An LP-based heuristic for Inspector Scheduling

Gerwin Gamrath · Markus Reuther ·
Thomas Schlechte · Elmar Swarat

Received: date / Accepted: date

Abstract We present a heuristic based on linear programming (LP) for the integrated tour and crew roster planning of toll enforcement inspectors. Their task is to enforce the proper paying of a distance-based toll on German motorways. This leads to an integrated tour planning and duty rostering problem; it is called Toll Enforcement Problem (TEP). We tackle the TEP by a standard multi-commodity flow model with some extensions in order to incorporate the control tours.

The heuristic consists of two variants. The first, called Price & Branch, is a column generation approach to solve the model's LP relaxation by pricing tour and roster arc variables. Then, we compute an integer feasible solution by restricting to all variables that were priced. The second is a coarse-to-fine approach. Its basic idea is projecting variables to an aggregated variable space, which is much smaller. The aim is to spend as much algorithmic effort in this coarse model as possible. For both heuristic procedures we will show that feasible solutions of high quality can be computed even for large scale industrial instances.

Keywords Crew Rostering · Column Generation · Heuristics

Mathematics Subject Classification (2010) 90B20 · 90C06

Gerwin Gamrath · Elmar Swarat
Zuse Institute Berlin, Takustr. 7, D-14195 Berlin
Tel.: +49-30-84185244
Fax: +49-30-84185269
E-mail: swarat@zib.de

Markus Reuther · Thomas Schlechte
LBW Optimization GmbH, Englerallee 19, 14195 Berlin

1 Introduction

Toll systems are an active research area, especially the aspect of toll road pricing, partly in consideration of congestion on roads. When designing toll systems a common approach is to utilize bi-level programming models as presented in [1] or [2]. Toll pricing to reduce traffic congestion is studied by [3] or more recently by [4].

We consider a rather neglected operational aspect of toll systems, namely the enforcement of the toll. But planning of control resources and in particular the rostering of employees, that conduct the enforcement, is not limited to the case of a toll. It is an important challenge in many real-world applications, e.g., police inspections, ticket inspections or other security related tasks. Here, we focus on the enforcement of the truck toll on German motorways and main roads, a network of around 50k kilometers. All trucks weighting at least 7.5 tonnes must pay a distance based toll. Fares differ according to the vehicle weight, the number of axis and the emission class. Introduced in 2005 on motorways and extended in 2018 to all main roads, it makes one of the most important contribution to the budget of public roads maintenance. Hence, it is needed to organize the limited control resource as effectively as possible.

Since the system is barrier-free, toll evasion is basically possible. The German Federal Office for Goods Transport (BAG) is responsible for the enforcement of the toll. It is conducted by a combination of traffic control gantries or devices for automatic stationary camera control and spot-checks by more than 400 mobile control inspectors. The spot checks are carried out as part of control tours by approximately 250 *control teams* of one or two inspectors. Due to practical aspects the teams are divided into more than 20 regions.

In an on-going research and development project with the BAG we develop methods and a software tool to compute optimal control tours and crew rosters of the inspectors. We have called this integrated tour planning and duty rostering problem the *Toll Enforcement Problem (TEP).* At BAG the planning is organized as monthly planning problem. Two or four sections of the network are controlled during a mobile tour each for a fixed time interval of approximately 2 or 4 hours.

In the TEP a duty corresponds to a control tour starting at a certain time in some depot. After some hours the tour ends in the same depot. The tours are not given in advance, they have to be created by *tasks* consisting of controlling a certain *section* of a motorway (or more precisely a subarea of the toll network) in a corresponding time interval. A tour is a combination of such tasks. The main difference to standard rostering lies in the integrated optimization of tours and rosters. This integration is necessary because it is unclear what work has to be done beforehand and crews can only conduct controls in the area of their home depots. Therefore, it is important to prevent the planning of tours for which no crew is available.

