Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

RALF BORNDÖRFER CARLOS CARDONHA

A Set Partitioning Approach to Shunting

The work of the second author is supported by CNPq-Brazil.

ZIB-Report 09-18 (April 2010)

A Set Partitioning Approach to Shunting

Ralf Borndörfer* Carlos Cardonha*§

Abstract

The VEHICLE POSITIONING PROBLEM (VPP), also know as the shunting problem, is a classical combinatorial optimization problem in public transport planning. It has been investigated using several models and approaches, which work well for small instances, but not for large ones. We propose in this article a novel set partitioning model and an associated column generation approach for the VPP and for a multi-period generalization. The model provides a tight linear description of the problem and can, in particular, produce non-trivial lower bounds. The pricing problem, and hence the LP relaxation itself, can be solved in polynomial resp. pseudo-polynomial time, for some versions of the problem. Computational results for large-scale instances are reported.

1 Introduction

The VEHICLE POSITIONING PROBLEM (VPP) is about the parking of vehicles (buses, trams, or trains) in a depot. The aim is to organize the parking in such a way that the pull-in operations in the evening and the pull-out operations on the subsequent morning can be done without shunting movements. The problem is that the parking positions are organized in tracks, which work as one- or two-sided stacks or queues. If at some point in time a required vehicle is not in the front of a track, *shunting* movements must be performed. Their number must be minimized. We also consider a multi-periodic generalization of the VPP. The p-PERIODIC VEHICLE POSITIONING PROBLEM (VPP^p) consists of planning a cyclic sequence of p successive pull-in-pull-out periods in order to model the depot activity of an entire week. This is important in practice, because many vehicles that pull-in on Friday evening will not move until Monday morning.

The VPP and its variants, such as the BUS DISPATCHING PROBLEM, the TRAM DISPATCHING PROBLEM, and the TRAIN UNIT DISPATCHING PROBLEM, are well-investigated in the combinatorial optimization literature, see Hansmann and Zimmermann [8] for a comprehensive survey. The problem was introduced by Winter [13] and Winter and Zimmerman [14]. They

*Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany; Email borndoerfer@zib.de, cardonha@zib.de

§The work of this author is supported by CNPq-Brazil.

modeled the VPP with two-index variables as a Quadratic Assignment Problem (QAP) and used linearization techniques to solve it as an integer linear program. Hamdouni, Soumis and Desaulniers [9] extend their work exploring robustness and introducing the concept of uniform tracks to solve larger problems. Gallo and Di Miele [7] propose a three-index formulation and extend the problem to deal with vehicles of different lengths and interlaced sequences of arrivals and departures. Freling, Kroon, Lentink, and Huisman [6] and Kroon, Lentink, and Schrijver [12] improved this model by a new formulation of shunting constraints involving additional binary variables. They also consider decomposable vehicles (trains) and different types of tracks (the number of uniform tracks is assumed to be known in advance). Lentink [11] continues by a heuristic and a column generation algorithm based on a decomposition strategy for the problem. Recently, Borndörfer and Cardonha [3] combined the original binary quadratic programming model of Winter [13] with the model improvement of Kroon, Lentink, and Schrijver [12] in order to derive the first non-trivial lower bounds for the number of shunting movements. However, all of the mentioned approaches work only satisfactorily for specifically structured or for very small instances, and, in particular, not for an integrated treatment of multi-period problems.

We propose in this paper novel set partitioning models for the VPP and the VPP^p as bases for a column generation solution approach. We show that the associated pricing problem, and hence the entire LP relaxation, can be solved in polynomial resp. pseudo-polynomial time for some versions of the problem, namely, if a certain objective function that we call "first crossings" is minimized. Our computational results show that this approach can produce non-trivial lower bounds. In this way, large-scale instances can be solved.

An approach similar, but not identical, to ours was used by Diepen [5] in the context of airport gate assignments; this problem involves interlaced sequences of arrivals and departures, which are already matched, and stacks of size one. A set partitioning approach for a shunting problem involving matched arrivals and departures was also considered by Lentink [11].

2 The Vehicle Positioning Problem

The Vehicle Positioning Problem (VPP) is a *3-dimensional matching problem* that can be described as follows. Some number n of vehicles *arrive* in a sequence $\mathcal{A} = \{a_1, \ldots, a_n\}$, $a_1 < \cdots < a_n$. They must be assigned to *parking positions* in some number $m \leq n$ of parking *tracks* $\mathcal{S} = \{s_1, \ldots, s_m\}$ in a depot; from these tracks, the vehicles *depart* to service a sequence of timetabled trips $\mathcal{D} = \{d_1, \ldots, d_n\}$, $d_1 < \cdots < d_n$. Denote by (a, s, d) the *assignment* of the arriving vehicle a to a parking position in track s in order to service the departing trip d. Note that the parking position in track s is determined implicitly by the assignments involving the preceding vehicle arrivals.

