max_{x \in X}(f_1(x), f_2(x))$ be a bi-objective problem. A feasible solution \hat{x} is called *Pareto optimal*, if there is no other solution x such that $f_i(x) \geq f_i(\hat{x})$ for $i = 1, 2$ and $f_i(x) > f_i(\hat{x})$ for at least one $i \in \{1, 2\}$. A feasible solution \hat{x} is called *weakly Pareto optimal*, if either $f_1(\hat{x}) \geq f_1(x)$ or $f_2(\hat{x}) \geq f_2(x)$ holds for all other solutions $x \in X$.

In this section, our main contribution is the analysis of parts of the Pareto frontier of one exemplary chosen problem instance to get an indicator for the Pareto frontier of TEP instances. To this purpose the bi-objective problem is transformed into a single objective optimization problem by the standard weighted sum approach Ralphs et al (2006). Therefore, we introduce a parameter $\alpha \in [0, 1]$ and combine the objectives (1) and (6) to:

$$\max (1 - \alpha) \sum_{p \in P} w_p z_p - \alpha \sum_{m \in M} \sum_{(u,v) \in \tilde{A}} c_{(u,v)} x_{(uv)}^m \quad (14)$$

The theoretical benefit of using the weighted-sum approach is well-known Ralphs et al (2006). But for the sake of completeness we repeat in a proposition a short proof of this result applied to the TEP.

Proposition 1 (Weak Pareto-optimal solutions). *We are given a TEP instance in the integrated IP formulation, but with the weighted objective (14) depending on the parameter α . We call this problem $TEP(\alpha)$. Then $\forall \alpha \in [0, 1]$ an optimal solution (x^*, z^*) of the $TEP(\alpha)$ is weak Pareto-optimal.*

Proof. Let (x^*, z^*) be an optimal solution of the $TEP(\alpha)$. Assume this solution is not weak Pareto-optimal. Hence, there exists another solution (\tilde{x}, \tilde{z}) with both $\sum_{p \in P} w_p \tilde{z}_p > \sum_{p \in P} w_p z_p^*$ and

$$\sum_{m \in M} \sum_{(u,v) \in \tilde{A}} c_{(u,v)} \tilde{x}_{(uv)}^m > \sum_{m \in M} \sum_{(u,v) \in \tilde{A}} c_{(u,v)} (x^*)_{(uv)}^m.$$

There, we replaced the subtraction in the objective by a sum by setting $c_{(u,v)} < 0, \forall (u,v) \in \tilde{A}$. Then it follows that

$$\begin{aligned} (1 - \alpha) \sum_{p \in P} w_p z_p^* + \alpha \sum_{m \in M} \sum_{(u,v) \in \tilde{A}} c_{(u,v)} (x^*)_{(uv)}^m &< \\ (1 - \alpha) \sum_{p \in P} w_p \tilde{z}_p + \alpha \sum_{m \in M} \sum_{(u,v) \in \tilde{A}} c_{(u,v)} \tilde{x}_{(uv)}^m. \end{aligned}$$

But this is a contradiction to the condition that (x^*, z^*) is an optimal solution of the $TEP(\alpha)$ □ □

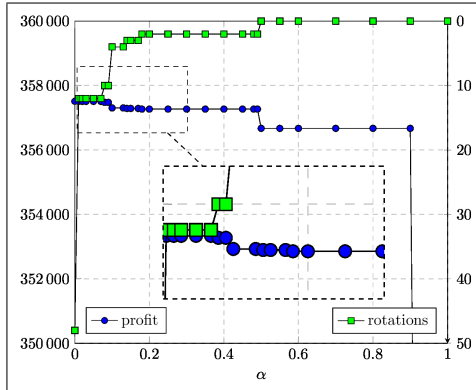


Figure 2: Subset of the Pareto frontier of instance **r1-test**, on the x-axis weight factors α and on the y-axis the control profit on the left and the number of rotations on the right in reverse direction

Using the same reasoning, it is easy to show that every solution of $\text{TEP}(\alpha)$ is strongly Pareto-optimal $\forall \alpha \in (0, 1)$.