The TEP was presented inter alia in [5] where an extensive description of the modeling power of the approach is given, including an analysis of the bi-criterion character of the TEP and computational results that analyze the complexity of real-world instances. In the literature there are different approaches to solve inspection problems that are similar to our application. The authors of [8] consider the problem of fare evasion. Optimal control strategies are derived by game theory. While our focus is rostering, another recent report [9] considers the generation of duties for railway security teams by concatenating inspection tasks on trains or platforms.

In this paper, we propose a column generation approach to solve the TEP. In a standard approach we solve the LP relaxation of the TEP IP by an arc generation method. After the root node LP is solved to optimality the restricted IP is solved with the set of variables generated during column generation in the root node. In fact, we develop a heuristic since feasibility is not guaranteed.

Our main contribution - that can be seen as a modification of the standard approach - is the adoption of a method called *Coarse-to-Fine*. This method was first used by the authors of [6]. They introduced a coarsening approach to the Railway Rolling Stock Rotation Problem. It is also presented in [7]. The basic idea is to introduce a coarsening projection on the variables of the original model. It leads to a coarser model with significantly less variables. Then the problem is solved via column generation on the coarse level. We will show that the coarse reduced costs overestimate the fine reduced costs. Afterwards, again the IP is solved with the columns generated in the root node. The major benefit in comparison to other heuristic approaches is that we are able to provide a quality measure.

The paper is structured as follows: In Section 2 we shortly define the TEP and recapitulate a corresponding formulation. Thereafter, in Sections 3, 4 and 5 the Coarse-to-Fine approach is presented. Finally, Section 6 discusses computational results for both approaches and Section 7 concludes the paper.

2 A Multicommodity Flow Formulation

The TEP is formulated by a standard multi-commodity flow model with some extensions in order to incorporate the control tours. It is based on a scheduling graph $\mathcal{G} = (D, \mathcal{A}))$. There, the set D corresponds to duties for the inspectors plus artificial source and sink nodes. If two duties $u \in D$ and $v \in D$ can be performed successively by the same inspector we construct an arc $(u, v) \in A$ in $\mathcal{G}$. Additionally, there are arcs $(\hat{s}, v) \in A$ leaving the source node $\hat{s} \in D$ and arcs $(u, \hat{t})$ entering the sink node $\hat{t} \in D$, respectively. Hence, a feasible duty roster corresponds to a $\hat{s}$-$\hat{t}$-path in $\mathcal{G}$. The model uses binary variables z_d to decide if a control tour $d \in D$ is chosen. In addition, there are binary flow variables x_a^m that are one if arc $a \in A$ is used by inspector $m \in M$.

Typical constraints for the tour variables are section covering and section capacity constraints. The first guarantee each section to be visited and the latter that not more than one control team conducts a control on a section during the same time. For the rostering part classical flow value and flow conservation constraints are combined with different types of resource constraints.

Coupling constraints link the tour variables with the roster graph. For more details we refer to our earlier work [5].

The overall goal is to compute a reward-maximal set of control tours. The reward of a section depends both on its amount of traffic depending on the time and the day of the week and on the quota of fare evaders in the past. In addition, costs on the sequence arcs represent penalties for soft rule violations. With this model a lot of rules can be modeled implicitly in the scheduling graph. A typical example are minimum rest times. There exist also other rules, e.g., monthly working hours, that are modeled as resource constraints.

A major task is to find a compromise between quality and quantity of controls as well as providing fair roster schedules for the inspectors such that the acceptance of the optimized schedules can be increased. The modeling power of Mixed Integer Programming and the ability of rapid model modifications in order to cope with moving targets have been an important instrument to solve real-world instances. For practical reasons the tours consist of two parts only (for an efficient control it is required to stay for some time in an area). Therefore, the main complexity of the model stems from the rostering part. The incorporation of the tours can be seen as an extension of the classical rostering flow model. According to the requirement that inspectors can only conduct controls in their regional area, the number of tours and roster sequence variables is still in a range that allows a complete tour generation.

