Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

RALF BORNDÖRFER MARKUS REUTHER
THOMAS SCHLECHTE STEFFEN WEIDER

Vehicle Rotation Planning for Intercity Railways

This work was funded by DB Fernverkehr AG.

ZIB-Report 12-11 (February 2012)

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

Vehicle Rotation Planning for Intercity Railways*

Ralf Borndörfer** Markus Reuther** Thomas Schlechte**
Steffen Weider**

February 17, 2012

Abstract

This paper provides a generic formulation for rolling stock planning problems in the context of intercity passenger traffic. The main contributions are a graph theoretical model and a Mixed-Integer-Programming formulation that integrate all main requirements of the considered Vehicle-Rotation-Planning problem (VRPP). We show that it is possible to solve this model for real-world instances provided by our industrial partner DB Fernverkehr AG using modern algorithms and computers.

1 Introduction

Railway Vehicle-Rotation-Planning is the task to schedule vehicle resources, also called rolling stock, in order to cover the trips of a given timetable by a cost optimal set of *vehicle rotations*. The problem differs significantly from application to application. We consider the case of long distance passenger traffic, namely, the Strategic Vehicle-Rotation-Planning problem (VRPP) of DB Fernverkehr AG (Deutsche Bahn AG [2009]), which is one of the largest intercity railway companies in Europe.

The main challenge in the VRPP is to integrate the treatment of *vehicle composition*, *maintenance*, and *regularity*:

Vehicle composition. The types of basic units of rail cars are called *vehicle groups*. The set of vehicles of the same vehicle group can be seen as a fleet. A timetabled trip can be operated by several alternative *vehicle configurations*. A vehicle configuration is a multiset of vehicle groups. It is part of

*This work was funded by DB Fernverkehr AG.
**Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Email mailto:[surname]@zib.de, URL http://www.zib.de

the VRPP to decide which vehicle configuration is used for which timetabled and, moreover, for which deadhead trip.

Maintenance. A major problem in rolling stock rotation planning is that rail vehicles must be maintained frequently. To ensure safety and robustness of the rolling stock roster, many different maintenance rules have to be obeyed. We focus on cumulative resource constraints with predefined time or distance intervals, *i.e.,* time or distance bounds. This means that one has to sum up the resources (driven length or elapsed time) consumed by a physical vehicle since the last maintenance. If the resource consumption is going to exceed a bound of a maintenance constraint, a suitable *maintenance task* has to be performed in order to replenish the resource.

Regularity. We focus on strategic rolling stock decisions. That is, we consider a cyclic planning horizon over one week – called a *standard week*. The structure of our input schedule is an almost periodic timetable. Only a part of all given trips differ over the week days of the standard week. It is desirable to produce a *regular* rolling stock roster which is compact representable, easy to communicate, and easy to operate, utilizing the periodicity of the timetable. This objective is called *regularity* and is explicitly integrated in our optimization model.

Each of these requirements is already complex in its own right. Moreover, it is almost impossible to treat them sequentially, *i.e.,* a step by step approach easily produces infeasibilities and/or high costs.

The main contributions of this paper are a generic integrated problem description of rolling stock planning problems, a graph theoretical concept for the VRPP, as well as a corresponding Mixed-Integer-Programming formulation. We show that it is possible to solve the resulting model for instances provided by our industrial partner DB Fernverkehr AG using modern algorithms and computers in a reasonable amount of computation time.

The paper is organized as follows. Section 2 gives a literature overview of the known approaches with an emphasis on the maintenance constraints. Section 3 defines the considered problem from a mathematical point of view by introducing a formal description of the VRPP. We embed the rolling stock optimization problem of our industrial partner into our modelling frame in Section 4. Section 5 presents our Integer-Programming approach to solve the VRPP. Computational results for instances given by our industrial partner DB Fernverkehr AG are presented in the final Section 6.

2 Literature

Vehicle scheduling is extensively discussed in the literature, see Löbel [1997] for a survey. Basic multi-commodity flow versions of the problems can nowadays be solved well. Advanced algorithmic approaches, such as Lagrangian pricing, have been developped and it was shown that such methods are able to solve models with up to 70 million arc variables. Current research concentrates on the integration of additonal aspects into the problem; this is indispensible in a railway context. We survey here only the literature that is relevant for this application.

The authors of Ahuja et al. [2005] present a Mixed-Integer-Programming formulation for a locomotive scheduling problem. The model is solved by a very large-scale neighborhood search technique but does not include any maintenance constraints. Savings of over 400 locomotives resulting in over one hundred million dollars annually are reported.

Ziarati et al. [1997] developed a large-scale non-linear Integer-Programming formulation for the integrated optimization of locomotive schedules including maintenance constraints. The proposed model is solved by a Dantzig-Wolfe decomposition within a Branch-and-Bound framework.

Cordeau et al. [2001] proposed an Integer-Programming model based on a Multi-Commodity-Flow formulation for the integrated assignment of locomotives and passenger cars to passenger trips. Maintenance constraints are taken into account by using a time-expanded graph model. Various decomposition techniques embedded in a Branch-and-Bound-and-Cut framework are utilized to solve the problem.

A three stage heuristic approach to incorporate maintenance tasks in precomputed rolling stock rosters is described in Anderegg et al. [2003].

Furthermore two Integer-Programming formulations which can be used to post-optimize rolling stock rosters to incorporate maintenance tasks with very detailed models can be found in Maróti & Kroon [2005] and Maróti & Kroon [2007].

Constraints similar as in vehicle rotation planning come up in related problems as well. In Duty-Rostering problems (especially for public transport companies) there are several constraints on the maximal working time per week and the maximal number of successive working days of the drivers, see Borndörfer et al. [2009]. These constraints are very similar to the maintenance constraints in the VRPP. Behrendt introduced several Mixed-Integer-Programming formulations for the Duty-Rostering Problem in Behrendt [2008] including constraints on cumulative resources, *e.g.,* working time. The author of Behrendt [2008] proposed an integrated model (page 24) for the

optimization of duty rosters under working time constraints. This model is an integration of a binary and a continuous flow modeling the rostering and resources, respectively. The possibility to use this model to solve real world problems was evaluated on data sets from public transport companies of Santiago de Chile and Potsdam in 2008. We will present an adaption of this model for the VRPP.