Assignments are restricted by a number of constraints. We consider t vehicle types $\mathcal{T} = \{T_1, \ldots, T_t\}$ with t_i vehicles of type T_i, $i = 1, \ldots, t$. Each arriving vehicle a has a *type* $t(a)$, and each departing trip d has a type $t(d)$. Departure d can only be serviced by vehicles of type $t(d)$. The assignment (a, s, d) is feasible if $t(a) = t(d)$; we will henceforth only consider feasible assignments. Each arriving vehicle a also has a *size* (or length) $l(a)$, and each track $s \in \mathcal{S}$ has size β. We assume that $\beta|S| \geq \sum_{a\in\mathcal{A}} l(a) + |S| \max_{a\in\mathcal{A}} l(a)$, i.e., there is enough space to park all vehicles. Note that in this version of the problem, the tracks are all identical. If all vehicle types have the same size, which we also assume, we have $l = 1$ and $\beta|S| \geq n$, i.e., β is the number of parking positions in each track. Furthermore, we assume that the first departure trip starts after the last arrival of an incoming vehicle, and that each track is operated as a FIFO stack, that is, vehicles enter the track at one end and leave at the other. Consider assignments (a, s, d) and (a', s, d'), whose parking positions are located in the same track s; then a *shunting* movement is required if either $a < a'$ and $d > d'$ or if $a' < a$ and $d' > d$. In this case, we say that these assignments are in *conflict* and denote the associated *crossings* by $(a, s, d) \dagger (a', s, d')$ or $(a, d) \dagger (a', d')$. A crossing of two assignments $(a, s, d) \dagger (a', s, d')$ such that a' is parked immediately after a (i.e., all arrivals between a and a' are assigned to tracks different from s) is called a *first crossing*. Winter [13] showed that any solution involving crossings also has first crossings. Clearly, the number of crossings is in general larger than or equal to the number of first crossings.

A *configuration* q is a set of assignments (a_j, s, d_j), $j = 1, \ldots, r$, to some track s, such that $a_j < a_{j+1}$, $j = 1, \ldots, r-1$, i.e., the parking positions in the track are filled consecutively, and $\sum_{j=1}^{r} l(a_j) \leq \beta$, i.e., the track is big enough for all assigned parkings. When the meaning is clear from the context, we write $a \in q$, $d \in q$, and $(a, d) \in q$ if $(a, s, d) \in q$. In other words, a configuration records all parking assignments for an entire track. Let $\mathcal{Q}_s$ denote the set of all configurations for track s; let us write $s \in q$ for $q \in \mathcal{Q}_s$. Let finally $\mathcal{Q} = \bigcup_{s\in\mathcal{S}}$ denote the set of all configurations.

A *matching* is a set of configurations q_s, one for each track $s \in \mathcal{S}$, such that each arriving vehicle and each departing trip is assigned to a parking position in exactly one track (or configuration). Then the *vehicle positioning problem* is to find a matching that minimizes the total number of crossings; in fact, we focus on the case that minimizes the number of first crossings. Note that it is equivalent to find a matching without crossings and without first crossing.

The Multi-Periodic Vehicle Positioning Problem (VPP^p) involves a cyclic sequence of p single period instances of the VPP, i.e., the departures d_j^h of period h define the arrivals a_i^{h+1} in period $h+1$, where period indices are taken modulo p. Denote the arrival sequence for period h by $\mathcal{A}^h = \{a_1^h, \ldots, a_n^h\}$ and the departure sequence by $\mathcal{D}^h = \{d_1^h, \ldots, d_n^h\}$; then $\mathcal{D}^h = \mathcal{A}^{h+1}$, $h = 1, \ldots, p$. The last wp periods of the sequence form "the weekend". Typically, the number wn of timetabled trips on weekend periods is smaller than the number n of trips on regular (weekday) periods. We therefore consider dummy arrivals a_i^h, $h = p - wp + 1, \ldots, p$, $i = wn + 1, \ldots, n$, and dummy departures d_j^h, $h = p - wp, \ldots, p$, $j = wn + 1, \ldots, n$, such that $d_j^h = a_j^{h+1}$, $h = p - wp, \ldots, p$, $j = wn + 1, \ldots, n$, that indicate parking during the weekend. We also stipulate that only a subset of wm tracks can be used for arrivals and departures, and that the idle $n - wn$ vehicles stay parked on the remaining $m - wm$ tracks on the weekend, i.e., the only legal assignments for the *weekend parking tracks* $s = wm + 1, \ldots, m$ in the weekend periods $h = p - wp + 1, \ldots, p$ are of the form (a^h, s, d^h), where $a^h = d^h$ is a weekend arrival/departure. We denote the set of configurations for a track s in period p by $\mathcal{Q}_s^p$. The remaining concepts are the same as for the single period VPP.

We remark that a sequential solution of the VPP^p by a subsequent treatment of successive single period subproblems might not yield good solutions, in particular, if myopic choices on "Friday" do not take the upcoming "Monday" into account.