For the test we choose an instance **r1-test** (corresponding to a German control area called r_1). This instance involves 21 inspectors, 17 control sections and 6 duty types. The resulting MIP (before presolve) has then 7738 constraints, 96526 variables and 1233369 non-zeros. We varied the weight parameter α for several values including the extremal ones. The tests were done on a Dell Power Edge M620 computer with an 8-core Intel Xeon CPU of 2,70 GHz using Cplex 12.5 with 8 threads and default parameter settings. Since this instance was quite easy, each run with a different value of α was solved to optimality within two hours.

In Figure 2 one can see the results of our computations. The x-axis corresponds to different values of α , while on the left y-axis the control profit and on the right axis the number of rotations in reverse direction is shown. In the case of no rotation penalties, i.e., $\alpha = 0$, there are quite a lot rotations (48). But even for a very small value of α , e.g., 0.02, the number of rotations is rapidly reduced to a total number of 12, while the profit value remains almost the same. An explanation for this observation is that there is symmetry in the problem which allows for changes of duties without losing control quality. Since each section can be controlled by more than one team, in some cases a simple permutation of the tours of two teams can resolve a rotation without changing the profit value. As an example, consider two groups t_1 and t_2

that both can control on sections s_1 and s_2 . We are given a plan with two rotations, for t_1 on Monday at 7am on section s_1 and on Tuesday at 10am again s_1 . For t_2 on Monday at 10am and on Tuesday at 7am on s_2 on both days. The schedule on Tuesday is now changed as follows: group t_1 controls from 7am s_2 and group t_2 from 10am s_1 . Then the profit remains the same, but two rotations are resolved.

A value $\alpha > 0.1$ again reduces the number of rotations until $\alpha = 0.5$, where the number of rotations equals zero. Indeed the profit value is decreasing when the weight factor for the rotation penalties increases but the loss is very small. It is approximately as high as the profit of a tour on low-traffic sections. Another interesting case is $\alpha = 1$, where no profit value is considered for the toll enforcement. The resulting profit value in the solution, i.e., the value that is assigned to nodes on solution paths in the tour planning network, can be arbitrary bad, since there is no incentive for the model to control sections with high traffic more than those with low traffic. In this case it was only 238963.

The restriction on the weak Pareto-optimality property can also be observed with our example. Namely, in the case of $\alpha = 0$ we discussed that a simple permutation of duties between two teams can reduce the number of rotations without changing the control profit. If we denote our optimal solution again by x^* and the permuted by \tilde{x} , it holds that $f_1(x^*) = \sum_{p \in P} w_p z_p^* = \sum_{p \in P} w_p \tilde{x}_p = f_1(\tilde{x})$ but $f_2(x^*) < f_2(\tilde{x})$. The same holds for the case of $\alpha = 1$ by exchanging f_1 and f_2 in the equations of the previous sentence. This violates the condition of (strong) Pareto-optimality.

The results of the test indicate our assumption in the previous publications that a rotation free plan can be achieved without a big loss of the profit. Hence, finally we tried to verify that this result was not an exceptional case. For this purpose, four other real world instances from control area r_1 were tested with regard to the influence of rotation penalties on the control reward. The control profit value of two parameter settings was compared. First, a setting without rotation penalties was used, i.e., $\alpha = 0$. In this setting any number of rotations can be part of the solution. The only focus lies on the control rewards. Second, we used a setting where the penalties lead to an optimal solution without any rotations, e.g., $\alpha = 0.6$ in our previous example. The solutions in the second setting with no rotations never had a loss of more than 0.8% of the control profit achieved by the first setting.

We conclude this section with a short comment on the strength of a mathematical optimization approach in case of more than one objective. A bi-criterion objective function allows to use different parameter settings and focuses on objectives to execute several optimization runs for a planning scenario. Then the planner can compare the different solutions and choose one as the best. In manual planning a comparison of alternatives is not possible since it takes a long time to generate even one plan.

7 Results from production operation in Germany

We implemented our model and algorithm in a computer program, called TC-OPT, which has an interface to the commercial planning suite IVU.plan since release 11.2 from the IVU Traffic Technologies AG. In the following we will analyse the performance of TC-OPT with respect to various aspects on instances from different control areas. In all tests, scenarios from the test environment of BAG or from real-world productive operations are used.

The computational results are divided into three parts, each of them with a different scope. The first two parts are parameter tests. In the first one the computational complexity of the basic model depending on different input sizes is shown. The second one compares varying settings of the new feature regarding bounds on unsocial working hours. The third one includes more regions and more duty types than the tests published so far, demonstrating that the TEP and our solution approach is able to handle these extensions.