Since the beginning of 2014, the algorithm and software based on this model and the commercial MIP solver CPLEX is in production to schedule all toll enforcement inspectors of BAG in Germany. In contrast to many other rostering problems, model TEP is directly solvable by a general-purpose solver such as CPLEX if the number of arcs and tours does not exceed, say, 2 million. But in some cases the resulting MIP formulation becomes too large and intractable or slow even for commercial MIP solvers. In addition the vast majority of arcs and tours will not be part of a feasible solution. This motivated us to develop a heuristic approach and transfer the idea of dynamic variable generation methods and Coarse-to-Fine to this application.

3 Applying an LP-based heuristic

The main contribution is a heuristic based on Linear Programming (LP) to solve the integrated problem. An important precondition is the fact that the reduced costs can easily be computed in the TEP, since all tours and duty sequence arcs are generated in advance. The heuristic uses a *Coarse-to-Fine* approach to solve the LP-relaxation of the TEP by column generation. The basic idea of Coarse-to-Fine is to identify unprofitable clusters of variables and to focus on the parts that promise an improvement of the objective value. In this concept our original model is called the *fine model*. We introduce a coarsening projection for the variables of the fine model. Several variables from the fine model are mapped to a single variable in the *coarse model*.

The general definition is as follows: We are given a linear program

$$\max p^T x, s.t. Ax \leq b, x \geq 0$$

with J as the index set of the variables. We introduce a coarsening projection

$$[\cdot] : J \to [J]$$

that maps J onto a smaller index set $[J]$, where some variables are aggregated. Therefore, the coarse model has significantly fewer variables than the fine model. Furthermore, for a matrix A let A_{ij} be the entry in the i-th row and in the j-th column. We introduce the coarse matrix $[A] \in \mathbb{R}^{I \times [J]}$ by defining for each $j \in [J]$ the coarse column vector $[A]_{\cdot j}$ as follows:

$$\begin{aligned}[A]_{ij} :=& ([A]_{ij1}, [A]_{ij2}) \\ :=& (\min \{A_{ik} \mid k \in J \,:\, [k] = j\}, \max \{A_{il} \mid l \in J \,:\, [l] = j\}).\end{aligned}$$

The coarse objective coefficients are defined by $[p]_j := \max_{k \in J}\{p_k \mid [k] = j\}$ for all $j \in [J]$. A similar approach where rows were aggregated was introduced to the Railway Rolling Stock Rotation Problem in [6].

The trick is now to solve the fine problem by a column generation algorithm that operates mainly on the coarse level. To price the variables in the coarse model, a proper definition of the *coarse reduced cost* is necessary. Let $\alpha \in \mathbb{R}^I$ be an optimal dual solution for the restricted master problem in the fine layer. Then let us define how to multiply such a vector with a coarse matrix:

$$\hat{x} \in \mathbb{R}^a, \hat{y} \in (\mathbb{R}, \mathbb{R})^a, a \in \mathbb{N}, a \geq 1$$

that

$$\hat{x}^T * \hat{y} := \sum_{i=1}^{a} \min \{\hat{x}_i \hat{y}_{i1}, \hat{x}_i \hat{y}_{i2}\}$$

where $\hat{y}_i = (\hat{y}_{i1}, \hat{y}_{i2})$. Then we can define the coarse reduced cost as:

$$[\tau]_j := [p]_j - \alpha^T * [A]_{\cdot j}, j \in [J].$$

We are now ready to state the crucial property that the coarse reduced costs overestimate the (fine) reduced costs.

Lemma 1 (Coarse Reduced Cost Lemma) *The coarse reduced cost overestimate the (fine) reduced cost:*

$$[\tau]_j = [p]_j - \alpha^T * [A]_{\cdot j} \geq p_k - \alpha^T \cdot A_{\cdot k} = \tau_k \qquad \forall k \in J : [k] = j, j \in [J].$$

Proof Since $[p]_j = \max_{k \in J}\{p_k | [k] = j\}$ it holds that $[p]_j \geq p_k$. It also holds that $[A]_{ij1} \leq A_{ik} \leq [A]_{ij2}$ and with the $*$-operation we get $\alpha^T * [A]_{\cdot j} = \sum_{i \in I} \min \{\alpha_i \cdot [A]_{ij1}, \alpha_i \cdot [A]_{ij2}\} \leq \sum_{i \in I} \alpha_i \cdot A_{ik} = \alpha^T \cdot A_{\cdot k}$. Hence, $[p]_j - \alpha^T * [A]_{\cdot j} \geq p_k - \alpha^T * [A]_{\cdot j} \geq p_k - \alpha^T \cdot A_{\cdot k}$.

