Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

Ralf Borndörfer, Carlos Cardonha

A Binary Quadratic Programming Approach to the Vehicle Positioning Problem

ZIB-Report 09-12 (March 2010)

A Binary Quadratic Programming Approach to the Vehicle Positioning Problem

Ralf Borndörfer[1] and Carlos Cardonha[2*]

[1] Zuse Institute Berlin borndoerfer@zib.de
[2] Zuse Institute Berlin cardonha@zib.de

Summary. The VEHICLE POSITIONING PROBLEM (VPP) is a classical combinatorial optimization problem that has a natural formulation as an integer quadratic program. This MIQCP is closely related to the QUADRATIC ASSIGNMENT PROBLEM and, as far as we know, has not received any attention yet. We show in this article that such a formulation has interesting theoretical properties. Its QP relaxation produces, in particular, the first known nontrivial lower bound on the number of shuntings. In our experiments, it also outperformed the integer linear models computationally. The strengthening technique that raises the lower bound might also be useful for other combinatorial optimization problems.

1 Introduction

The VEHICLE POSITIONING PROBLEM (VPP) is about the assignment of vehicles (buses, trams, or trains) to parking positions in a depot and to timetabled trips. 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 type of vehicle is not available in the front of any track, shunting movements must be performed in order to change the vehicle positions. This is undesirable and should be avoided.

The VPP and its variants, such as the BUS DISPATCHING PROBLEM (BDP), the TRAM DISPATCHING PROBLEM (TDP), and the TRAIN UNIT DISPATCHING PROBLEM (TUDP), are well-investigated in the combinatorial optimization literature. The problem was introduced by Winter [10] and Winter and Zimmerman [11], who modelled the VPP as a QUADRATIC ASSIGNMENT PROBLEM and used linearization techniques to solve it as an integer linear program. This approach was extended by Gallo and Di Miele [3] to deal with vehicles of different lengths and interlaced sequences of arrivals and departures. Similarly, Hamdouni *et al.* [4] explored robustness and the idea of uniform tracks (tracks which receive just one type of vehicle) to solve larger

* The work of this author is supported by CNPq-Brazil

problems. Recently, Freling, Kroon, Lentink, and Huisman [2] and Kroon, Lentink, and Schrijver [7] proposed an integer linear program to consider decomposable vehicles (trains) and different types of tracks; they assume that the number of uniform tracks is known in advance.

Although the VPP was originally modeled as a binary quadratic program, this formulation was not explored theoretically and it was not used for computational studies. All research efforts that we are aware of concentrated on integer linear models, that used more and more indices in order to produce tighter linearizations. Recent progress in mixed integer nonlinear programming (MINLP) and, in particular, in mixed integer quadratic constrained programming (MIQCP) methods [8], however, has increased the attractivity of the original quadratic model. Besides the compactness of this formulation, quadratic programming models also yield potentially superior lower bounds from fractional quadratic programming relaxations. In fact, the LP relaxations of all known integer linear models yield only the trivial lower bound zero.

We investigate in this article two binary quadratic programming formulations for the VPP. Our main result is that the QP relaxation of one of these models yields a nontrivial lower bound on the number of shunting movements, that is, the fractional QP lower bound is nonzero whenever shunting is required. This model also gave the best computational performance in our tests, even though it is not convex. We also tried to apply convexification techniques, but the results were mixed. Convexification basically worked when the smallest eigenvalue of the objective function was not too negative.

The article is organized as follows. The VPP is described in Section 2. Section 3 discusses integer linear and integer quadratic 2-index models, i.e., we revisit the original approach of Winter. In Section 4 we present integer linear and integer quadratic 3-index models. One of them produces the already mentioned QP bound.

All our computational experiments were done on an Intel(R) Core 2 Quad 2660 MHz with 4Gb Ram, running under openSUSE 11.1 (64 bits). We used `CPLEX` 11.2 [5] to solve linear programs, `SCIP` 1.0 for integer programs [1], and `SNIP` 1.0 for integer non-linear programs [9].