First, we give some basic settings we used for all our computations. As an IP solver Cplex 12.6 by IBM with the default parameter setting was applied by using up to eight threads. All computations were done on a Dell Power Edge M620 workstation with an 8-core Intel Xeon CPU with 2,70 GHz and Ubuntu 14.04 as operating system. The memory limit for the solution tree was 40 GB. Furthermore, there was a time limit of 6 hours (= 21600 seconds) for each instance. All instances are based on a time discretization of $\Delta = 4h$. All presented times will be in seconds.

Instance	Region	Inspectors	Sections	Fixed Duties	Duty Types	Nodes	Arcs
I1	r_2	23	24	0	7	219	452811
I2	r_2	23	24	106	7	219	331151
I3	r_2	23	24	299	7	219	144874
I4	r_1	22	22	238	4	126	60581
I5	r_1	22	22	238	6	188	134179
I6	r_1	22	22	238	8	250	236092
I7	r_1	22	22	238	12	374	523286
I8	r_1	22	22	238	16	498	923751

Table 3: Key characteristics of test data for varying duty types and fixed duties.

7.1 Correlation between the tractability of the TEP and the size of the roster graph

At the end of Section 4.1 we discussed the basic values that have a major influence on the size of the roster graph. In the following we will examine how different basic values lead to different graph sizes and if they have an influence on the tractability of the TEP. To that end, we study two scenarios from different control areas r_1 and r_2 .

Table 3 includes the description of the test set where both scenarios are tested with three or five different parameter values. First, we describe the columns indicating the parameter setting of the instances. The first two columns give the name of the instance and corresponding region. Instances I1, I2 und I3 belong to the first scenario and the others to the second. The number of inspectors is in the third column while the fourth one presents the number of sections. The fifth column gives the number of fixed duties according to holidays, duty-off requests or other fixed duties, e.g., to conduct stationary controls or staff meetings. The sixth column “duty types” gives the number of duty types allowed. The last two columns give the number of nodes and arcs in the roster graph \tilde{D} .

As stated earlier, the number of arcs indeed decreases when more fixed assignments are recorded. Obviously, the number of arcs significantly increases when more duty types are considered. The number of roster arcs for the instance with 16 duty types is more than 15 times as high as the number of arcs of the instance with only four duty types.

In Table 4 the IP solution analysis is presented. Columns two and three give the number of constraints and variables of the IP. The fourth one gives

Instance	Rows	Columns	Time (LP)	Time 1st sol	Gap (%)	Time (sec.)
I1	16144	606664	206.90	1000	2.02	21600.00
I2	15335	469849	108.53	3600	3.82	21600.00
I3	13779	249062	35.07	150	1.43	21600.00
I4	8855	117917	7.96	80	-	2418.32
I5	11231	220183	20.99	140	0.38	21600.00
I6	13607	350764	24.17	1070	0.66	21600.00
I7	18965	695294	56.76	1550	1.57	21600.00
I8	23753	1153095	212.20	7275	2.39	21600.00

Table 4: Results of performance tests for varying duty types and fixed duties.

the solution time for the linear root relaxation and the fifth the time when the first integer was found. The following gives the integrality gap in percent and the last the overall solution time.

The first observation is that with our modeling techniques we are indeed able to limit the number of constraints such that never more than 24000 are present in the model. Furthermore, one could observe a proportional ratio between the number of roster arcs (and therefore also between the number of variables) and the solution time of the linear root relaxation, i.e., the time to solve the root LP increases with a higher amount of roster arcs. In case of the scenario with the different number of duty types this holds also for the time of the first solution and for the final solution gap. Only one instance, I4, could be solved to optimality. But even for the huge instance with 16 duty types and almost one million roster arcs, I8, the final gap is not more than 4%. Hence, in each case a feasible solution with a proven high quality can be found. We can conclude that both the number of fixed assignments and the number of duty types have a major influence of the complexity of TEP instances.