An alternative "Kanalmodell" is described by Uffmann in her diploma thesis Uffmann [2010]. This model is a transformation of the assignment problem. It uses the structure of the objective function,*i.e.,* all possible assignments within a single station are separately considered. This is done by introducing a so called "Kanal" (engl. channel) which is a kind of cyclic time-line for each station. The transformation is only valid if the objective function satisfies several conditions and has the goal to reduce the number of variables to be considered. A concept for integrating maintenance constraints in this model in an exact way is not provided.

Our adaptation of the formulation for the maintenance requirements proposed in Behrendt [2008] is mathematically equivalent to the model developed independently in Giacco et al. [2011]. The authors of Giacco et al. [2011] reported very promising preliminary computational results for scenarios of an Italian railway company.

Unlike Giacco et al. [2011], we do not assume that the same timetable is repeated every day. Further we do not use sub-tour elimination constraints in our problem specification. In Giacco et al. [2011] sub-tour elimination is done by the approach proposed in Miller et al. [1960] which can also be combined with our model. In addition our objective function is directly related to the cost of a rolling stock roster, in particular it does incorporate deadhead costs and real vehicle cost.

Our previous paper Borndörfer et al. [2011] shows how vehicle composition and regularity aspects can be integrated in a very compact hypergraph based Integer-Programming formulation. The model of this paper is a further development of this work in order to handle maintenance constraints.

3 The Vehicle-Rotation-Planning Problem

In this section we will introduce the VRPP using a graph theoretical description. We start with a formal description. Afterwards we will describe how the VRPP can be instantiated and interpreted for intercity railway planning and we will explain some technical aspects.

First, we define the main technical sets to provide a relation between the

graph theoretical definitions and the real-world problem. We use the following terminology.

The set of timetabled passenger trips is denoted by T. In a solution of the VRPP one has to perform *maintenance tasks* on vehicles. Let M be the set of all possible maintenance tasks. A *service* s is a non-empty set of maintenance tasks, *i.e.,* $s \in 2^M \setminus \{\emptyset\} =: S$. We say that $s \in S$ *implements* $m \in M$ if $m \in s$.

The Vehicle-Rotation-Planning problem integrates three main aspects, namely vehicle composition, regularity, and maintenance. To express the specific settings clear, and exact, we define the problem in terms of a hypergraph, *i.e.,* a graph which contains standard arcs as well as *hyperarcs*. From a high level point of view, one could say that the standard arcs model what is possible to do for physical vehicles while the hyperarcs model what is possible to decide for the VRPP.

Definition 1 *(VRPP hypergraph) Let V be a set of nodes, S be a set of services, and let $A \subseteq (V \cup S)^2$ be a set of directed arcs. We define a set $\mathcal{A} \subseteq 2^A$, called* hyperarcs. *The VRPP* hypergraph *is denoted by $G = (V \cup S, A, \mathcal{A})$.*

In contrast to most of the hypergraph literature, it is convenient in our setting to conceive a hyperarc as a set of standard arcs. Namely, the hypergraph G can then be seen as a standard directed graph $(V \cup S, A)$ extended by a set of hyperarcs $\mathcal{A}$.

A node $v \in V$ of G represents the departure or arrival of a physical vehicle which operates a timetabled trip of T. A node $v \in S$ represents a service, *i.e.,* a set of maintenance tasks.

We say that the arc $a = (u, v)$ *operates* a trip $t \in T$ if $u \in V$ represents the departure of t and $v \in V$ represents the arrival of t. Therefore the arcs of G model how vehicles can operate the timetable, how they can move between trips, and how they can be maintained.

A *hyperarc* $\mathfrak{a} \in \mathcal{A}$ is a set of arcs of A, *i.e.,* $\mathfrak{a} \subseteq A$. We say that the hyperarc $\mathfrak{a} \in \mathcal{A}$ *covers* $t \in T$, if each arc $a \in \mathfrak{a} \subseteq A$ operates t.

Definition 2 *(maintenance constraint) A* maintenance constraint l *is represented by a resource function $r_l : S \cup A \mapsto \mathbb{Q}_+$, a* resource upper bound $U_l \in \mathbb{Q}+$, *and a set of maintenance tasks $m_l \subseteq M$. We say that a maintenance task of m_l must be performed to* reset *the resource r_l to fulfill the bound U_l.*

Definition 3 *(feasible path) A* feasible path $P \subseteq A$ *in G w.r.t. the maintenance constraint l is a simple path starting and ending at nodes of S (which*

implement a appropriate maintenance task of $m_l \subseteq L$) resetting the resource of l such that:

$$\sum_{S(P)} r_l(v) + \sum_{a \in P} r_l(a) \leq U_l. \tag{1}$$

The inequality (1) states that the sum of all consumed resources on a feasible path P has to be smaller then or equal to the bound of a maintenance constraint. Note that even a service $s \in S(P) \subset S$ can consume resources if it does not implement a maintenance task resetting the constraint.

Definition 4 *(feasible rotation) A* feasible rotation *w.r.t. the maintenance constraint l is a cycle $C \subseteq A$ such that each node covered by C is contained in a feasible path w.r.t. l.*

Performing a maintenance tasks consumes some commodities, *i.e.,* crew, machines, and infrastructure. Those commodities have usually a limited availability.

Definition 5 *(capacity constraint) A* capacity constraint *b is represented by a resource function $r_b : \mathcal{A} \mapsto \mathbb{Q}_+$ and a* capacity bound *$U_b \in \mathbb{Q}+$. The set of all capacity constraints is denoted by B. We say that the set of hyperarcs $\mathcal{A}_0 \subseteq \mathcal{A}$ fulfills the capacity constraint $b \in B$ if $\sum_{\mathfrak{a} \in \mathcal{A}_0} r_b \leq U_b$.*

Now we have defined all needed terms to state the problem.

Definition 6 *(Vehicle-Rotation-Planning Problem (VRPP))*

Given a VRPP hypergraph $G = (V \cup S, A, \mathcal{A})$ with a cost function $\mathbf{c} : \mathcal{A} \mapsto \mathbb{Q}_+$, a set of maintenance constraints L, and a set of capacity constraints B. The VRPP is to find a cost minimal set of hyperarcs $\mathcal{A}_0 \subseteq \mathcal{A}$ such that:

- *Each timetabled trip $t \in T$ is covered by exactly one hyperarc $a \in \mathcal{A}_0$.*
- *The set $\bigcup_{a \in \mathcal{A}_0} a$ is a set of feasible rotations w.r.t. all maintenance constraints of L.*
- *The set $\mathcal{A}_0$ fulfills all capacity constraints of B.*

Let $t \in T$ be a trip. We define the set of all hyperarcs covering t as $\mathcal{A}(t) := \{\mathfrak{a} \in \mathcal{A} \mid \mathfrak{a} \text{ covers } t\}$.