The computational complexity of the VPP was analyzed by Winter in [13]. Some of his results can be extended to the VPP^p as well.

Theorem 1. *(Winter) It is possible to decide in polynomial time if an instance of* VPP *needs shuntings when all tracks have size two and the assignment of vehicles to trips is fixed.*

Corollary 1. *It is possible to decide in polynomial time if an instance of* VPP^p *needs shuntings when all tracks have size two and the assignment of vehicles to trips is fixed.*

Proof. The problem can be reduced to a bipartite matching problem similar as in [13]. The only difficulty is with the arrivals on the last weekday period and the departures on the first weekday period. These assignments can also be included in a bipartite matching problem, where arrivals on the last weekday period can be assigned not only to the departures on the last weekday period, but also to departures on the first weekday period. □

Theorem 2. *(Winter) Deciding if an instance of* VPP *needs shuntings is* $\mathcal{NP}$*-complete.*

Corollary 2. *Deciding if an instance of* VPP^p *needs shuntings is* $\mathcal{NP}$*-complete.*

Proof. Let I be an instance of VPP with arrival sequence $\mathcal{A}$, departure sequence $\mathcal{D}$ and m tracks. We create an instance I' of VPP^p with at least 3 weekday periods. The second weekday period is such that $\mathcal{A}^2 = \mathcal{A}$, $\mathcal{D}^2 = \mathcal{D}$, and the number of tracks is m (i.e., the second weekday period is the single period instance I).

The other periods are such that both the arrival and the departure sequences are either equal to $\mathcal{A}$ (in the case of the weekday periods) or equal to the sequence formed by the first wn elements of $\mathcal{A}$ (in the case of the weekend periods). It is clear that there is a crossings-free solution for these periods.

On the other hand, the second weekday period is non-trivial and its solution is independent of the matchings for the other periods. Consequently, we obtain an optimal solution for I' from an optimal solution for I and vice-versa. Therefore, if it is possible to decide in polynomial time if an instance of VPP^p needs shuntings, than it is possible to do the same for VPP. □

Theorem 3. *(Winter) Optimizing* VPP *is* $\mathcal{NP}$*-hard.*

Corollary 3. *Optimizing* VPP^p *is* $\mathcal{NP}$*-hard.*

Proof. The proof is similar to the one presented for Corollary 2. □

3 Integer Programming Models

The following IP formulation (**L**) of the VPP is based on the model proposed by Lentink, Kroon and Schrijver [12]. It yields the best performance among the models of the literature. The formulation uses binary variables $x_{a,s,d}$ to indicate the assignment of arrival a to track s and departure d and binary variables $r_{a,s,d}$ to indicate a conflict on track s involving assignments that contain arrival a and departure d. Equations (**L**) (i) assure that each arriving vehicle a is assigned to exactly one configuration, (**L**) (ii) is analogous for departing trips d, (**L**) (iii) controls the size restriction of track s, and (**L**) (iv) indicates if there is a conflict between the assignment of a and the assignment of d on track s. The objective function counts the total number of crossings. The model reads as follows:

$$
\begin{array}{llll}
(\mathbf{L}) & \min & \displaystyle\sum_{(a,s,d)\in\mathcal{A}\times\mathcal{S}\times\mathcal{D}} r_{a,s,d} & \\
\text{(i)} & & \displaystyle\sum_{(s,d)\in\mathcal{S}\times\mathcal{D}} x_{a,s,d} = 1 & \forall a \in \mathcal{A} \\
\text{(ii)} & & \displaystyle\sum_{(a,s)\in\mathcal{A}\times\mathcal{S}} x_{a,s,d} = 1 & \forall d \in \mathcal{D} \\
\text{(iii)} & & \displaystyle\sum_{(a,d)\in\mathcal{A}\times\mathcal{D}} x_{a,s,d} \leq \beta & \forall s \in \mathcal{S} \\
\text{(iv)} & \displaystyle\sum_{a'<a} x_{a',s,d} + \sum_{d'\leq d} x_{a,s,d'} - r_{a,s,d} \leq 1 & & \forall (a,s,d) \in \mathcal{A}\times\mathcal{S}\times\mathcal{D} \\
 & & x_{a,s,d}, r_{a,s,d} \in \{0,1\}. &
\end{array}
$$

We propose the following *set partitioning model* for the VPP:

$$
\begin{array}{llll}
(\mathbf{X}) & \min & \sum_{q\in\mathcal{Q}} c_q x_q & \\
(\text{i}) & & \sum_{a\in q\in\mathcal{Q}} x_q = 1 & \forall a \in \mathcal{A} \\
(\text{ii}) & & \sum_{d\in q\in\mathcal{Q}} x_q = 1 & \forall d \in \mathcal{D} \\
(\text{iii}) & & \sum_{s\in q\in\mathcal{Q}} x_q = 1 & \forall s \in \mathcal{S} \\
& & x_q \in \{0,1\} & \forall q \in \mathcal{Q}.
\end{array}
$$

This model employs binary variables x_q, $q \in \mathcal{Q}$, to indicate the use of track configurations. Equations ($\mathbf{X}$) (i) assure that each arriving vehicle a is assigned to exactly one configuration, ($\mathbf{X}$) (ii) is analogous for departing trips d, and ($\mathbf{X}$) (iii) allows exactly one configuration for each track (which can be empty). The objective sums up the number of crossings or first crossings, depending on the definition of c.