We use the following notation. $V(M)$ denotes the optimal objective value of a model M. If M is an ILP, $V_{LP}(M)$ is the optimal objective value of its LP relaxation, and If M is an MIQCP, $V_{QP}(M)$ is the optimal objective value of its fractional quadratic programming relaxation.

2 The Vehicle Positioning Problem

The Vehicle Positioning Problem (VPP) is a 3-dimensional matching problem, where vehicles that arrive in a sequence $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ must be assigned to parking positions $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ in a depot and depart to service a sequence of timetabled trips $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$. We assume that the first departure trip starts after the last incoming vehicle arrived. Each vehicle a_i has a type $t(a_i)$ and each trip d_i can be serviced only by vehicles of type $t(d_i)$. The parking positions are located in tracks $\mathcal{S}$, and we assume that positions in the tracks are numbered consecutively. Each track $s \in \mathcal{S}$ has size β, and we assume that $\beta|S| \geq n$. Each track is operated as a FIFO stack, that is, vehicles enter the track at one end and leave at the other. Consider a matching with assignments (i, p, k) and (j, q, l), that is, the i-th arriving vehicle is assigned to parking position p in order to service the k-th departing trip and the j-th arriving vehicle is assigned to parking position q in order to service the l-th departing trip. Assume that p and q are located in the same stack; then a shunting movement is required if either $i < j$ and $p > q$ or $p < q$ and $k > l$. In this case, we say that these assignments are in *conflict* and denote the associated *crossings* by $(i, p) \dagger (j, q)$ or $(p, k) \dagger (q, l)$. Given $\mathcal{A}, \mathcal{P}, \mathcal{D}, \mathcal{S}, t$, and β, the VPP is to find a 3-dimensional matching that minimizes the number of crossings. The number of crossings is related to the number of required shuntings.

We remark that there are more complex versions of this problem involving different sizes of vehicles and parking positions, multiple periods, etc. However, we do not consider them here.

3 Two-Index Models

Winter [11] gave the following integer quadratic programming formulation for the VPP:

(**W**)

$$\min \sum_{(a,p)\dagger(a',q)} x_{a,p}x_{a',q} + \sum_{(d,p)\dagger(d',q)} y_{d,p}y_{d',q} \tag{1}$$

$$\textstyle\sum_{a\in\mathcal{A}} x_{a,p} = 1 \qquad p \in \mathcal{P} \tag{2}$$

$$\textstyle\sum_{p\in\mathcal{P}} x_{a,p} = 1 \qquad a \in \mathcal{A} \tag{3}$$

$$\textstyle\sum_{d\in\mathcal{D}} y_{d,p} = 1 \qquad p \in \mathcal{P} \tag{4}$$

$$\textstyle\sum_{p\in\mathcal{P}} y_{d,p} = 1 \qquad d \in \mathcal{D} \tag{5}$$

$$x_{a,p} + y_{d,p} \leq 1 \qquad \substack{(a,p,d)\in\mathcal{A}\times\mathcal{P}\times\mathcal{D}\\ t(a)\neq t(d)} \tag{6}$$

$$x_{a,p}, y_{d,p} \in \{0, 1\}.$$

The model uses binary variables $x_{a,p}$, with $a \in \mathcal{A}$ and $p \in \mathcal{P}$, and $y_{d,p}$, with $d \in \mathcal{D}$ and $p \in \mathcal{P}$. If $x_{a,p} = 1$ ($y_{d,p} = 1$), vehicle a (trip d) is assigned to parking position p. Constraints (2)-(5) define the assignments, the constraint (6) enforces the coherence of these assignments by allowing only vehicles and trips of the same type to be assigned to a given parking position. Finally, the quadratic cost function calculates the number of crossings.