If we assume that all arcs of the set $\mathcal{A}$ have cardinality one, we call the resulting problem the *non-hyper relaxation* of the VRPP and if we assume that the VRPP does not have any maintenance constraints and also does not have any capacity constraints, we call the relaxed problem the *non-maintenance* relaxation of the VRPP.

If we consider the non-hyper relaxation which is also a non-maintenance relaxation of the VRPP, the problem reduces to an Integer Multi-Commodity-Flow problem, which is known to be $\mathcal{NP}$-hard, see Löbel [1997]. In this reduction the commodities are represented by the sets $\mathcal{A}(t)$. Therefore, the VRPP is also an $\mathcal{NP}$-hard combinatorial optimization problem. In addition, if we assume that each set $\mathcal{A}(t)$ has cardinality one, the problem reduces to the standard assignment problem. It should also be mentioned that the non-hyper relaxation of the VRPP with exactly one maintenance constraint is closely related to a variant of the Vehicle-Routing-Problem, see Crevier et al. [2007].

4 The VRPP for intercity railway planning

In this section we give some motivation to explain how our formalism introduced in Section 3 is related to a real-world instance of the VRPP for intercity railway planning w.r.t. the main requirements as vehicle composition, regularity, and maintenance.

4.1 Vehicle composition

As introduced in Section 1 the types of basic vehicle units are called *vehicle groups*. A vehicle group can be seen as a fleet.

Definition 7 *(vehicle problem data) Let F be the set of vehicle groups and C be the set of vehicle configurations. A* vehicle configuration *$c \in C$ is a multiset of vehicle groups of F. The set $C(t) \subseteq C$ denotes the set of feasible vehicle configurations to cover the timetabled trip $t \in T$.*

The relation of vehicle groups and vehicle configurations plays an important role in intercity planning. This is because rail vehicles can be coupled together. In intercity problems even *on the fly*. This means that no technical equipment or crew is needed for a coupling. There exist vehicle groups for which the coupling time does not exceed ten minutes. Coupling activities create a huge number of degrees of freedom in intercity planning.

On the other hand, there are a lot of technical rules regarding to this coupling activities, *e.g.,* rules for the position of vehicles in a configuration. Moreover there are rules for the *orientation* of vehicles in a configuration, namely there are two possible directed orientations of physical vehicles. An orientation of a vehicle can be determined by considering the direction of movement of the vehicle on a track. In intercity planning these orientations are characterized by the alignment of the first-class carriages. For some stations it is desired

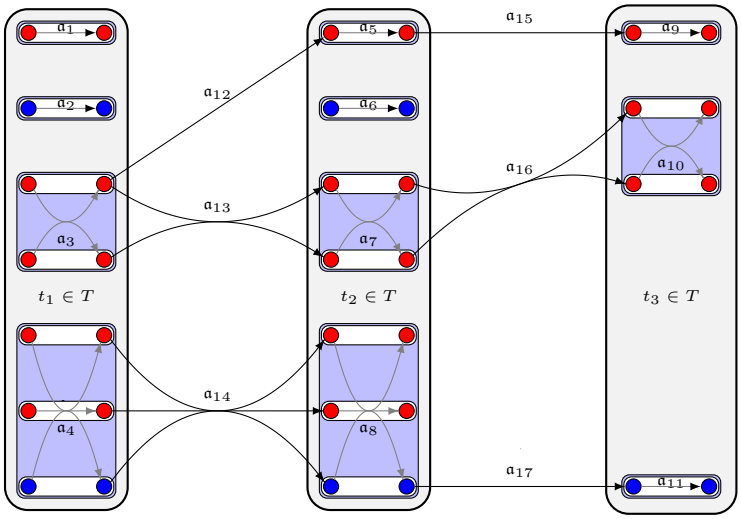

Figure 1: Hypergraph model.

that first-class carriage arrives or departs at special parts of the passenger platform.

Our model is directly based on decisions for coupled vehicles and thus it is able to handle all technical requirements in an integrated manner. This paper is not about detailed position and orientation requirements for intercity planning. We only wanted to mention that such rules exist and have to be integrated in a decision support system for real-world railway applications.

Figure 1 shows how vehicle groups and vehicle configurations are related and how they are modeled in the VRPP. The picture shows three timetabled trips $t_1, t_2, t_3 \in T$. All red and blue circles are nodes of the node set V of the VRPP hypergraph $G = (V, A, \mathcal{A})$, *i.e.,* departures or arrivals of physical vehicles of the three timetabled trips. The set of services S as well as the set of standard arcs A are not illustrated in this picture, *i.e.,* it only shows the decisions of the VRPP.

The colors of the circles indicate two vehicle groups, *i.e.,* two fleets – a red and a blue one. As one can imagine, in intercity rotation planning it is not allowed to connect nodes of different vehicle groups in the VRPP hypergraph and therefore this picture illustrates also the reduction of the VRPP to an Integer Multi-Commodity-Flow problem which was described in Section 3. Here the vehicle groups are the commodities.

The hyperarcs $\mathfrak{a}_1, \mathfrak{a}_2, \mathfrak{a}_3, \mathfrak{a}_4 \in \mathcal{A}$ form the set $\mathcal{A}(t_1)$, *i.e.,* the set of alternative hyperarcs which can be used to cover the timetabled trip t_1, *i.e.,* $\mathcal{A}(t_1)$ represents the set $C(t_1)$ of feasible vehicle configurations to operate t_1. From

a practical point of view this can be seen as follows: It is feasible to cover t_1 by a single vehicle of the red or blue fleet, see arcs $\mathfrak{a}_1$ and $\mathfrak{a}_2$. But it is also feasible to *haul* up to two vehicles by operating t_1, see arcs $\mathfrak{a}_3$ and $\mathfrak{a}_4$. As mentioned above, the position and orientation of vehicles in vehicle configurations must be also taken into account. This can be easily done by extending this approach, *i.e.,* declaring the nodes of G as states w.r.t. position and orientation of vehicles at departures and arrivals of trips. But all this hyper-detail-rocket-modeling can only be done at the expense of a growth of the VRPP hypergraph.