Recall that we consider in this article a version of the VPP with identical tracks. We can therefore improve model ($\mathbf{X}$) by working with generic configurations. Replacing ($\mathbf{X}$) (iii) by the single inequality

$$
(\text{iii}') \qquad \sum_{q\in Q} x_q \le m,
$$

produces an equivalent formulation ($\mathbf{X}'$). Doing so removes a significant amount of symmetry from the model. In fact, the number of variables is reduced by a factor of m, and the number of solutions by a factor of $m!$. Such symmetries are one of the reasons why the hitherto proposed models do not work well computationally.

Recall also that we assume that all vehicles have size 1, such that at most β of them can be assigned to a single track. Then there are (at most) $O(n^{2\beta})$ different sets of arrivals and departures that can be assigned to a track, and from each set it is possible to generate $O(\beta!)$ different arrival and departure sequences. Consequently, there are $O(sn^{2\beta}\beta!) = O(nn^{2\beta}\beta^\beta) = O(n^{3\beta+1})$ possible stack configurations.

Proposition 1. *If all vehicles have unit size, model $\boldsymbol{X}$ has $O(n)$ constraints and $O(n^{3\beta+1})$ variables.*

Formulation $\mathbf{X}$ has also some appealing theoretical properties. Interpreting the VPP as a partitioning problem for configurations, i.e., as a *combinatorial packing problem*, see Borndörfer [2], it follows:

Theorem 4. *The intersection graph associated with formulation ($\boldsymbol{X}$) for a* VPP *on two tracks is perfect.*

This means that the 2-track case of formulation ($\mathbf{X}$) can be solved by a cutting plane algorithm separating only clique constraints.

For the VPP^p, we propose the following similar *set partitioning model*:

$$
\begin{array}{llll}
(\mathbf{X}^p\ ) & \min & \sum\limits_{q\in\mathcal{Q}} c_q x_q^h & \\
\text{(i)} & & \sum\limits_{a\in q\in\mathcal{Q}^h} x_q^h = 1 & \forall h, a \in \mathcal{A}^h \\
\text{(ii)} & & \sum\limits_{d^h\in q\in\mathcal{Q}^h} x_q^h = 1 & \forall h, d \in \mathcal{D}^h \\
\text{(iii)} & & \sum\limits_{s^h\in q\in\mathcal{Q}^h} x_q^h = 1 & \forall h, s \in \mathcal{S}^h \\
 & & x_q^h \in \{0,1\} & \forall q \in \mathcal{Q}^h.
\end{array}
$$

Model $\mathbf{X}^p$ allows the same use of generic configurations as model $\mathbf{X}$. Furthermore, the configuration variables for the weekend parking tracks can be identified. This means that we can replace the inequalities ($\mathbf{X}^p$) (iii) by the following inequalities

$$
\begin{array}{lll}
(\text{iii}^a) & \sum\limits_{q\in Q^h} x_q \le m & \forall \text{ weekday periods } h \\
(\text{iii}^b) & \sum\limits_{q\in Q^h} x_q \le wm & \forall \text{ weekend periods } h,
\end{array}
$$

to obtain an equivalent formulation ($\mathbf{X}^{p\,\prime}$). Again, this trick removes symmetry from the model, improving the computational performance.

The number of possible stack configurations can be calculated independently for each period. Of course, vehicles that were parked on the first weekend period come back into use on the first weekday period. We therefore have $O(n^{3\beta+1})$ possible stack configurations on all periods except the first and the last weekday period, which should be considered simultaneously and have $O(2^{2\beta}n^{3\beta+1})$ possible stack configurations.

Proposition 2. *If all vehicles have unit size, model $\boldsymbol{X}^p$ has $O(pn)$ constraints and $O(pn^{3\beta+1} + 2^{2\beta}n^{3\beta+1})$ variables.*

We now compare the strengths of the linear relaxations of models $\mathbf{L}$ and $\mathbf{X}$. Denote by $V_{LP}(\mathbf{F})$ the optimal objective value of the LP relaxation of some integer programming formulation $\mathbf{F}$, and by $P_{LR}(\mathbf{F})$ the polytope associated with the LP relaxation of formulation $\mathbf{F}$.

Formulation $\mathbf{L}$ is the most successful model for VPP described in the literature so far. However, Borndörfer and Cardonha [3] have shown that its LP relaxation is weak:

Theorem 5. $V_{LP}(\boldsymbol{L}) = 0$ *if* $m > 1$.