In his work, Winter did not solve the quadratic program directly. Instead, he applied the linearization method of Kaufman and Broeckx [6], obtaining the following integer linear model:

(**LW**)

$$\min \quad \sum_{a\in\mathcal{A},p\in\mathcal{P}} w_{a,p} + \sum_{d\in\mathcal{D},p\in\mathcal{P}} u_{d,p} \tag{7}$$

$$\sum_{a\in\mathcal{A}} x_{a,p} = 1 \qquad p \in \mathcal{P} \tag{8}$$

$$\sum_{p\in\mathcal{P}} x_{a,p} = 1 \qquad a \in \mathcal{A} \tag{9}$$

$$\sum_{d\in\mathcal{D}} y_{d,p} = 1 \qquad p \in \mathcal{P} \tag{10}$$

$$\sum_{p\in\mathcal{P}} y_{d,p} = 1 \qquad d \in \mathcal{D} \tag{11}$$

$$x_{a,p} + y_{d,p} \le 1 \qquad \substack{(a,p,d)\in\mathcal{A}\times\mathcal{P}\times\mathcal{D} \\ t(a)\neq t(d)} \tag{12}$$

$$d^x_{a,p} x_{a,p} - w_{a,p} + \sum_{(a,p)\dagger(a',q)} x_{a',q} \le d^x_{a,p} \qquad \forall p \in \mathcal{P}, a \in \mathcal{A} \tag{13}$$

$$d^y_{d,p} y_{d,p} - u_{d,p} + \sum_{(d,p)\dagger(d',q)} y_{d',q} \le d^y_{d,p} \qquad \forall p \in \mathcal{P}, d \in \mathcal{D} \tag{14}$$

$$x_{a,p}, y_{d,p} \in \{0,1\}$$

$$w_{d,p}, u_{d,p} \in \mathbb{N}.$$

In this model, the integer variables $w_{a,p}$ and $u_{d,p}$ count the number of crossings envolving the assignments (a,p) and (d,p), respectively. $d^x_{a,p}$ and $d^y_{d,p}$ are upper bounds on these variables, respectively, that are computed a priori.

The following is known about these models:

Remark 1. The model **W** has $2n^2$ variables and $n^3 + 4n$ constraints.

Remark 2. The model **LW** has $4n^2$ variables and $n^3 + 2n^2 + 4n$ constraints.

Theorem 1 (WZ00). *The models* ***W*** *and* ***LW*** *are equivalent.*

Theorem 2 (WZ00). $V_{LP}(\boldsymbol{LW}) = 0$.

It is not difficult to modify Winter's proof of Theorem 2 in order to get a similar result for the QP relaxation of his quadratic model:

Theorem 3. $V_{QP}(\boldsymbol{W}) = 0$ *if* $|S| > 1$.

Proof. Let M be a matching where each a_i is assigned to d_i (i.e., first vehicle to first trip, second vehicle to second trip, and so on) and the assignment of the pairs (a_i, d_i) to the parking positions is made according to the following scheme, where each column represents a track:

$(a_{n-\lvert S\rvert}, d_{n-\lvert S\rvert})$	$(a_{n-\lvert S\rvert+1}, d_{n-\lvert S\rvert+1})$	$\ldots$	(a_n, d_n)
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(a_{\lvert S\rvert+1}, d_{\lvert S\rvert+1})$	$(a_{\lvert S\rvert+2}, d_{\lvert S\rvert+2})$	$\ldots$	$(a_{2\lvert S\rvert}, d_{2\lvert S\rvert})$
(a_1, d_1)	(a_2, d_2)	$\ldots$	$(a_{\lvert S\rvert}, d_{\lvert S\rvert})$

Such a matching has no crossings. However, it is not always feasible for **W** because of type mismatches (cf. the coherence equations 6). If the integrality of the variables is relaxed, assigning each pair (a_i, d_i) to the same relative position in each track avoids the restrictions given by the coherence equations. More precisely, if a pair (a_i, d_i) is assigned to the second position of some track (in other words, if $\left\lfloor \frac{i-1}{|S|} \right\rfloor = 1$), we fix $x_{a_i,p_{q'}} = y_{d_i,d_{q'}} = \frac{1}{|S|}$ for each position $q' \in \mathcal{P}$ which is the second position in some track (in other words, if $\left\lfloor \frac{q'-1}{|S|} \right\rfloor = 1$). If $|S| > 1$, equations 6 are satisfied. Since there are no crossings, the objective value is zero. $\square$

