

Reformulations for integrated planning of railway traffic and network maintenance

Tomas Lidén*

Linköping University, Department of Science and Technology, Norrköping SE-601 74, Sweden

Hamish Waterer†

University of Newcastle, School of Mathematical & Physical Sciences, Callaghan NSW 2308, Australia

August 27, 2019

Abstract

This paper addresses the scheduling problem of coordinating train services and network maintenance windows for a railway system. We present model reformulations, for a mixed integer linear optimization model, which give a mathematically stronger model and substantial improvements in solving performance — as demonstrated with computational experiments on a set of synthetic test instances. As a consequence, the solution times are reduced and more instances can be solved to optimality within a given time limit.

Keywords: Railway scheduling; Maintenance planning; Optimization

1 Introduction

Railway is an important transportation mode, both for passenger traffic and high-volume freight logistics. One of the core tactical planning problem for all railway systems is the scheduling of access to the railway infrastructure, which can be seen as resource planning for the infrastructure components (stations, lines, yards, tracks, switches, signalling blocks, etc). This planning includes producing a timetable for the train traffic as well as access (or possession) plans for maintenance and work tasks. Timetables and possession plans will in turn form the basis for other resource plans, such as rolling stock plans and crew schedules for the train operators, as well as equipment and work force plans for maintenance and renewal contractors.

Train services and maintenance tasks should ideally be planned together, but have mostly been treated as separate planning problems. While planning of train operations has been extensively studied in the research literature [2, 3, 4, 5, 6], maintenance planning [9] has received much less attention. As for the joint planning of train services and network maintenance, there are a few examples which consider the introduction of a small number of work possessions into an existing train timetable [7, 14] or operative plan [1] by allowing different types of adjustments to the trains.

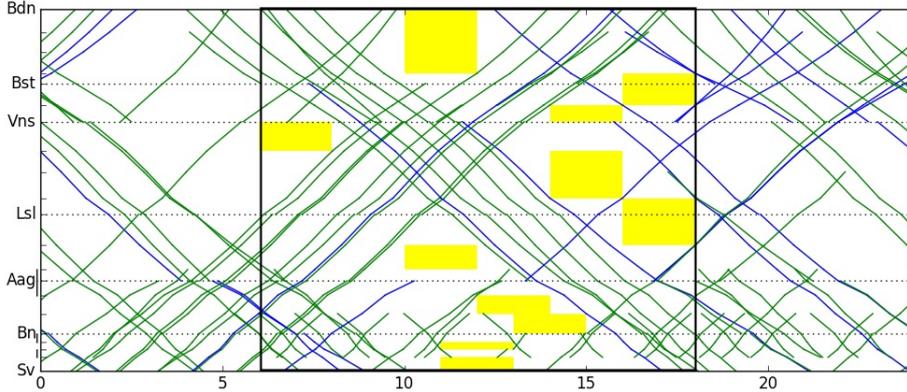
This research focuses on the long term tactical coordination of a large volume of *maintenance windows* and intended train services on a railway network. Maintenance windows is a planning concept where the infrastructure manager designs regular time windows, reserved for maintenance work, before the timetable is constructed and given as a prerequisite for the procurement of maintenance contracts. An example of how such plans might look is given in Figure 1, as a train and work graph. The geographic distance and sectioning into links is shown on the vertical axis while time is on the horizontal axis. Train services are shown as tilted lines while maintenance windows are shown as yellow boxes. In this case no trains are allowed to run “through” the maintenance windows.

An initial MILP model that solves this coordination problem to optimality has been presented in [12]. Model extensions for assigning maintenance crew resources and considering their costs and limitations regarding spatial availability as well as work and rest time regulations are treated in [13]. The latter

*Corresponding author, tomas.liden@liu.se

†hamish.waterer@newcastle.edu.au

Figure 1: Train and work graph example



model also has a stronger formulation for the train and maintenance window scheduling, but that paper only briefly summarizes these model improvements. Here we will examine these improvements more closely.

The purpose of this paper is to describe and compare model formulations for the coordination of train services and maintenance windows. The contributions are a detailed explanation of the different formulations, and computational experiments that compare the formulations and show the resulting performance improvements. In an earlier version [10], the experiments were carried out on two different solver versions, while we here use a common setup. The general results are the same as previously reported, but we analyse them a bit more cautiously here.

The optimization model consists of a train scheduling part and a maintenance scheduling part, coordinated by a capacity model that controls the number of trains scheduled over each link per time period. Reformulations are introduced in both of the scheduling parts, where we denote the original model by ORG and the improved model by IMP.