While the arcs $\mathfrak{a}_1, \mathfrak{a}_2, \ldots, \mathfrak{a}_{11}$ model how the trips can be covered, the arcs $\mathfrak{a}_{12}, \mathfrak{a}_{13}, \ldots, \mathfrak{a}_{17}$ model how the trips can be connected to build a set of feasible rotations. Arc $\mathfrak{a}_{12} \in \mathcal{A}$ implements a coupling activity after the arrival of t_1. The hyperarcs $\mathfrak{a}_{13}, \mathfrak{a}_{14}, \mathfrak{a}_{15}, \mathfrak{a}_{16} \in \mathcal{A}$ model connections between trips without coupling activities.

As defined in Section 3, each hyperarc $\mathfrak{a}_1, \mathfrak{a}_2, \ldots, \mathfrak{a}_{17}$ is a set of arcs of A. This implies that the arcs $\mathfrak{a}_1, \mathfrak{a}_2, \mathfrak{a}_5, \mathfrak{a}_6, \mathfrak{a}_9, \mathfrak{a}_{11}, \mathfrak{a}_{12}, \mathfrak{a}_{15}, \mathfrak{a}_{17}$ are sets of cardinality one of the arc set A, while the arcs $\mathfrak{a}_3, \mathfrak{a}_7, \mathfrak{a}_{10}, \mathfrak{a}_{13}, \mathfrak{a}_{16}$ are of cardinality two, and the arcs $\mathfrak{a}_4, \mathfrak{a}_8, \mathfrak{a}_{14}$ are of cardinality three. Note that it is important for the VRPP that hyperarcs are defined as sets of standard directed arcs because the set of rotations must be well defined for the exact treatment of the maintenance requirements, which we consider in the next sub-section.

4.2 Maintenance constraints

To model these requirements, we consider each possible path of length two of the form $\{(v, s), (s, w)\} \subseteq A$ with $s \in S$ and $v, w \in V$, called *service path*, as a single hyperarc $\mathfrak{a} = \{(v, s), (s, w)\} \in \mathcal{A}$. We call $\mathfrak{a}$ a *replenishment arc*. Because we assume that our objective function is non-negative, we do not have to consider any other structures where service nodes appear, *e.g.,* cycles, paths, or loops of service nodes.

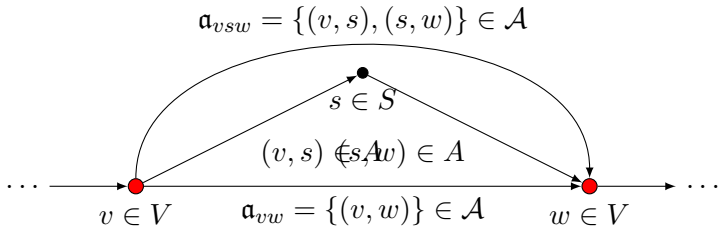

Figure 2: Service path representation.

Figure 2 illustrates the treatment of service paths in our VRPP hypergraph.

The hyperarc $\mathfrak{a}_{vw}$ models a direct connection of the arrival node $v \in V$ and the departure node $w \in V$. These connections can include deadhead trips (*i.e.,* empty rides) if the involved locations are different. The service path $\{(v, s), (s, w)\} \subseteq A$ is represented by the hyperarc $\mathfrak{a}_{vsw} \in \mathcal{A}$. It models that a vehicle configuration arrives at v, traverses all maintenance tasks that are implemented in the service $s \in S$, and finally departs on w. Also $\mathfrak{a}_{vsw}$ can include several deadhead trips. Note that the consideration of a service path as a single arc is necessary for the construction of the VRPP hypergraph for intercity railway planning, because it can only be decided if an arc exists or not by evaluating technical rules w.r.t. to a whole service path.

Since a service is an element of the power-set of all maintenance tasks the number of parallel hyperarcs representing service paths can grow very excessive. Especially in case of different locations of the maintenance tasks on a service path or if the number of maintenance constraints is large, the set of parallel service paths must be considered implicitly. In addition, if the detailed schedule of the maintenance tasks on a service path is important, the number of service paths to be considered does also increase significantly.

4.3 Regularity

As mentioned in Section 1, we focus in this paper on a cyclic planning horizon over one *standard week*. The structure of the given timetable is *almost* periodic. Only few trips of the timetable differ over the single week days of the standard week. In view of this structure, it is desirable to construct a vehicle rotation plan which utilizes this periodicity. We call such a plan a *regular* vehicle rotation plan.

Imagine that we are given a timetable for that each trip repeats every day in the standard week as it was considered in Giacco et al. [2011]. We call such a timetable a *periodic timetable* and we call a set of repeating trips in the standard week a *train*. The set of trains is denoted by $\mathfrak{T}$. A periodic timetable can also be considered as input data for the VRPP. But in this case the set of trains $\mathfrak{T}$ can be viewed as the set of trips T (this is not quite accurate w.r.t. maintenance constraints). In case of a standard week with seven week days, this reinterpretation leads to a VRPP hypergraph where the number of nodes is reduced by a divisor of seven. Since an instance of the VRPP hypergraph is very dense (almost complete), the number of arcs reduces by a divisor of 49. A hyperarc in the VRPP hypergraph for a periodic timetable can be seen as a set of hyperarcs if the trips are individually considered. This motivates our approach for integrating regularity aspects to the VRPP. We easily construct a set of hyperarcs which are sets of other (individual) hyperarcs.

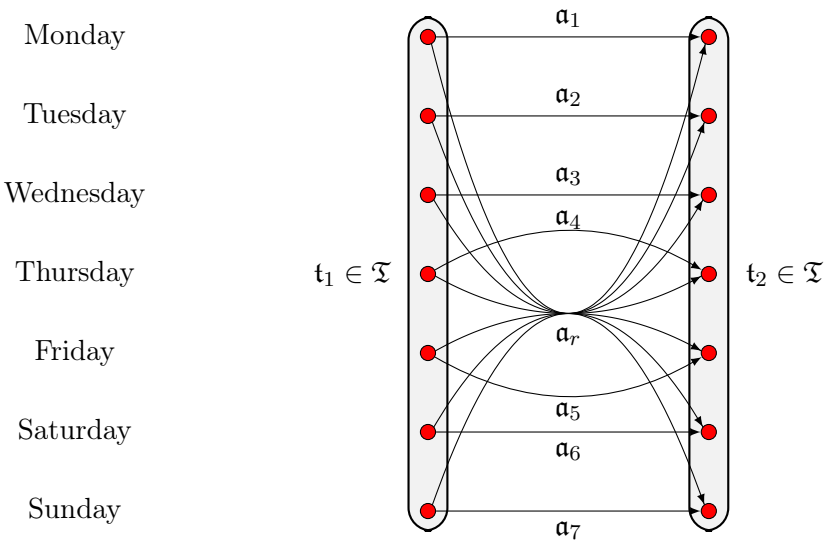

Figure 3: Hyperarc model for regularity.