7.2 The influence of resource constraints

In this section we study to which extent the resource constraints, that restrict the number of unsocial working hours, influence the solution behaviour of the TEP. Hence, we consider again a scenario from region r_1 , but different from the one used in the previous section, and a second from another region r_3 . For each scenario we tested three different variants of considering unsocial working hours, called case *A*, *B* and *C*. Case *A* represents a setting where no

Instance	Region	Inspectors	Sections	Fixed Duties	Case	Duty Types	Roster Arcs
I9	r_3	17	11	82	A	11	631626
I10	r_3	17	11	82	B	11	631626
I11	r_3	17	11	82	C	7	261956
I12	r_1	22	22	191	A	12	657334
I13	r_1	22	22	191	B	12	657334
I14	r_1	22	22	191	C	8	299014

Table 5: Key characteristics of instances with constraints for different unsocial work hour settings.

limitations on unsocial work hours exist. Case B corresponds to the resource constraints introduced in Section 4.1. The last case C represents the setting that unsocial work hours are not allowed at all. Hence, late, night or weekend duties must not be planned. At first glance, this setting seems to be unreal, but imagine the case that duties covering unsocial hours could be planned on the basis of individual duty requests of the inspectors. Then they are fixed duties and will not be changed by an algorithm.

Table 5 gives all important data concerning the test of the unsocial working hours. The columns start with the name of the instance, the region and the number of inspectors and sections. One can observe, that in region r_3 the number of inspectors and sections is less than in region r_1 and r_2 . The fifth column gives the number of fixed duties. Column six indicates the different settings for the unsocial hours. The seventh column presents the number of duty types and the last the corresponding number of roster arcs. As shown in the previous section a higher number of duty types also leads to a quite large number of roster arcs (> 500000 arcs). But in case C there is a huge decrease in the number of arcs. The reason is that duty types covering unsocial work hours can be neglected.

The performance analysis of the second test is given in Table 6. We refer the reader to Section 7.1 for a description of the headers. Comparing cases A and B one can see that the B instances are slightly easier to solve, since the root lp can be solved a little faster and the final gap is smaller. An explanation of this behaviour might be that inspectors have different upper limits for unsocial hours in a month, depending on their unsocial work duties done in the past. Then a few groups may have an upper bound close or equal to zero. For those inspectors, sequence arcs connecting duty types consisting

Instance	Rows	Columns	Time (LP)	Time 1st sol	Gap (%)	Time (sec.)
I9	14305	761305	193.78	9600	1.63	21600.00
I10	14339	761305	153.04	5850	1.14	21600.00
I11	9471	344479	30.42	70	-	252.05
I12	17258	897022	199.04	900	1.97	21600.00
I13	17302	897022	137.37	6380	1.35	21600.00
I14	12202	458806	37.32	200	-	355.03

Table 6: Results of performance tests of resource constraints for unsocial work hours.

Instance	Region	Inspectors	Sections	Fixed Duties	Duty Types	Roster Arcs
I3	r_2	23	24	299	7	144874
I4	r_1	22	22	238	4	60581
I10	r_3	17	11	82	11	631626
I15	r_2	23	24	189	8	268997
I16	r_4	21	14	267	5	86991
I17	r_2	23	28	539	10	158326
I13	r_1	22	22	191	12	657334

Table 7: Key characteristics of instances for different regions.

of many unsocial hours are easily cut off by the IP solver, similar to case C .

Omitting duties covering unsocial working hours like in case C does not only lead to a smaller graph size and less variables, but also makes the instances easy to solve. For both scenarios an optimal solution can be found within six minutes.

7.3 Results from the Roll-out in Germany

Last but not least we tested the general performance of the TEP with seven instances from four different control areas (regions). Those instances directly originate from pilot tests and production operation at the BAG.

Table 7 gives the input data for all instances. We refer to Section 7.2 for a detailed description of the column names. It is apparent that the number of sections and duty types vary a lot among different regions and scenarios. Hence, also the number of roster arcs has a broad variation ranging from 60000 for region r_1 in instance I4 to more than 650000 for the same region in I13. For three instances (I10, I13, I17) resource constraints for unsocial

Instance	Rows	Columns	Time (LP)	Lower Bound Gap (%)	Time 1st sol	Gap (%)	Time (sec.)
I3	13779	249062	35.07	7.48	150	1.43	21600.00
I4	8855	117917	7.96	11.77	80	-	2418.32
I10	14339	761305	153.04	2.04	5850	1.14	21600.00
I15	15401	402285	77.82	9.34	500	0.63	21600.00
I16	8068	101941	9.53	5.13	100	-	204.35
I17	17871	352396	24.46	5.08	175	-	939.59
I13	17302	897022	128.12	4.05	6200	1.36	21600.00

Table 8: Results of performance tests for different regions.

working hours were part of the model. As in Section 7.2, instances I10 and I13 use case B and therefore the regular setting of unsocial working hours. Furthermore, we remark that instances I3 and I4 were first introduced in Section 7.1.