The following theorem shows that $\mathbf{X}$ is stronger than $\mathbf{L}$:

Theorem 6. $P_{LR}(\boldsymbol{X}) \subseteq P_{LR}(\boldsymbol{L})$.

Proof. Let x_q be a solution of the linear relaxation of an instance I of VPP. We set the values of $(x_{a,s,d}, r_{a,s,d})$ for each $(a,s,d) \in \mathcal{A} \times \mathcal{S} \times \mathcal{D}$ as follows:

$$x_{a,s,d} = \sum_{(a,d)\in q, q\in \mathcal{Q}_s} x_q$$
$$r_{a,s,d} = \sum_{\substack{(a,d'),(a',d)\in q \\ (a,d')\dagger(a',d) \\ q\in\mathcal{Q}_s}} x_q.$$

We show that $(x_{a,s,d}, r_{a,s,d})$ belongs to $P_{LR}(\mathbf{X})$.

$$\text{(i)} \quad \textstyle\sum_{sd} x_{a,s,d} = \sum_{sd}\sum_{(a,d)\in q,q\in\mathcal{Q}_s} x_q = \sum_{sd,(a,d)\in q,q\in\mathcal{Q}_s} x_q = 1 \qquad \forall a \in \mathcal{A}$$

$$\text{(ii)} \quad \textstyle\sum_{as} x_{a,s,d} = \sum_{as}\sum_{(a,d)\in q,q\in\mathcal{Q}_s} x_q = \sum_{as,(a,d)\in q,q\in\mathcal{Q}_s} x_q = 1 \qquad \forall d \in \mathcal{D}$$

$$\text{(iii)} \quad \textstyle\sum_{ad} x_{a,s,d} = \sum_{q\in\mathcal{Q}_s} x_q|q| \leq \sum_{q\in\mathcal{Q}_s} x_q\beta \leq \beta \qquad \forall s \in \mathcal{S}$$

$$\begin{aligned}
\text{(iv)} \quad & \textstyle\sum_{a'<a} x_{a',s,d} + \sum_{d'\leq d} x_{a,s,d'} - 1 \\
\leq & \sum_{\substack{(a',d),(a,d')\in q \\ (a,d')\dagger(a',d) \\ q\in\mathcal{Q}_s}} 2x_q + \textstyle\sum_{(a',d)\vee(a,d')\notin q, q\in\mathcal{Q}_s} x_q - 1 \\
= & \; 2\sum_{\substack{(a',d),(a,d')\in q \\ (a,d')\dagger(a',d) \\ q\in\mathcal{Q}_s}} x_q - \sum_{\substack{(a',d),(a,d')\in q \\ (a,d')\dagger(a',d) \\ q\in\mathcal{Q}_s}} x_q \\
= & \sum_{\substack{(a',d),(a,d')\in q \\ (a,d')\dagger(a',d) \\ q\in\mathcal{Q}_s}} x_q = r_{a,s,d} \qquad \forall (a,s,d) \in \mathcal{A}\times\mathcal{S}\times\mathcal{D}
\end{aligned}$$

The inequalities above show that the pair $(x_{a,s,d}, r_{a,s,d})$ is a solution for the linear relaxation of Lentink's model. ☐

Theorem 7. $V_{LR}(\boldsymbol{X}\ ) > 0$ *for some instances of* VPP *that require shuntings.*

Proof. We give a family of instances of VPP which have a pair (a,d) that must belong to every matching and that will cross in any feasible configuration. For such a pair, $c_q > 0$ for $(a,d) \in q$, and $\sum_{(a,d)\in q,q\in\mathcal{Q}} x_q = 1$.

One such family can be generated as follows. Let $\mathcal{A}$ and $\mathcal{D}$ be such that $t(a_1) = t(d_n)$, there is no other arrival a_i (departure d_j) with $i \neq 1$ ($j \neq n$) such that $t(a_1) = t(a_i)$ ($t(d_n) = t(d_j)$) and assume that the depot has n parking positions (i.e., the instance has a feasible solution). In such an instance, a_1 must be assigned to d_n, and every configuration q which contains this pair has cost $c_q > 0$. ☐

A similar proof yields the following corollary regarding VPP^p:

Corollary 4. *$V_{LR}(\boldsymbol{X}^p\ ) > 0$ for some instances of* VPP^p *that require shuntings.*

Theorem 7 and Corollary 4 show that the linear relaxations of $\mathbf{X}$ and $\mathbf{X}^p$ can yield nontrivial lower bounds for the problem. However, this is not the case in every instance of the problem that require shuntings. Both cases will come up in our computational results.

4 Column Generation

We propose to solve $\mathbf{X}$ and $\mathbf{X}^p$ using a column generation approach. In order to choose the variables that enter the master problem in each iteration, we must solve a *pricing* subproblem. It will turn out that the complexity of the pricing subproblem depends on the way shuntings are counted. The number of shuntings is often estimated in terms of the total *number of crossings*. We propose to minimize the *number of first crossings*, which is also a reasonable estimate, because it counts the occasions on which problems comes up in pratice.