Figure 3 illustrates our regularity approach. The red circles for train $\mathfrak{t}_1 \in \mathfrak{T}$ can be seen as *equal* arrivals of each trip $t \in \mathfrak{t}_1 \subseteq T$. Equal means equal arrival locations and equal points in time in the standard week. The set of circles for train $\mathfrak{t}_2 \in \mathfrak{T}$ represents equal departures of all trips of $\mathfrak{t}_2$. The individual hyperarcs $\mathfrak{a}_1, \mathfrak{a}_2, \ldots, \mathfrak{a}_7$ may not be simultaneously chosen in a solution of the VRPP. To express that this is desired, we create the hyperarc $\mathfrak{a}_r \in \mathcal{A}$ as $\mathfrak{a}_r = \bigcup_{i=1}^{7} \mathfrak{a}_i \subseteq A$. From an applied point of view, the purpose of this regularity policy is to create rolling stock plans that are compactly representable, easy to communicate, and easy to operate. A more detailed description of the technical aspects of train composition and regularity can be found in Borndörfer et al. [2011].

5 A Mixed-Integer-Programming approach

Let $G = (V \cup S, A, \mathcal{A})$ be a given VRPP hypergraph with a cost function $\mathbf{c} : \mathcal{A} \mapsto \mathbb{Q}_+$, $\mathbf{c}(\mathfrak{a}) := \mathbf{c}_{\mathfrak{a}}$ as introduced in Section 3. Further, let L be a set of maintenance constraints with a resource function $r_l : S \cup A \mapsto \mathbb{Q}_+$ and a resource upper bound $U_l \in \mathbb{Q}+$ for each constraint $l \in L$. In addition, let B be a set of capacity constriants with a resource functions $r_b : \mathcal{A} \mapsto \mathbb{Q}_+$ and capacity bounds $U_b \in \mathbb{Q}+$.

Let $t \in T$ be a trip and $a \in A$ be a set of standard arcs. We define the set $\mathcal{A}(t)$ of arcs covering t and the set $\mathcal{A}(a)$ of hyperarcs of a as:

$$\begin{aligned}\mathcal{A}(t) &:= \{\mathfrak{a} \in \mathcal{A} \,|\, \mathfrak{a} \text{ covers } t\},\\ \mathcal{A}(a) &:= \{\mathfrak{a} \in \mathcal{A} \,|\, a \in \mathfrak{a}\}.\end{aligned}$$

W.l.o.g. we assume that $\mathcal{A}(a) \neq \emptyset$ for each arc $a \in A$. If $\mathcal{A}(a) = \emptyset$, the standard arc a can never be contained in a feasible set of rotations.

For a node $v \in V$ we define sets of incoming and outgoing (hyper-) arcs of v in the VRPP hypergraph G as:

$$\begin{aligned}\mathcal{A}(v)^{\text{in}} &:= \{\mathfrak{a} \in \mathcal{A} \,|\, \exists\, a \in \mathfrak{a} \,:\, a = (u, v)\},\\ A(v)^{\text{in}} &:= \{(u, v) \in A\},\\ \mathcal{A}(v)^{\text{out}} &:= \{\mathfrak{a} \in \mathcal{A} \,|\, \exists\, a \in \mathfrak{a} \,:\, a = (v, w)\},\\ A(v)^{\text{out}} &:= \{(v, w) \in A\}.\end{aligned}$$

We introduce a binary decision variable $x_{\mathfrak{a}}$ for each hyperarc $\mathfrak{a} \in \mathcal{A}$. In addition we define a non-negative continuous variable w_a^l for each standard arc $a \in A$ and each maintenance constraint $l \in L$ fulfilling the upper bound U_l.

Let $\mathfrak{a} \in \mathcal{A}$ be a hyperarc and $a = (u, v) \in \mathfrak{a}$ a standard arc. To permit simple notation we define the resource consumption $r_l(v) := 0$ for a node $v \in V$ and we introduce the symbol $r_l^v(\mathfrak{a})$:

$$r_l^v(\mathfrak{a}) := r_l(\,(u, v) \in \mathfrak{a}\,) + r_l(v). \tag{2}$$

The symbol $r_l^v(\mathfrak{a})$ states the sum of the resource consumption of the from v outgoing standard arc $(u, v) \in \mathfrak{a}$ and the resource consumption of the node v. W.l.o.g. we can assume that the standard arc (u, v) is unique in $\mathfrak{a}$, because we the set of standard arcs of a hyperarc must forms a perfect matching between the tail and head nodes of this arc set. If this is not the case, there must be a node with two or more incoming/outgoing standard arcs and this can not be contained in a set of rotations.

5.1 Mixed-Integer-Program

The VRPP can now be stated as a Mixed-Integer-Program as follows:

$$\min \sum_{\mathfrak{a} \in \mathcal{A}} \mathbf{c}_{\mathfrak{a}} x_{\mathfrak{a}}, \qquad \text{(objective)}$$

$$\sum_{\mathfrak{a} \in \mathcal{A}(t)} x_{\mathfrak{a}} = 1 \qquad \forall t \in T, \tag{3}$$

$$\sum_{\mathfrak{a} \in \mathcal{A}(v)^{\text{in}}} x_{\mathfrak{a}} = \sum_{\mathfrak{a} \in \mathcal{A}(v)^{\text{out}}} x_{\mathfrak{a}} \qquad \forall v \in V, \tag{4}$$

$$w_a^l \leq \sum_{\mathfrak{a} \in \mathcal{A}(a)} U_l \, x_{\mathfrak{a}} \qquad \forall a \in A,\, l \in L, \tag{5}$$

$$\sum_{a \in A(v)^{\text{out}}} w_a^l - \sum_{a \in A(v)^{\text{in}}} w_a^l = \sum_{\mathfrak{a} \in \mathcal{A}(v)^{\text{out}}} r_l^v(\mathfrak{a}) \, x_{\mathfrak{a}} \qquad \forall v \in V,\, l \in L, \tag{6}$$

$$\sum_{\mathfrak{a} \in \mathcal{A}} r_b(\mathfrak{a}) \, x_{\mathfrak{a}} \leq U_b \qquad \forall b \in B, \tag{7}$$

$$x_{\mathfrak{a}} \in \{0, 1\} \qquad \forall \mathfrak{a} \in \mathcal{A}, \tag{8}$$

$$w_a^l \in [0, U_l] \subset \mathbb{Q}_+ \qquad \forall a \in A,\, l \in L. \tag{9}$$