4 Coarse-to-Fine column generation

The bounding property of the coarse reduced cost is used to design a Coarse-to-Fine column generation algorithm. Briefly, it works as follows: In the first place, we only have to compute the reduced costs of the coarse variables. If these have a positive value then the reduced costs of the corresponding fine variables will be computed afterwards. Note that we maximize and search for positive reduced cost in this setting. It is described in Algorithm 4.1. There we denote the master problem by MP and the restricted master problem by RMP.

1 **Input** *feasible RMP with start columns* J_0*, coarsening projection* $[\cdot]$ *and finite set of total columns* J *(not part of the RMP yet)*
2 **Output** *RMP with columns* J' *and an optimal solution for the MP*

3 **init** $J' = J_0$
4 **solve** RMP with columns J'
5 **compute** coarse reduced costs $[\tau]$
6 **let** $\hat{J} := \{j \in [J] : [\tau]_j > 0\}$
7 **if** $\hat{J} \neq \emptyset$ **then**
8   **compute** fine reduced costs $\tau_k \quad \forall k \in J : [k] \in \hat{J}$
9   **let** $J^* := \{k \in J \setminus J' : [k] \in \hat{J} : \tau_k > 0\}$
10   **if** $J^* \neq \emptyset$ **then**
11   add $\tilde{J} \subseteq J^*$ to $J', \tilde{J} \neq \emptyset$
12   **goto** 4
13   **end**
14 **end**

Algorithm 4.1: Coarse-to-Fine column generation algorithm.

The input of Algorithm 4.1 is a feasible RMP and a given coarsening projection $[\cdot]$. In line 4 we solve the current RMP with columns J'. In the first iteration these are only the initial columns J_0 that ensure feasibility. As a next step we compute the coarse reduced costs in line 5. If there are no coarse reduced costs with positive value, the algorithm terminates. Otherwise we compute the fine reduced costs for all fine columns which are projections of the coarse columns with positive coarse reduced costs. If none of the fine columns have positive reduced costs, the algorithm finishes as well. But if there are fine columns with positive reduced cost, then we add at least one of them to the model, jump back to line 4 and repeat the same procedure again.

5 Coarse-to-Fine for the TEP

We consider a problem specific approach for the coarsening projection. Namely, for the TEP, the coarsening is processed on the arc variables. Let us consider the scheduling graph $\mathcal{G} = (D, \mathcal{A})$ introduced in Section 2. To prepare the coarsening projection for each day and possible tour, all duty nodes that have similar starting times are aggregated.

We divide a day into blocks of two hours and create coarse nodes (one for each tour, i.e., for each feasible sequence of sections) representing these blocks. The motivation is basically that the expected traffic does not change dramatically during neighboring hours as opposed to considering the whole day. Thus, a coarse decision when a mobile tour starts (or ends) has already a considerable significance.

This coarsening projection on the nodes induces a coarsening projection on the arcs: We aggregate all arcs into a coarse arc that share the same coarse tail node and the same coarse head node. Therefore, for each coarse arc we introduce a coarse arc variable and map the variables belonging to the fine arcs that are aggregated in the current coarse arc to the corresponding coarse variable. In addition, we use a slightly modified definition of coarse reduced costs that takes the particular structure of the TEP scheduling graph into account. All constraints, that correspond to the sequence arc variables, only depend on the corresponding tail and head nodes.

The Coarse-to-Fine approach handles only the inspector arc variables. An artificial tour variable is maintaining feasibility in the beginning and we do one step of column generation on the tour variables in each iteration of the Coarse-to-Fine algorithm. Algorithm 4.1 is applied to the root node LP and afterwards the IP of the fine model is solved with the columns generated during the Coarse-to-Fine algorithm.