A problem with model **W** is that the objective is not convex. This obstacle can be overcome as follows. Initially, we observe that $\sum_{(a,p)\dagger(a',q)} x_{a,p}x_{a',q}$ can be written as $x^T Ax$, where $A \in \{0,1\}^{n^2} \times \{0,1\}^{n^2}$ is the symmetric incidence matrix of all arrival crossings. A classical convexification trick makes use of the eigenvalues of the matrix A. More precisely, if α is the absolute value of the minimum eigenvalue of A, we have

$$x^T Ax = x^T(A - \alpha I)x + \alpha x^T x. \tag{15}$$

As x is binary, this equation can be rewritten as

$$x^T Ax = x^T(A - \alpha I)x + \alpha \sum_i x_i. \tag{16}$$

Finally, in our case, we have $\sum_i x_i = n$ for every feasible solution, that is,

$$x^T Ax = x^T(A - \alpha I)x + \alpha n. \tag{17}$$

The same ideas yield $\sum_{(d,p)\dagger(d,q')} y_{d,p}y_{d,q'} = y^T A'y$. Moreover, it is easy to see that $A' = A$. In explicit terms, the objective can therefore be written as

$$\sum_{(a,p)\dagger(a',q)} x_{a,p}x_{a',q} - \alpha \sum_{(a,p)} x^2_{a,p} + \sum_{(d,p)\dagger(d',q)} y_{d,p}y_{d',q} - \alpha \sum_{(d,p)} y^2_{d,p} + 2\alpha n \tag{18}$$

$$\sum_{(a,p)\dagger(a',q)} x_{a,p}x_{a',q} - \alpha \sum_{(a,p)} (x^2_{a,p} - x_{a,p}) + \sum_{(d,p)\dagger(d',q)} y_{d,p}y_{d',q} - \alpha \sum_{(d,p)} (y^2_{d,p} - y_{d,p}). \tag{19}$$

Applying this substitution to the model **U**, we obtain:

(**CW**)

$$\min \sum_{(a,p)\dagger(a',q)} x_{a,p}x_{a',q} - \alpha \sum_{(a,p)} (x^2_{a,p} - x_{a,p}) + \sum_{(d,p)\dagger(d',q)} y_{d,p}y_{d',q} - \alpha \sum_{(d,p)} (y^2_{d,p} - y_{d,p}) \tag{20}$$

$$\textstyle\sum_{a\in\mathcal{A}} x_{a,p} = 1 \qquad p \in \mathcal{P} \tag{21}$$

$$\textstyle\sum_{p\in\mathcal{P}} x_{a,p} = 1 \qquad a \in \mathcal{A} \tag{22}$$

$$\textstyle\sum_{d\in\mathcal{D}} y_{d,p} = 1 \qquad p \in \mathcal{P} \tag{23}$$

$$\textstyle\sum_{p\in\mathcal{P}} y_{d,p} = 1 \qquad d \in \mathcal{D} \tag{24}$$

$$x_{a,p} + y_{d,p} \leq 1 \qquad \begin{smallmatrix}(a,p,d)\in\mathcal{A}\times\mathcal{P}\times\mathcal{D}\\ t(a)\neq t(d)\end{smallmatrix} \tag{25}$$

$$x_{a,p}, y_{d,p} \in \{0,1\}.$$

Tables 1 and 2 give the results of a computational comparison of models **W** and **LW**, and **W** and **CW**, respectively, on a test set of ten randomly generated instances of small and medium sizes. The first column in these tables give the name x-y-z of the problem. Here, x is the number of vehicle types, y is the number of tracks, and z= β is the number of parking positions per track. The columns labeled Row, Col, and NZ give the number of constraints, variables, and nonzeros of the respective model. Columns Nod give the number of nodes in the search tree generated by the respective solver (`SCIP` with LP solver `CPLEX` for **LW** and `SNIP` for **W**) and T/s the computation time in seconds.

Comparing the results for models **CW** and **W** shows that convexification led to an improvement, but not enough to outperform the linearized model **LW**, in particular not on the larger instances.