The linear (objective) function minimizes the total cost and is directly related to the cost of operating a timetable. For each trip $t \in T$ the covering constraints (3) assign exactly one hyperarc of $\mathcal{A}(t)$ to t. The equalities (4) are flow conservation constraints for each node $v \in V$ that imply the set of rotations in the arc set A. The subset of constraints (3), (4), and (8) state the non-maintenance relaxation of the VRPP. This can also be seen as a hyper-assignment as it was considered in Borndörfer et al. [2011].

The constraints (5) and (6) ensure that the hyper-assignment is feasible w.r.t. to all maintenance constraints $l \in L$.

Let the x-variables be fixed such that they imply a set of rotations, *i.e.,* a set of cycles. The constraints (5) imply that a w_a^l can only be non-zero if a corresponding hyperarc of $\mathcal{A}$ consists of a standard arc which corresponds to a. Therefore the w-flow traverses the same set of cycles that the x-flow implies. Let $v \in V$ be a node and let $l \in L$ be a maintenance constraint. Suppose that $a^{\mathtt{out}} \in A(v)^{\text{out}}$, $a^{\mathtt{in}} \in A(v)^{\text{in}}$, and $\mathfrak{a} \in \mathcal{A}(v)^{\text{out}}$ are the in v outgoing and incoming (hyper-) arcs in the considered fixed x. Since $x_{\mathfrak{a}} = 1$ equation (6) for v and l reduces to:

$$w_{a^{\mathtt{out}}}^l = w_{a^{\mathtt{in}}}^l + r_l^v(\mathfrak{a}). \tag{10}$$

Equation (10) states that the flow value of an arc, namely $w_{a^{\mathtt{out}}}^l$, is always the sum of the flow value of the predecessor arc $w_{a^{\mathtt{in}}}^l$ and the actual resource

consumption $r_l^v(\mathfrak{a})$. We call (6) *resource flow constraints*. Note that there are no resource flow constraints for service nodes, such that the cumulative flow for each constraint is replenished by arcs that implement appropriate maintenance tasks.

The inequalities (7) are the canonical formulation of the capacity constriants.

5.2 Solving the VRPP

In this sub-section we describe the current state of our algorithm to solve the VRPP. Currently, this algorithm is under development and we therefore give a short high-level description at this time.

Since the number of variables and constraints is very large, *i.e.,* one variable for each hyperarc and one constraint for each standard arc (this is the dominating part), we shrink the set of active variables by a column generation algorithm. This algorithm is quite simple. The pricing problem is to decide whether there exists a hyperarc with negative reduced cost. Note that we only have to price x-variables, if we add the corresponding w-variables simultaneously. More precisely, if we found a hyperarc $\mathfrak{a} \in \mathcal{A}$ with negative reduced cost to be added to the active model, we add also all w_a with $a \in \mathfrak{a}$. This is correct because constraints (5) state that the w-variables can only be active if the corresponding x-variables are active. Moreover we add the coupling constraints (5) dynamically, *i.e.,* if the corresponding w-variable is not in the active model, we do not have to consider the corresponding constraint. Which hyperarcs are priced during an iteration can be summarized by: The best (w.r.t. the reduced cost) $n \in \mathbb{N}_+$ outgoing hyperarcs for each node. This simple strategic provides the ability to solve the LP-relaxation within a Branch-and-Bound search tree in a reasonable amount of time. We do not price any variables in the nodes of the search tree.

After we generated a set of columns and constraints such that no more arcs with negative reduced cost can be found we try to fix the hyperarcs which cover the timetabled trips, *i.e.,* we declare all variables for the columns $\bigcup_{t \in T} \mathcal{A}(t)$ to be integer feasible. This is done by the Rapid-Branching method, see Weider [2007], Borndörfer et al. [2010], and Borndörfer et al. [2011]. After that algorithm is finished, it is fixed which hyperarc covers which timetabled trip and this decisions are fixed during the whole algorithm. An Integer-Multi-Commodity-Flow problem would be solved at this state.

To compute a primal upper bound we use a local search heuristic, which starts with the optimal solution of the non-maintenance relaxation of the active model and aims to find an integer feasible solution.

After the column generation, Rapid-Branching, and primal heuristic we use `CPLEX 12.2` to solve the generated model so far, *i.e.,* a restricted variant of the overall model, as a static MIP. If the local search heuristic succeeded, we provide the primal solution to `CPLEX`. By means of this approach we can provide valid global lower and upper bounds for the value of the objective function of a optimal solution.

6 Computational study

In this section we provide a computational study for real-world instances of our industrial partner DB Fernverkehr AG. The considered instances include scenarios for the currently operated high speed intercity vehicles (ICE) as well as studies for future rolling stock fleets. All our computations were performed on computers with an Intel(R) Xeon(R) CPU X5672 with 3.20 GHz, 12 MB cache, and 48 GB of RAM. CPLEX Barrier as LP-solver was running with 4 threads as well as the CPLEX MIP-solver.

Table 1 summarizes characteristics of our instances. The second column ($|\mathfrak{T}|$) states the number of trains, which result in the number of trips ($|T|$, column 3) in the considered timetable. The next two columns indicate how many vehicle configurations C and vehicle groups F (fleets) are given. The number of maintenance constraints that we consider is denoted in column $|L|$ and the last column states the number of capacity constraints.

Table 2 gives our computational results including some facts about the VRPP hypergraph. The overall number of hyperarcs is denoted in column $|\mathcal{A}|$, while the columns $|\mathcal{A}_C|$, $|\mathcal{A}_S|$, and $|\mathcal{A}_{\mathtt{reg}}|$ state the number of hyperarcs $\mathfrak{a} \in \mathcal{A}$ with $|\mathfrak{a}| > 1$ to model vehicle composition, the number of replenishment arcs, and the number of hyperarcs that where constructed to model the regularity requirements. The column $\mathfrak{v}$ is the number of physical vehicles that are needed to operate the given timetable in the computed solution. The values in column gap are defined by the difference between the objective value of the best integer feasible solution $\mathbf{c}_{\mathtt{UB}}$ and the objective value of the best lower bound $\mathbf{c}_{\mathtt{LB}}$ as $100 \cdot (\mathbf{c}_{\mathtt{UB}} - \mathbf{c}_{\mathtt{LB}}) / (\mathbf{c}_{\mathtt{UB}} + 10^{-10})$. Finally, we denote the total running time in CPU seconds.