We compare this approach with a standard column generation approach, called Price & Branch, to solve the model's LP relaxation by pricing tour and roster arc variables. We can deviate this approach from Algorithm 4.1 by omitting lines $(5-7)$. Then, we again compute an integer feasible solution by restricting to all variables that were priced.

6 Computational Results

In this section we present several computational results for the two presented heuristic approaches. We show how the column generation algorithms decrease the model size of several industrial instances and that both heuristics (Price-and-Branch and Coarse-to-Fine) lead in almost all cases to feasible solutions with a high quality. We aggregated duty nodes sharing the same day, the same control tour, and starting in the same time block, as described in Section 5. The time blocks have a length of two hours.

Table 1 gives the basic data for each instance. Every instance represents a planning scenario for a control region and comprises a time horizon of several weeks. The length of a section control equals four hours. We set a time limit of one day for solving the IP. We performed all computations on an Intel(R) Xeon(R) CPU E5-2670 v2 machine with 2.50GHz and 10 CPU cores. For all computations we used CPLEX (version 12.6.0.0) as LP and IP solver.

There is a broad range of instances from different regions, with a varying number of duty types and fixed duties. Instance *I3* deviates from the other instances in that it is only generated for testing purposes. It covers only a plan-

Inst.	Region	Inspectors	Sections	Fixed Duties	Duty Types	Rows	Columns
I1	r_1	21	28	308	8	16469	164775
I2	r_2	21	13	206	8	14583	347128
I3	r_4	6	28	0	8	2242	27160
I4	r_6	22	22	64	9	20727	705410
I5	r_3	24	28	137	14	31758	2008131
I6	r_5	20	20	37	16	32080	2178483

Table 1 Key characteristics of the TEP instances

ning horizon of one week. Instances I1, I2 and I3 can be solved to optimality without column generation within one day.

For each instance we ran three different variants. The first, the default IP run, is called `noCG`. The second is the Price-and-Branch approach and the third Coarse-to-Fine. Table 2 presents in columns two, three and five the number of variables of the original (master) problem, the number of variables priced by Algorithm `CG`, and by algorithm `CtF`, respectively. In column four and six the relation of the restricted LP for `CG` and `CtF` to the `noCG` is shown in percent. As expected the number of priced variables is much smaller compared to the number of variables in the original model. As an example, instance *I4* has in total around 705,000 variables. The restricted problem after running the Price-and-Branch algorithm has only 97,000 columns. In case of Algorithm 4.1 the column size even reduces a litte more to 84,000. Instance *I5* has even a higher reduction with more than 90%.

Table 3 gives the results of solving the TEP IP both with the full model and with the columns generated by the column generation algorithms. Columns two to four give the solution of the IP run and columns five to seven the objective values either when the time limit is reached or if optimality is proven. We remind that an optimal solution of the IP restricted to the generated variables is not necessary optimal for the original problem but feasible.

On average, we achieved a considerable speed-up with all versions. For instance *I1* the decrease of running time is extreme. For *I3* there is in fact no need for column generation. Fortunately, for all instances feasible solutions could be found when solving the IP with the generated columns. Furthermore, all instances finished before reaching the time limit for `CG` and `CtF`. Raising

Instance	`noCG`	`CG`	%	`CtF`	%
I1	164775	31898	19.4	26366	16.0
I2	347128	60371	17.4	60082	17.4
I3	27160	2749	10.1	2788	10.3
I4	705410	97170	13.8	83921	11.9
I5	2008131	158497	7.9	156550	7.8
I6	2178483	143860	6.6	127646	5.8

Table 2 Model reduction measured by number of variables in the different LP relaxations

Instance	Time IP noCG	CG	CtF	Objective noCG	CG	CtF
I1	29,168.24	58.78	9.73	385,744	379,987	381,568
I2	15,573.29	171.86	131.76	776,919	768,434	766,635
I3	0.48	0.08	0.15	65,392	65,392	65,392
I4	86,408.14	1807.30	482.98	492,754	478,913	475,269
I5	86,400.93	49,973.54	14,930.88	530,344	518,593	517,169
I6	86,414.98	32,005.40	79,280.61	538,749	521,458	520,542

Table 3 Comparison of solution time and quality for the different methods

the time limit for I4 to I6 would lead to a longer running time for `noCG` and therefore to a smaller proportion in terms of running time for `CG` and `CtF`. On the other hand, despite not finishing, version `noCG` yields good IP solutions. In all cases, the IP best incumbent values of `noCG` are better (to a small fraction) than the ones by the heuristics. But in many cases (e.g. *I1* or *I5*) the differences are small enough to claim that our approaches yield good integral solutions. The results give a strong indication that `CtF` yields the smallest models and in many cases the best running times.