4.1 Pricing First Crossings

The idea is to use a dynamic program, recording the arrival and departure assigned to the last parking position. The problem with this approach is that in the case of shuntings, book-keeping becomes necessary in order not to assign an arrival or departure twice. This difficulty can be addressed as follows. First, Winter [13] has shown that considering only configurations where the arriving vehicles do not produce crossings, i.e., with ascending arrival times, does not change the minimal number of crossings. His proof can be easily modified in order to show that this also holds when first-crossings must be minimized. Note we have already defined the possible configurations $\mathcal{Q}$ and $\mathcal{Q}^p$ in this way. Second, allowing several assignments to cover a departure does also not hurt, but, as we will see in a minute, makes the pricing problem easier. We therefore consider relaxations ($\mathbf{X}''$) and ($\mathbf{X}^{p\prime\prime}$) of the formulations ($\mathbf{X}'$) and ($\mathbf{X}^{p\prime}$), that extend $\mathcal{Q}$ and $\mathcal{Q}^p$ to sets $\mathcal{Q}''$ and $\mathcal{Q}^{p\prime\prime}$ that include such configurations, and which, in addition, relax constraints ($\mathbf{X}'$) (ii) and ($\mathbf{X}^{p\prime}$) (ii) to

$$\text{(ii}'\text{)} \qquad \sum_{d \in q \in \mathcal{Q}''} x_q \geq 1 \quad \forall d \in \mathcal{D}$$

$$\text{(ii}'\text{)} \qquad \sum_{d \in q \in \mathcal{Q}^{h\prime}} x_h^p \geq 1 \quad \forall h, d \in \mathcal{D},$$

respectively. This construction is similar to the so-called q-path relaxation that is popular in vehicle routing, see Baldacci, Toth, and Vigo [1].

Proposition 3. $0 \leq V_{LR}(\boldsymbol{X}'') \leq V_{LR}(\boldsymbol{X})$ *and* $0 \leq V_{LR}(\boldsymbol{X}^{p\prime\prime}) \leq V_{LR}(\boldsymbol{X}^p)$*.*

Theorem 8. *The pricing problem for the LP relaxation of model* $\boldsymbol{X}''$ *can be solved in pseudo-polynomial time* $O(mn^4\beta)$ *if first crossings are minimized.*

Proof. The pricing problem can be solved by dynamic programming. For a fixed track s, consider a state space $\mathcal{H}$ indexed by tuples in $\mathcal{A} \times \mathcal{D} \times [0, \beta]$. Let C be a matrix indexed by such tuples (a, d, k), such that entry $c[a][d][k]$ holds the maximum reduced cost of a configuration of size k, that terminates by assigning arrival a to departure d. Let us denote the dual variables associated with constraints ($\mathbf{X}''$) (i), (ii$'$), and (iii$'$) by σ_a, ω_d, and π, respectively. Then the recursion formula for the entries of C is as follows:

$$c[a'][d'][k'] = \mathtt{max}_{a<a',d\neq d',k=k'-l(a)} \left\{c[a][d][k] + \sigma_a + \omega_d + \alpha_{ad}\right\},$$

where $\alpha_{kd} = 1$, if $d < d'$, and 0 else. The initialization is $c[a][d][0] := \pi$. The recursion takes time $O(mn^4\beta)$. ☐

Corollary 5. *The pricing problem for the LP relaxation of model* $\boldsymbol{X}^{p\prime\prime}$ *can be solved in pseudo-polynomial time* $O((p+1)mn^4\beta)$ *if first crossings are minimized.*

Proof. The multi-period pricing problem is similar to the single period pricing problem. Basically, the dynamic program in the proof of Theorem 8 must be applied for each period. In addition, we must consider possible assignments of arrivals on the last weekday period to departures on the first weekday period. Consequently, we execute the pricing routine $\mathbf{X}''$ $(p + 1)$ times, which leads to an $O((p + 1)mn^4\beta)$ pseudo-polynomial time algorithm. ☐

Recall that the vehicles have unit size, i.e., $\beta = O(n)$. Then:

Corollary 6. *The pricing problem for the LP relaxation of model* $\boldsymbol{X}''$ *can be solved in polynomial time* $O(mn^5)$*, if first crossings are minimized and vehicles have unit size.*

Corollary 7. *The pricing problem for the LP relaxation of model* $\boldsymbol{X}^{p\prime\prime}$ *can be solved in polynomial time* $O((p+1)mn^5)$*, if first crossings are minimized and vehicles have unit size.*

4.2 Generating columns and uniform tracks

In order to speed up the convergence of our column generation algorithm, we initialize the column pool by a promising set of configurations. Namely, we use a greedy procedure to construct a set of configurations with a hopefully small number of crossings. This procedure is inspired by the concept of *uniform tracks* of Hamdouni, Soumis and Desaulniers [9]. Uniform tracks are tracks that receive just one type of vehicle (and therefore do not have crossings). Solutions with uniform tracks are more robust (as pointed out by [9]).