	LW					**W**				
Name	Row	Col	NZ	Nod	T/s	Row	Col	NZ	Nod	T/s
3-6-4	10465	2305	43741	1343	58	9325	1165	21889	215	142
4-6-4	11617	2305	46045	12849	265	10477	1165	24193	816	214
5-6-4	12289	2305	47389	32870	654	11149	1165	25537	1010	237
3-7-3	7141	1765	25257	234	18	6273	897	14995	590	58
4-7-3	7897	1765	26769	17220	15	7029	897	16507	523	52
5-7-3	8359	1765	27693	114	19	7491	897	17431	651	64
3-7-4	16297	3137	68391	17220	124	14743	1583	33937	480	121
4-7-4	18145	3137	72087	7393	574	16591	1583	37633	1609	251
5-7-4	19209	3137	74215	60590	2171	17655	1583	39761	113997	11845
3-7-5	31151	4901	152125	59992	3251	28715	2465	64471	6612	76685

Table 1. Comparing models **LW** and **W**.

	CW					**W**				
Name	Row	Col	NZ	Nod	T/s	Row	Col	NZ	Nod	T/s
3-6-4	9325	1165	21913	1543	116	9325	1165	21889	215	142
4-6-4	10477	1165	29977	24217	690	10477	1165	24193	816	214
5-6-4	11149	1165	25561	586	96	11149	1165	25537	1010	237
3-7-3	6273	897	15023	245	29	6273	897	14995	590	58
4-7-3	7029	897	16535	324	32	7029	897	16507	523	52
5-7-3	7491	897	17459	858	42	7491	897	17431	651	64
3-7-4	14743	1583	33965	2122	176	14743	1583	33937	480	121
4-7-4	16591	1583	37661	1526	242	16591	1583	37633	1609	251
5-7-4	17655	1583	39789	1320	1544	17655	1583	39761	113997	11845
3-7-5	28715	2465	64499	627	40145	28715	2465	64471	6612	76685

Table 2. Comparing models **CW** and **W**.

4 Three-Index Models

Gallo and Di Miele [3] improved Winter's model by noting that assignments (a, s) and (s, d) of arrivals and departures to stacks implicitly determine the parking positions uniquely; this produces a substantially smaller model. Kroon, Lentink and Schrijver [7] took this idea in order to create a 3-index model with a stronger LP relaxation (although the lower bound is still equal to zero):

$$
\begin{aligned}
&(\mathbf{LU}) \\
&\min \sum_{(a,s,d)\in\mathcal{A}\times\mathcal{S}\times\mathcal{D}(a,s,d)} r_{a,s,d} && && (26)\\
&\textstyle\sum_{(s,d)\in\mathcal{S}\times\mathcal{D}} x_{a,s,d} = 1 && a\in\mathcal{A} && (27)\\
&\textstyle\sum_{(a,s)\in\mathcal{A}\times\mathcal{S}} x_{a,s,d} = 1 && d\in\mathcal{D} && (28)\\
&\textstyle\sum_{(a,d)\in\mathcal{A}\times\mathcal{D}} x_{a,s,d} \le \beta && s\in\mathcal{S} && (29)\\
&\sum_{a'<a} x_{a',s,d} + \sum_{d'\le d} x_{a,s,d'} + r_{a,s,d} \le 1 && (a,s,d)\in\mathcal{A}\times\mathcal{S}\times\mathcal{D} && (30)\\
&x_{a,s,d}, r_{a,s,d} \in \{0,1\}.
\end{aligned}
$$

This model uses binary variables $x_{a,s,d}$, with $a \in \mathcal{A}$, $s \in \mathcal{S}$, and $d \in \mathcal{D}$, where $x_{a,s,d} = 1$ if and only if vehicle a is assigned to the trip d and is parked on the track s. Equations (27) and (28) are assignment constraints for vehicles and trips, equations (29) are capacity restrictions for each track in $\mathcal{S}$. Inequalities (30) count crossings using binary variables $r_{a,s,d}$.