The analysis of the computational performance of the third test is shown in Table 8. For the explanation of the column names we again refer to the previous tables. We only mention the new entry in column five, “Lower Bound Gap (LB)”. This corresponds to the linear root relaxation before adding cuts, which is used as an initial lower bound, i.e., as dual bound, in our algorithm. To define this value we introduce the optimum of the root relaxation by “ $\text{lp}(\text{opt})$ ” and let `best_int` the best found integer solution. Then we define the Lower Bound Gap by $\text{LB} := \frac{\text{lp}(\text{opt}) - \text{best_int}}{\text{best_int}}$. As expected from the different number of roster arcs (see Table 7), there is also a broad variation in the number of variables, up to almost 900000. The solution time of the linear relaxation in the root node (without cuts) ranges from 8 seconds up to 3 minutes. We state that feasible solutions were found for all instances with a final integrality gap that never exceeds 1.5%.

One reason for this encouraging result is the good quality obtained by the initial lower bound. The optimality gap w.r.t. LB is always less than 12%. Especially the two huge instances I10 and I13 benefit from a lower bound that is quite close to the best primal bound. A first feasible solution is computed for almost all instances within 10 minutes, except for instances I10 and I13 that require more than one hour. Three instances could be solved to optimality. The longest of them, I4, took around 40 minutes, while I16 was solved optimally in less than 4 minutes. We note that the instances solved to optimality correspond to those having less variables and hence a faster

LP-solution time. In addition for all of those feasible solutions were found within three minutes. We again see the influence of fixed assignments on the computational performance since for all optimal solved instances more than 200 duties were fixed.

This paragraph discusses some additional findings of our tests and presents some further parameter tests. As shown in Section 7.1, we were able to solve an instance with 16 duty types with only a small integrality gap. This does not indicate that we can solve all scenarios with 16 duty types during six hours. In fact, using only one additional duty type for instance I10, results in finding no solution within six hours. Another observation is that all instances solved to optimality in our three different tests have less than 500000 variables. We also tried to solve two instances with a time discretization of two hours. For instance I1 no solutions were found within six hours of computation time but I4 could be solved to optimality within four hours. This is in fact not relevant for the current real-world operation but could be an issue for future research. In addition a test run covering two control areas together was executed. There the number of inspectors was 39 and the number of sections 31. It was also solved within six hours but with a slightly higher gap of 7.6%. Finally, the results indicate that the number of sections is not decisive for the computational complexity (in contrast to duty types and fixed duties). The instance I17 has more than twice the number of sections compared to I10, but it is much easier to solve.

In Borndörfer et al (2012b) a proof of concept with results for test computations of one region was provided. After these successful pilot tests the use of our algorithm was extended in 2013 to additional regions as part of an extensive Roll-Out project. Since the beginning of 2014 all control areas in Germany are planned by using mathematical optimization. Hence, we can proudly conclude that our integer programming approach gained acceptance in practice. This is the first time that toll enforcement tours and duties are planned by an integrated optimization approach. Our approach supports the planners to achieve a better quality of control plans by using state of the art mathematical optimization techniques. This makes their daily work easier, more transparent, more objective, and most of all more efficient.

8 Conclusion and Future Research

In this paper we extended our previous work on the Toll Enforcement Problem (TEP), a special type of an integrated Vehicle Routing Problem, into several directions. The TEP is used to compute optimal tours and inspector rosters for the truck toll on German motorways. First a classification of the TEP into the field of routing and rostering problems was presented. Then we shortly presented an IP formulation for the tour planning and a graph model for the rostering of the crews which provides new insight in the studied problem. In our opinion, the models are sufficiently general to deal also with other inspection problems, where inspection tours are conducted. The rostering problem corresponds to a multi-commodity flow problem formulated as an IP with additional resource constraints.

One main focus of this report was the description of typical requirements and legal rules for the crew rostering. In many cases these conditions can be satisfied according to an appropriate local modeling in the roster graph. Therefore no constraints were needed in the IP formulation to guarantee compliance of those rules. Furthermore, the limitations of a sequential approach were presented using a small exemplary computation. Another aspect was an extended analysis of the bi-criterion character of our problem. We analysed a set of these solutions located on the Pareto-frontier. Most of the rotations could be eliminated according to some symmetry in the problem structure. Therefore even a very small penalty value leads to a significant decrease of rotations. We were able to provide solutions without rotation costs at the expense of a hardly smaller control quality.