We first construct a lower bound on the number of *non-uniform* tracks in a feasible solution for an instance of VPP (or for one period in an instance of VPP^p).

Proposition 4. *If all tracks have the same size, there must be at least*

$$B_{NU} := \frac{\sum_{T_i \in \mathcal{T}} t_i \mod \beta}{\beta}$$

non-uniform tracks in any solution of VPP.

Proof. It is not possible to assign more than $t_i \mod \beta$ vehicles of type T_i to uniform tracks. □

Based on this lower bound, we develop an algorithm that assigns $t_i \mod \beta$ arrivals and departures of type T_i to B_{NU} tracks. By assigning the remaining arrivals and departures to uniform tracks, we obtain a feasible matching for one period. Clearly, this algorithm is does in general not produce optimal solutions.

Theorem 9. *There are instances of* VPP *for which there is no optimal solution that uses only B_{NU} non-uniform tracks.*

Proof. We give a family of instances of VPP for which any optimal solution contains $B_{NU} + 1$ non-uniform tracks. Let $t \geq 2$, with $t_1 = \beta - 1$, $t_2 = \beta + 1$, and $t_i = \beta$, $2 < i \leq t$. Assume that the first 2β arrivals and the last 2β departures are of type T_1 or T_2. Let $t(d_{n-2\beta+1}) = t(a_{2\beta}) = T_1$, and assume that arrivals and departures of type T_1 appear distributed in subsequences with the following property: if the subsequence containing $d_{n-2\beta+1}$ has size k, the first subsequence of type T_1 in $\mathcal{A}$ contains $k + k'$ arrivals, with $k' > 0$. Similarly, the second subsequence of type T_1 in $\mathcal{D}$ contains $k' + k''$ departures, with $k'' > 0$. We repeat this procedure to determine the position in the sequences of all arrivals and departures of type T_1 (and, consequently, T_2 as well). Figure 1 shows an example for $\beta = 4$ and $n = 8$.

Clearly, any assignment involving the remaining $n - 2\beta$ arrivals and departures crosses with any assignment involving pairs of type T_1 or T_2. Besides, elements of type T_i, $i > 2$, can be assigned to uniform tracks.

Consequently, instances of this family can only have crossing-free solutions if pairs of type T_1 are assigned to non-uniform tracks with pairs of type T_2. □

Finally, we observe that this heuristic can be performed in polynomial time.

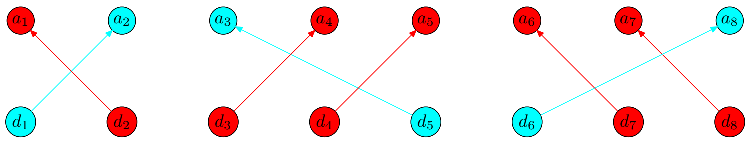

Figure 1: Example for an instance with 0 crossings only for solutions with non-uniform tracks.

5 Computational Results

This section presents the results of a computational evaluation of our column generation approaches. All computations were done on an 64-bits Intel(R) Core(TM)2 Quad with 2.83 GHz, 8 GB of RAM memory, running openSuse Linux 11.2. Our code is implemented in `C++` and was compiled using `g++` 4.4.1 and with the callable library of CPLEX 12.1.0 [4].

Our code solves formulations ($\mathbf{X}''$) and ($\mathbf{X}^{p}{}''$) to minimize first crossings. We initialize the column pool using the greedy heuristics of Section 4.2. In each iteration of the column generation algorithm, the pricing routine of Section 4.1 is called and the minimal reduced cost configuration is added. In addition, the greedy heuristic is called to complete this configuration to a matching; the configurations of this matching are also added. Then the LP is resolved. This procedure is repeated until no more improving configuration exists, i.e., until the LP is solved to optimality. After that, the column pool is passed to the IP solver of CPLEX in order to compute a (heuristic) integer solution.

Table 1 reports computational results for instances of the VPP. The names $n - m - t$ of the instances indicate the number n of arrivals and departures, the number m of tracks, and the number t of vehicle types. The columns list the number Q_{LP} of configurations priced, its value V_{LR}, the value V_{IP} of the heuristic integer solution, and the CPU time in seconds to solve the LP relaxation. It can be seen that the number of shuntings is very low and often zero for the larger problems, which have a size that is typical in practice. Smaller problems can enforce shuntings, in particular, if many vehicle types are involved. Such instances can be computationally difficult, featuring large LP/IP gaps.