A zero in column $|L|$ of Table 1 indicates an instance of the non-maintenance relaxation of the VRPP. All other instances include one maintenance constraint. If the value of the column $|C|$ is equal to the value of the column $|F|$ in Table 1 the instance has no train composition aspects. In addition, if there are no regularity aspects, *i.e.,* if $|\mathcal{A}_{\mathtt{reg}}| = 0$ in Table 2, the resulting problem is an instance of the non-hyper relaxation of the VRPP. Therefore the instances `vrp14`, `vrp16`, `vrp18`, `vrp20`, `vrp23`, `vrp25`, `vrp27`, `vrp29`, `vrp31`, `vrp33`, `vrp35`, and `vrp37` are standard Integer Multi-Commodity-Flow prob-

lems of large scale. Instances of real practical interest are the instances `vrp38` – `vrp69`. The results show the expected behavior of algorithms that try to solve the VRPP. Train composition and regularity aspects are not too hard to tackle. While the maintenance constraints and moreover capacity constraints increase the complexity of the VRPP significantly. As one can see, we have to do further development of our algorithm, see instances `vrp63`, `vrp61`, `vrp52`, and `vrp50`.

Our computational study demonstrate that our solution approach can be used to produce high quality solutions for large-scale Vehicle-Rotation-Planning problems.

References

Ahuja, Liu, Orlin, Sharma & Shughart (2005). Solving Real-Life Locomotive-Scheduling Problems. *Transportation Science 39, 503–517.*

Anderegg, Eidenbenz, Gantenbein, Stamm, Taylor, Weber & Widmayer (2003). Train Routing Algorithms: Concepts, Design Choises, and Practical Considerations. In *ALENEX*, pp. 106–118.

Behrendt (2008). Dienstreihenfolgeplanung mit ganzzahliger Optimierung. Master's thesis, TU Berlin.

Borndörfer, Grötschel & Jaeger (2009). Planning Problems in Public Transit. Technical Report 09-13, ZIB, Takustr.7, 14195 Berlin.

Borndörfer, Reuther, Schlechte & Weider (2011). A Hypergraph Model for Railway Vehicle Rotation Planning. In Caprara & Kontogiannis (Eds.), *11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 20 of *OpenAccess Series in Informatics (OASIcs)*, pp. 146–155., Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Borndörfer, Schlechte & Weider (2010). Railway Track Allocation by Rapid Branching. In Erlebach & Lübbecke (Eds.), *Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 14 of *OpenAccess Series in Informatics (OASIcs)*, pp. 13–23., Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Cordeau, Soumis & Desrosiers (2001). Simultaneous Assignment of Locomotives and Cars to Passenger Trains. *Oper. Res. 49, 531–548.*

Crevier, Cordeau & Laporte (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research 176(2), 756–773.*

Deutsche Bahn AG (2009). DB Mobility Logistics Daten & Fakten 2009.

Giacco, D'Ariano & Pacciarelli (2011). Rolling stock rostering optimization under maintenance constraints. In *Proceedings of the 2nd International Conference on Models and Technology for Intelligent Transportation Systems*, pp. 1–5., Leuven, Belgium. MT-ITS 2011.

Löbel (1997). *Optimal Vehicle Scheduling in Public Transit*. Aachen: Shaker Verlag. Ph.D. thesis, Technische Universität Berlin.

Maróti & Kroon (2005). Maintenance Routing for Train Units: The Transition Model. *Transportation Science 39, 518–525*.

Maróti & Kroon (2007). Maintenance routing for train units: The interchange model. *Computers & OR 1, 1121–1140*.

Miller, Tucker & Zemlin (1960). Integer programming formulation of traveling salesman problems. *J. Assoc. Comput. Mach. 7, 326–329*.

Uffmann (2010). Das Kanalmodell zur Effizienzsteigerung in der Fahrzeugumlaufplanung. Master's thesis, Göttingen, Univ., Inst. f. Num. u. Angew. Math.,.

Weider (2007). *Integration of Vehicle and Duty Scheduling in Public Transport*. PhD thesis, TU Berlin.

Ziarati, Soumis, Desrosiers, Gélinas & Saintonge (1997). Locomotive assignment with heterogeneous consists at CN North America. *Eur. J. Oper. Res. 97(2), 281–292*.

instance	$\lvert\mathfrak{T}\rvert$	$\lvert T\rvert$	$\lvert C\rvert$	$\lvert F\rvert$	$\lvert L\rvert$	$\lvert B\rvert$
vrp01	126	617	1	1	1	0
vrp02	126	617	1	1	1	0
vrp03	126	617	1	1	1	0
vrp04	126	617	1	1	0	0
vrp05	288	884	1	1	0	0
vrp06	165	884	1	1	1	0
vrp07	126	617	1	1	1	0
vrp08	43	267	1	1	0	0
vrp09	43	267	1	1	1	0
vrp10	61	310	1	1	1	0
vrp11	61	310	1	1	1	0
vrp12	288	2031	6	4	1	0
vrp13	298	1443	6	6	0	0
vrp14	298	1443	6	6	0	0
vrp15	298	1443	24	24	0	0
vrp16	298	1443	24	24	0	0
vrp17	298	1443	2	2	0	0
vrp18	298	1443	2	2	0	0
vrp19	298	1443	8	8	0	0
vrp20	298	1443	8	8	0	0
vrp21	298	1443	18	18	1	0
vrp22	298	1443	18	18	0	0
vrp23	298	1443	18	18	0	0
vrp24	298	1443	8	8	0	0
vrp25	298	1443	8	8	0	0
vrp26	298	1443	7	7	0	0
vrp27	298	1443	7	7	0	0
vrp28	443	3101	16	16	0	0
vrp29	443	3101	16	16	0	0
vrp30	443	3101	16	16	0	0
vrp31	443	3101	16	16	0	0
vrp32	252	406	1	1	0	0
vrp33	252	406	1	1	0	0
vrp34	252	406	1	1	1	0
vrp35	443	3101	24	24	0	0
vrp36	443	3101	24	24	0	0
vrp37	443	3101	24	24	0	0
vrp38	19	278	4	2	1	0
vrp39	19	278	4	2	1	0
vrp40	19	278	2	1	1	0
vrp41	11	168	4	2	1	0
vrp42	8	140	3	2	1	0
vrp43	19	278	4	2	0	0
vrp44	61	310	1	1	1	56
vrp45	61	310	1	1	1	0
vrp46	61	310	1	1	1	56
vrp47	61	310	1	1	1	56
vrp48	61	310	1	1	1	56
vrp49	288	2033	6	4	1	70
vrp50	288	2033	6	4	1	70
vrp51	137	1426	6	3	1	147
vrp52	137	1426	6	3	1	147
vrp53	137	1420	6	3	1	0
vrp54	137	1113	7	3	1	315
vrp55	19	270	4	2	1	315
vrp56	19	270	2	1	1	0
vrp57	11	174	4	2	1	315
vrp58	8	146	3	2	1	315
vrp59	137	1113	7	3	1	0
vrp60	19	270	4	2	1	0
vrp61	556	4194	19	10	1	318
vrp62	135	1048	6	3	1	318
vrp63	559	4194	19	10	1	343
vrp64	19	270	4	2	1	84
vrp65	19	270	4	2	1	84
vrp66	19	270	4	2	1	84
vrp67	19	270	4	2	1	91
vrp68	19	270	4	2	1	84
vrp69	137	1113	7	3	1	294