Finally computational results from real world instances were presented. We showed in detail the influence of different input data, like fixed duties, duty types or bounds on unsocial work hours, on the problem complexity and the solution behaviour. Instances from several control areas representing different problem sizes were analysed. The results indicate that our graph-theoretic model and algorithmic approach is indeed able to solve the instances quite fast and achieves a very good solution quality. Practical instances can still be solved by modern state-of-the-art solvers. Therefore, we can conclude that with our model and in particular with the approach of omitting infeasible duty sequences during the construction of the roster graph it is possible to tackle this challenging real-world problem.

An outline for future research could be three main aspects: An important issue is to increase the simultaneous number of duty types used during one

computation. A second research direction is to further decrease the problem complexity, e.g., by a dynamic generation of control tours or duty rosters. In addition, advanced problem specific algorithmic approaches like heuristics, multi-level algorithms or branch and price should be developed. The third aspect is the generation of rosters based on a probability distribution of inspectors, which is the output of the game-theoretic approach Borndörfer et al (2013) on toll enforcement. Other research goals are the transfer of our model to other inspection applications and incorporation of additional features and requirements in the TEP.

Acknowledgments

We thank Doris Ludwig-Schreiber and Christian Hoffmann from the German Federal Agency for Goods Transport (BAG) for initiating the project on optimal toll enforcement with the Zuse Institute Berlin. Furthermore, we thank the technical project managers Eduardo Pinto and Thomas Dankert for organising the project, and in addition, for giving a lot of technical support according to the installation and operation of our tool at the BAG. Our sincere thanks go also to Daniel Schneider, Ralf Haas and Uta Sperling from the planning department of the BAG, who provided us with many test instances, gave us excellent feedback, and helped us to understand the manifold requirements from real operations. Furthermore, we thank Hans-Stefan Madlung and his colleagues from IVU Traffic Technologies AG for the joint development of a small and flexible interface to exchange the data between TC-OPT and the commercial planning tool “IVU.Plan”.

References

- Archetti C, Bianchessi N, Speranza M (2013a) Optimal Solutions for Routing Problems with Profits. *Discrete Applied Mathematics* 161(4-5):547 – 557, DOI <http://dx.doi.org/10.1016/j.dam.2011.12.021>, Seventh International Conference on Graphs and Optimization 2010
- Archetti C, Speranza M, Vigo D (2013b) Vehicle Routing Problems with Profits. Tech. rep., WPDEM2013/3, University of Brescia
- Borndörfer R, Omont B, Sagnol G, Swarat E (2012a) A Stackelberg Game

- to Optimize the Distribution of Controls in Transportation Networks. In: Krishnamurthy V, Zhao Q, Huang M, Wen Y (eds) *Game Theory for Networks*, Springer Berlin Heidelberg, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 105, pp 24–35, DOI 10.1007/978-3-642-35582-0_17
- Borndörfer R, Sagnol G, Swarat E (2012b) A Case Study on Optimizing Toll Enforcements on Motorways. In: Ravizza S, Holborn P (eds) *3rd Student Conference on Operational Research*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, OpenAccess Series in Informatics (OASICs), vol 22, pp 1–10, DOI <http://dx.doi.org/10.4230/OASICs.SCOR.2012.1>, URL <http://drops.dagstuhl.de/opus/volltexte/2012/3541>
- Borndörfer R, Sagnol G, Swarat E (2012c) An IP Approach to Toll Enforcement Optimization on German Motorways. In: Klatte D, Lüthi HJ, Schmedders K (eds) *Operations Research Proceedings 2011*, Springer Berlin Heidelberg, Operations Research Proceedings, pp 317–322, DOI 10.1007/978-3-642-29210-1_51
- Borndörfer R, Buwaya J, Sagnol G, Swarat E (2013) Optimizing Toll Enforcement in Transportation Networks: a Game-Theoretic Approach. *Electronic Notes in Discrete Mathematics* 41(0):253–260, DOI <http://dx.doi.org/10.1016/j.endm.2013.05.100>
- Boussier S, Feillet D, Gendreau M (2007) An Exact Algorithm for Team Orienteering Problems. *4OR* 5(3):211–230, DOI 10.1007/s10288-006-0009-1
- Burke EK, De Causmaecker P, Berghe GV, Van Landeghem H (2004) The State of the Art of Nurse Rostering. *J of Scheduling* 7(6):441–499, DOI 10.1023/B:JOSH.0000046076.75950.0b
- Cappanera P, Gallo G (2004) A Multicommodity Flow Approach to the Crew Rostering Problem. *Operations Research* 52(4):583–596
- Ehrgott M (2005) *Multicriteria Optimization*. Springer
- Ernst A, Jiang H, Krishnamoorthy M, Nott H, Sier D (2001) Rail Crew Scheduling and Rostering Optimization Algorithms. In: Vo S, Daduna J (eds) *Computer-Aided Scheduling of Public Transport*, Lecture Notes in Economics and Mathematical Systems, vol 505, Springer Berlin Heidelberg, pp 53–71, DOI 10.1007/978-3-642-56423-9_4