Table 2 reports computational results for the multi-period VPP^p. The names $nPer - wPer - n - m - wn - t$ of the instances indicate the number $nPer$ of weekday periods, the number $wPer$ of weekend periods, the number n of arrivals on weekday periods, the number m of tracks, the number wn of weekend periods, and the number t of vehicle types. The columns are the same as in Table 2. Note that the number of periods for a real-world instance would be 14, with a morning and evening period for every day of the week. In other words, the instances named `10-4-*-*-*-*` correspond

Instance	Q_{LP}	V_{LR}	V_{IP}	Time (sec)
20-4-12	101	2	2	0
40-5-15	1012	0.2	3	6
63-7-8	37	0	0	1
64-8-12	257	0	0	1
96-12-7	12	0	0	1
150-15-6	15	0	0	3
150-15-15	740	0	1	82
160-20-10	20	0	0	2
160-20-40	694	0	0	14
200-20-10	20	0	0	0

Table 1: Solving problem VPP using formulation $\mathbf{X}''$.

Instance	Q_{LP}	V_{LR}	V_{IP}	Time (sec)
4-2-20-4-15-5	976	1	3	1
5-2-36-6-24-5	5682	0	1	4
6-3-30-5-18-5	4947	1	5	3
10-3-64-8-48-6	12942	0	3	36
10-4-80-10-56-8	25454	0	5	80
10-4-120-12-90-8	80314	0	5	1166
10-4-150-15-110-9	20217	0	1	129
10-4-150-15-100-10	50122	0	4	861

Table 2: Solving problem VPP^p using formulation $\mathbf{X}^{p\prime\prime}$

to a typical "standard week". As far as we know, instances of this size and complexity have not been considered or solved in the VPP literature before. The multi-period instances are much larger and substantially more difficult than their single-period counterparts; often, we could not eliminate shuntings completely. Nevertheless, the number of shuntings is still very low and the time consumption of the algorithm is still reasonable. It is therefore possible to solve even difficult multi-period vehicle positioning problems with very good results.

6 Conclusions

We presented in this article novel set partitioning models for the VPP and for the VPP^p, which are suitable for a column generation solution approach. These models have better theoretical properties than the models that have been used previously. For the minimization of first crossings, the associated pricing problems and hence the entire LP relaxation can be solved in polynomial time (if all vehicles have unit size). In this way, large scale multi-period instances can be solved with good quality and in reasonable time.

References

[1] Baldacci, R., and Toth, P., and Vigo, D., *Recent advances in vehicle routing exact algorithms*, Springer Berlin/Heidelberg, 4OR: A Quarterly Journal of Operations Research, vol. 5 (2007), Number 4, 269-298.

[2] Borndörfer, R., *Combinatorial packing problems*, "The Sharpest Cut - The Impact of Manfred Padberg and His Work", Martin Grötschel (Ed.), MPS-SIAM, vol. 4 (2004), Series on Optimization, 19–32.

[3] Borndörfer, R., and Cardonha, C. *A Binary Quadratic Programming Approach to the Vehicle Positioning Problem*, ZIB Technical Report 09-12, (2009).

[4] ILOG, CPLEX Website,http://www.ilog.com/products/cplex/.

[5] Diepen, G., and van den Akker, J.M., and Hoogeveen, J.A., *Integrated Gate and Bus Assignment at Amsterdam Airport Schiphol*, Robust and Online Large-Scale Optimization (2009), Lecture Notes in Computer Science, 338–353.

[6] Freling, R., and Lentink, R., and Kroon, L., and Huisman, D., "Shunting of passenger train units in a railway station", ERIM Report Series Research in Management, 2002.

[7] Gallo, G., and Di Miele, F., *Dispatching buses in parking depots*, Transportation Science, 35 (2001), 322–330.

[8] Hansmann, R. S. and Zimmermann, U. T., *Optimal Sorting of Rolling Stock at Hump Yards*, "Mathematics – Key Technology for the Future: Joint Projects Between Universities and Industry", Jäger, W. and Krebs, H.–J. (Ed), Springer Berlin, 2008, 189–203.

[9] Hamdouni, M., and Soumis, F., Desaulniers, G., "Dispatching Buses in a Depot Using Block Patterns", Transportation Science, 40, 3 (2006) , 364–377.

[10] Kaufmann, L., and Broeckx, F., *An Algorithm for the Quadratic Assignment Problem*, European J. Oper. Res., 2 (1978), 204–211.

[11] Lentink,R.M., "Algorithmic Decision Support for Shunt Planning", Ph.D. thesis, Erasmus Research Institute of Management, Erasmus University Rotterdam, 2006.

[12] Kroon, L., Lentink, R., Schrijver, A., "Shunting of passenger train units: an integrated approach", ERIM Report Series Reference No. ERS-2006-068-LIS, 2006, `http://ssrn.com/abstract=1317605`.

[13] Winter, T., "Online and Real-Time Dispatching Problems", Ph.D. thesis, TU Braunschweig, 1998.

[14] Winter, T., and Zimmermann, U., *Real-time dispatch of trams in storage yards*, Annals of Operations Research, 96 (2000), 287–315.