Table 1: Table of VRPP instances of intercity railway planning.

instance	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{A}_C\rvert$	$\lvert\mathcal{A}_S\rvert$	$\lvert\mathcal{A}_{\mathtt{reg}}\rvert$	$\mathfrak{v}$	gap in %	run time [sec]
vrp01	582086	0	152914	58854	39	2.96	179670.64
vrp02	563309	0	134137	58854	39	2.85	179643.36
vrp03	574530	0	145358	58854	39	3.22	179646.88
vrp04	429172	0	0	58854	39	0.87	340.76
vrp05	878234	0	0	121339	53	0.75	858.88
vrp06	1105810	0	227576	121339	53	1.35	179636.74
vrp07	563309	0	134137	58854	39	1.43	179677.37
vrp08	80149	0	0	11245	16	0.17	56.29
vrp09	103308	0	23159	11245	16	0.77	249.47
vrp10	130222	0	21007	15249	17	0.32	236.20
vrp11	130258	0	20778	15290	17	0.99	332.66
vrp12	2264050	32794	576566	118074	104	0.72	10114.59
vrp13	10706858	0	0	1614334	117	0.67	21429.34
vrp14	9092524	0	0	0	117	0.65	11332.22
vrp15	34414350	0	0	5191325	116	0.76	34712.70
vrp16	29223025	0	0	0	116	0.63	26242.76
vrp17	4327786	0	0	634147	116	0.00	6032.71
vrp18	3693639	0	0	0	116	0.00	3936.92
vrp19	14016082	0	0	2059788	116	0.00	23229.05
vrp20	11956294	0	0	0	116	0.00	10569.47
vrp21	9843154	0	1768479	1217166	117	1.59	179602.63
vrp22	8078051	0	0	1217626	117	0.33	17438.61
vrp23	6860425	0	0	0	117	0.29	6723.69
vrp24	3932241	0	0	590485	117	0.30	4917.49
vrp25	3341756	0	0	0	117	0.01	3017.92
vrp26	3312613	0	0	486636	117	0.01	4434.07
vrp27	2825977	0	0	0	117	0.00	3335.84
vrp28	24996128	0	0	3124516	187	0.47	11560.39
vrp29	21871612	0	0	0	0	0.00	231.70
vrp30	10314680	0	0	1289335	190	0.00	3493.52
vrp31	9025345	0	0	0	0	0.00	116.24
vrp32	167231	0	0	8434	127	0.00	314.45
vrp33	158797	0	0	0	127	0.00	276.71
vrp34	240130	0	73000	8432	127	0.52	1141.47
vrp35	52823198	0	0	0	189	1.13	179602.12
vrp36	24278350	0	0	3498899	192	0.86	29996.07
vrp37	20779451	0	0	0	191	0.55	20786.69
vrp38	63639	1043	16082	2237	13	0.39	57.11
vrp39	63639	1043	16082	2237	13	0.29	51.74
vrp40	122158	1260	31163	2268	13	0.85	92.50
vrp41	22150	329	5462	669	9	0.87	35.36
vrp42	17301	329	5182	393	7	0.30	34.57
vrp43	47733	1043	0	2237	14	0.04	29.49
vrp44	130222	0	21007	15249	17	0.78	83751.80
vrp45	129988	0	20677	15266	17	0.36	220.39
vrp46	129988	0	20677	15266	17	0.89	51523.46
vrp47	130222	0	21007	15249	17	0.49	23812.34
vrp48	129988	0	20677	15266	17	0.45	22052.07
vrp49	2258455	32518	575472	117978	105	3.44	179571.16
vrp50	2258455	32518	575472	117978	105	5.14	179623.26
vrp51	1864911	22806	434878	67647	66	1.72	179550.38
vrp52	1864911	22806	434878	67647	66	6.05	179548.87
vrp53	1865198	22807	435163	67646	66	0.75	10444.74
vrp54	1446648	22621	307601	67398	64	5.69	179565.14
vrp55	52180	861	11793	2209	13	0.29	48.61
vrp56	94924	1042	22421	2234	13	0.29	84.98
vrp57	19491	261	4582	659	8	0.35	27.31
vrp58	15171	261	4369	383	7	0.39	35.45
vrp59	1465441	22804	321943	67570	65	0.79	7044.00
vrp60	52180	861	11793	2209	13	0.29	52.20
vrp61	6273258	178259	1647562	243456	230	7.18	179617.94
vrp62	2486160	119495	743222	51632	60	1.04	179541.27
vrp63	6195808	179713	1560894	244117	228	9.24	179598.80
vrp64	51882	851	11708	2198	13	0.39	49.64
vrp65	51882	851	11708	2198	13	0.39	46.48
vrp66	51882	851	11708	2198	13	0.39	58.21
vrp67	51882	851	11708	2198	13	0.39	52.91
vrp68	51882	851	11708	2198	13	0.39	53.27
vrp69	1446381	22611	307591	67377	64	2.09	179545.53

Table 2: Computational results for VRPP instances.