- Ernst A, Jiang H, Krishnamoorthy M, Sier D (2004) Staff Scheduling and Rostering: A Review of Applications, Methods and Models. *European Journal of Operational Research* 153(1):3 – 27, DOI [http://dx.doi.org/10.1016/S0377-2217\(03\)00095-X](http://dx.doi.org/10.1016/S0377-2217(03)00095-X), timetabling and Rostering
- Feillet D, Dejax P, Gendreau M (2005) Traveling Salesman Problems with Profits. *Transportation Science* 39(2):188–205, DOI 10.1287/trsc.1030.0079, URL <http://transci.journal.informs.org/content/39/2/188.abstract>
- Jiann-Sheng W, Tze-Chiang L (2010) Highway Patrol Officer Scheduling Using an Optimization-Based Scheduling Model. In: *Advanced Computer Theory and Engineering (ICACTE)*, 2010 3rd International Conference on, vol 2, pp V2–552–V2–557, DOI 10.1109/ICACTE.2010.5579460
- Knauth P, Hornberger S (2003) Preventive and Compensatory Measures for Shift Workers. *Occupational Medicine* 53(2):109–116, DOI 10.1093/occmed/kqg049
- Kohl N, Karisch S (2004) Airline Crew Rostering: Problem Types, Modeling, and Optimization. *Annals of Operations Research* 127(1-4):223–257
- Lau HC, Gunawan A (2012) The Patrol Scheduling Problem. In: et al DK (ed) *PATAT 2012 - 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT)*, pp 175–192
- Pita J, Jain M, Marecki J, Ordóñez F, Portway C, Tambe M, Western C, Paruchuri P, Kraus S (2008) Deployed ARMOR Protection: the Application of a Game Theoretic Model for Security at the Los Angeles International Airport. In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*, International Foundation for Autonomous Agents and Multiagent Systems, pp 125–132
- Ralphs TK, Saltzman MJ, Wiecek MM (2006) An Improved Algorithm for Solving Biobjective Integer Programs. *Annals of Operations Research* 147(1):43–70, DOI 10.1007/s10479-006-0058-z
- Thorlacius P, Clausen J (2010) Scheduling of Inspectors for Ticket Spot Checking in Urban Rail Transportation. *Trafikdage på Aalborg Universitet*

- Toth P, Vigo D (2002) *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Philadelphia, USA
- Tsai J, Kiekintveld C, Ordonez F, Tambe M, Rathi S (2009) IRIS-a Tool for Strategic Security Allocation in Transportation Networks. In: *Proceedings of the 8th international joint conference on Autonomous agents and multiagent systems: industrial track*, pp 37–44
- Vansteenwegen P, Souffriau W, Oudheusden DV (2011) The Orienteering Problem: A Survey. *European Journal of Operational Research* 209(1):1–10, DOI <http://dx.doi.org/10.1016/j.ejor.2010.03.045>
- Weider S (2007) *Integration of Vehicle and Duty Scheduling in Public Transport*. PhD thesis, Technische Universität Berlin, Berlin, Deutschland
- Yin Z, Jiang AX, Tambe M, Kiekintveld C, Leyton-Brown K, Sandholm T, Sullivan JP (2012) TRUSTS: Scheduling Randomized Patrols for Fare Inspection in Transit Systems Using Game Theory. *AI Magazine* 33(4):59–72