We propose the following integer quadratic 3-index formulation for the problem:

$$
\begin{aligned}
&(\mathbf{U}) \\
&\min \sum_{s,(a,d)\dagger(a',d')} x_{a,s,d}x_{a',s,d'} && && (31)\\
&\textstyle\sum_{(s,d)\in\mathcal{S}\times\mathcal{D}} x_{a,s,d} = 1 && a\in\mathcal{A} && (32)\\
&\textstyle\sum_{(a,s)\in\mathcal{A}\times\mathcal{S}} x_{a,s,d} = 1 && d\in\mathcal{D} && (33)\\
&\textstyle\sum_{(a,d)\in\mathcal{A}\times\mathcal{D}} x_{a,s,d} \le \beta && s\in\mathcal{S} && (34)\\
&x_{a,s,d} \in \{0,1\}.
\end{aligned}
$$

Equations (32), (33), and (34) are equal to (27), (28), and (29), respectively. Crossings are counted directly by the quadratic cost function (31).

The models **U** und **LU** have the following properties:

Remark 3. Model **LU** has $2sn^2$ variables and $2n + s + sn^2$ constraints.

Remark 4. Model **U** has sn^2 variables and $2n + s$ constraints.

Theorem 4. *$V_{LP}(\boldsymbol{LU}) = 0$ if $|S| > 1$.*

Proof. Let M be a matching where each a_i is assigned to d_i (i.e., first vehicle to first trip, second vehicle to second trip, and so on). Assign $\frac{1}{|S|}$ to each variable $x_{a,s,d}$ such that $(a,d) \in M$. In this case, constraints (27) and (28) clearly hold, as

$$\sum_s x_{a,s,d} = \sum_s \frac{1}{|S|} = 1$$

for each $a \in \mathcal{A}$ and $d \in \mathcal{D}$. Moreover, as $|M| = n$,

$$\sum_{(a,d)} x_{a,s,d} = n\frac{1}{|s|} \leq \beta$$

for each $s \in \mathcal{S}$, satisfying (29). Finally, if $|S| > 1$, the shunting restrictions (30) hold with $r_{a,s,d} = 0$ for each (a,s,d), yielding a solution of cost zero. □

Model **U** can be strenghtened by penalizing not only crossings but also inconsistent assignments:

(**UI**)

$$\min \sum_{s,(a,d)\dagger(a',d')} x_{a,s,d}x_{a',s,d'} + \sum_{a,(s,d)\neq(s',d')} x_{a,s,d}x_{a,s',d'} \tag{35}$$

$$\textstyle\sum_{(s,d)\in\mathcal{S}\times\mathcal{D}} x_{a,s,d} = 1 \qquad a \in \mathcal{A} \tag{36}$$

$$\textstyle\sum_{(a,s)\in\mathcal{A}\times\mathcal{S}} x_{a,s,d} = 1 \qquad d \in \mathcal{D} \tag{37}$$

$$\textstyle\sum_{(a,d)\in\mathcal{A}\times\mathcal{D}} x_{a,s,d} \leq \beta \qquad s \in \mathcal{S} \tag{38}$$

$$x_{a,s,d} \in \{0,1\}$$

The objective function of **UI** contains an additional penalty term

$$\sum_{a,(s,d)\neq(s',d')} x_{a,s,d}x_{a,s',d'}$$

for inconsistent assignments of vehicles (i.e., if a vehicle is assigned to more than one track and/or more than one trip, the value of the product of the variables representing such inconsistent assignment is added). The penalty term is zero for every feasible integer solution, but it increases the objective value of the QP relaxation.

Theorem 5. *The models* ***U****,* ***UI****, and* ***LU*** *are equivalent.*

Theorem 6. $V_{QP}(\boldsymbol{UI}) > 0$ *if* $V(\boldsymbol{UI}) > 0$.

Proof. If $V(\mathbf{UI}) > 0$, there is a crossing for each possible assignment of vehicles to trips and tracks. Let x^* be an optimal solution of the QP relaxation of **UI**. Consider the vector $\lceil x^* \rceil$. If $\lceil x^* \rceil$ contains an integer solution, there is a crossing and

$$\sum_{s,(a,d)\dagger(a',d')} \lceil x^*_{a,s,d} \rceil \lceil x^*_{a',s,d'} \rceil > 0.$$

Then

$$\sum_{s,(a,d)\dagger(a',d')} x^*_{a,s,d} x^*_{a',s,d'} > 0.$$

If $\lceil x^* \rceil$ does not contain an integer solution, there is an inconsistent assignment and therefore

$$\sum_{a,(s,d)\neq(s',d')} x^*_{a,s,d} x^*_{a,s',d'} > 0.$$

□

As far as we know, $V_{QP}(\mathbf{UI})$ is the first nontrivial lower bound for the VPP. We remark that the same idea can also be used to strengthen some of the linear models such that they sometimes also produce nonzero lower bounds. We have, however, not been able to prove a result similar to Theorem 6, that is, that the lower bound is *always* nonzero if shuntings are required.