7 Conclusion

We presented two heuristics based on linear programming for a rostering problem in the area of toll enforcement on German motorways. We tackle the TEP by a standard multi-commodity flow model with some extensions in order to incorporate the control tours. One heuristic, called Price-and-Branch, is a column generation approach to solve the model's LP relaxation by pricing tour and roster arc variables.

The main contribution is a coarse-to-fine approach. There, several variables from the original model, called fine model, are mapped to a single variable in an aggregated model, the coarse model. First, we presented a generic approach to general linear programs and applied it to the TEP. We discussed the important property that the coarse reduced costs overestimate the fine reduced costs. Then column generation is performed on the coarse level. In both cases (Price-and-Branch and Coarse-to-Fine), we compute an integer feasible solution by restricting to all variables that were priced.

For both heuristic procedures we showed that feasible solutions with high quality can be computed even for large industrial instances. An important issue for future research is to solve instances with a large number of duty types (> 16) or a minor control duration that could not be solved so far. Another idea would be to add an additional algorithmic step to the Coarse-to-Fine, like the coarse reduction [6] in the railway setting, to compute additional suitable columns by fast combinatorial algorithms for a faster convergence.

References

1. Martine Labbé and Patrice Marcotte, and Gilles Savard A Bilevel Model of Taxation and Its Application to Optimal Highway Pricing *in Management Science*, 44(12-part-1):1608-1622, 1998, 10.1287/mnsc.44.12.1608.
2. Luce Brotcorne and Martine Labbé and Patrice Marcotte and Gilles Savard A Bilevel Model for Toll Optimization on a Multicommodity Transportation Network *in Transportation Science*, 35(4):345-358, 2001, 10.1287/trsc.35.4.345.10433.
3. Pia Bergendorff and Donald W. Hearn and Motakuri V. Ramana editor="Pardalos, Panos M. and Hearn, Donald W. and Hager, William W.", Congestion Toll Pricing of Traffic Networks *in Network Optimization*, 51–71, 1997, Springer Berlin Heidelberg, 10.1007/978-3-642-59179-2_4.
4. Tobias Harks and Ingo Kleinert and Max Klimm and Rolf H. Mhring Computing network tolls with support constraints *in Networks*, 65(3):262–285, 2015, 10.1002/net.21604.
5. Ralf Borndörfer and Guillaume Sagnol and Thomas Schlechte and Elmar Swarat Optimal Duty Rostering for Toll Enforcement Inspectors *in Annals of Operations Research*, 252(2):383–406, 2017, 10.1007/s10479-016-2152-1.
6. Ralf Borndörfer and Markus Reuther and Thomas Schlechte A Coarse-To-Fine Approach to the Railway Rolling Stock Rotation Problem *in 14th Workshop on Algorithmic Approaches for Transportation, Modelling, Optimization, and Systems*, 42, 79–91, 2014, 10.4230/OASIcs.ATMOS.2014.79.
7. Markus Reuther Mathematical Optimization of Rolling Stock Rotations PhD Thesis, 2016, Technische Universitt Berlin.
8. Jose Correa and Tobias Harks and Vincent J. C. Kreuzen and Jannik Matuschke Fare Evasion in Transit Networks *in Operations Research*, 65(1):165–183, 2017, 10.1287/opre.2016.1560.
9. Hilbert Snijders and Ricardo L. Saldanha Decision support for scheduling security crews at Netherlands Railways *in Public Transport*, 9(1):193–215, 2017, 10.1007/s12469-016-0142-y.