	LU					**U**				
Name	Row	Col	NZ	Nod	T/s	Row	Col	NZ	Nod	T/s
3-6-4	3511	4609	38017	1	1	61	1159	4621	28	4
4-6-4	3511	4321	30241	1	0	61	871	3463	69	3
5-6-4	3511	4153	25675	59	15	61	703	2797	16	2
3-7-3	3137	4117	30871	12	8	57	1037	4124	16	3
4-7-3	3137	3865	24816	1	1	57	785	3123	20	2
5-7-3	3137	3711	21274	54	6	57	631	2507	27	28
3-7-4	5552	7323	67803	1	1	71	1842	7351	22	10
4-7-4	5552	6861	53509	41	29	71	1380	5503	33	7
5-7-4	5552	6595	45389	1	1	71	1114	4432	21	4
3-7-5	8653	11439	126099	1	4	85	2871	11467	17	34
4-7-5	8653	10725	98582	59	44	85	2157	8604	22	21
5-7-5	8653	10291	82321	26	38	85	1723	6875	31	12
6-7-6	12440	14407	117307	227	200	99	2066	8240	27	32

Table 3. Comparing models **LU** and **U**.

Table 3 gives the results of a computational comparision of models **U** and **LU** on the same set of test problems as in Section 3 plus one additional model that could not be solved there. Model **UI** could not be tested yet because of a software problem.

The comparision of the results for models **CW** and **W** from Section 3 and those for **LU** and **U** show a clear superiority of the **U** models over the **W** models. Among the **U** models, the integer quadratic model **U** outperformed the integer linear model **LU**. The next instance 7-8-7, however, could not be solved using any of our formulations.

We have also tried to apply the convexification technique of Section 3 to model **U**, but this time it did not bring any performance gain. A possible explanation for this behavior is that the spectra of the objectives of the **U** instances have negative eigenvalues of much larger magnitude than those in the **W** instances.

Acknowledgement

We thank Stefan Vigerske for his advice with respect to the formulation of integer quadratic programs and SNIP support.

References

1. T. Achterberg, *Constraint integer programming*, Ph.D. thesis, TU Berlin, (2007).
2. R. Freling, R. Lentink, L. Kroon, and D. Huisman, *Shunting of passenger train units in a railway station.* ERIM Report Series Research in Management, 2002.
3. G. Gallo and F. Di Miele, *Dispatching buses in parking depots*, Transportation Science, 35 (2001), pp. 322–330.
4. M. Hamdouni, F. Soumis, and G. Desaulniers, *Dispatching buses in a depot minimizing mismatches.* 7th IMACS, Scientific Computing Toronto, Canada, 2005.
5. ILOG, *CPLEX website.* http://www.ilog.com/products/cplex/.
6. L. Kaufmann and F. Broeckx, *An algorithm for the quadratic assignment problem*, European J. Oper. Res., 2 (1978), pp. 204–211.
7. L. Kroon, R. Lentink, and A. Schrijver, *Shunting of passenger train units: an integrated approach*, ERIM Report Series Reference No. ERS-2006-068-LIS, (2006).
8. I. Nowak, *Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming*, Birkhäuser Verlag, 2005.
9. S. Vigerske, *Nonconvex mixed-integer nonlinear programming.* http://www.math.hu-berlin.de/ stefan/B19/.
10. T. Winter, *Online and Real-Time Dispatching Problems*, PhD thesis, TU Braunschweig, 1998.
11. T. Winter and U. Zimmermann, *Real-time dispatch of trams in storage yards*, Annals of Operations Research, 96 (2000), pp. 287–315.