The train scheduling part in ORG uses cumulative train entry/exit variables (for each link and time period) and implicit link usage variables. In contrast, IMP uses binary train entry/exit detection variables and explicit link usage variables. These changes increase the number of variables, but decrease the number of constraints. The linear relaxation does not become tighter, but the reformulation leads to a model structure that the MILP solver can take advantage of, as shown by the computational results.

The main improvement in IMP concerns the maintenance scheduling part. First of all, some coupling constraints have been aggregated, but more importantly a tighter formulation has been used for the maintenance work and window start variables — according to the modelling for bounded up/down sequences as presented in [15, Section 11.4, pp 341–343]. These improvements do make the linear relaxation tighter and in addition the MILP solver is able to more effectively reduce the size of the problem in presolve.

The net effect of these model improvements is that the solver performance improves: the solution times are reduced and more instances can be solved to optimality within a given time limit.

The remainder of the paper is organised as follows: Section 2 gives the mathematical formulation after first introducing the necessary notation and giving an overview of the model structure. The reformulations for the train scheduling part are then described, followed by the changes for the maintenance window scheduling part. Computational experiments are presented in Section 3, followed by some concluding remarks.

2 Mathematical formulation

2.1 Notation and model structure

The railway network is modelled by a link set L . The scheduling problem has a planning horizon of length H , defined by the set $T = \{0, \dots, H-1\}$ of unit size time periods $t \in T$, each representing the real-valued event time interval $[t, t+1)$.

For the train traffic we have a set S of train services. Each train service $s \in S$ has a set R_s of possible routes. Each route $r \in R_s$ implies a sequence L_r of links, and the set L_s of all possible links that train service s can traverse is given by the union of the sets L_r for all $r \in R_s$. The scheduling of train service s requires the selection of a route $r \in R_s$ as well as determining entry and exit times for each link in that route, such that all event times are within the scheduling window defined by an interval of time periods $T_s \subseteq T$.

For the maintenance, a subset $L^M \subseteq L$ of the links are scheduled with maintenance windows. For each link $l \in L^M$ we have a set W_l of window options. The scheduling of maintenance windows on link l requires the selection of a window option $o \in W_l$, defined as a tuple $o = (\eta_o, \theta_o)$ that gives the required number η_o of maintenance windows to schedule and the window length θ_o expressed as an integer number of time periods. As an example, $W_l = \{(1, 3), (2, 2)\}$ means that either one window of length three or two windows of length two shall be scheduled on link l .

The model has two groups of variables. The main variables for scheduling train services are

z_{sr}	route choice: whether train service s uses route r or not
e_{sl}^+, e_{sl}^-	event time: entry(+)/exit(-) time for service s on link l
x_{slt}^+, x_{slt}^-	link entry/exit: whether train service s enters/exits link l in time period t or not
u_{slt}	link usage: whether train service s uses link l in time period t or not
n_{lt}^h	number of train services traversing link l in direction h during time period t

The variables for scheduling maintenance windows are:

w_{lo}	maintenance window option choice: whether link l is maintained with window option o or not
y_{lt}	maintenance work: whether link l is maintained in time period t or not
v_{lot}	work start: whether maintenance on link l according to window option o is started in time period t or not

The model can be summarized as follows:

$$\text{minimize } c(\mathbf{z}, \mathbf{e}, \mathbf{y}, \mathbf{v}) \quad (1)$$

$$\text{subject to } \mathbf{A}(\mathbf{z}, \mathbf{e}, \mathbf{x}, \mathbf{u})^{\text{route}} \quad (2)$$

$$\mathbf{A}(\mathbf{z}, \mathbf{e})^{\text{trains}} \quad (3)$$

$$\mathbf{A}(\mathbf{w}, \mathbf{y}, \mathbf{v})^{\text{maintenance}} \quad (4)$$

$$\mathbf{A}(\mathbf{u}, \mathbf{n}, \mathbf{y})^{\text{capacity}} \quad (5)$$

variable types and bounds

where $c(\cdot)$ is the objective function and $\mathbf{A}(\cdot)$ are linear constraint functions over one or more of the indicated variables. The objective (1) is a linear combination of the train and maintenance scheduling variables, while the constraints enforce: (2) correct (feasible) bounds on the train events and linking of entry/exit and usage variables according to the selected route, (3) sufficient travel durations and dwell times along the chosen route, (4) sufficient maintenance windows scheduled according to the chosen option, and (5) that the available network capacity is respected.

The ORG and IMP formulations differ regarding the variables x and u , and the constraints (2) and (4), which will be described in the following sections. For details regarding the objective function and other constraints, see [13].

2.2 Train scheduling

The ORG formulation uses cumulative variables $\bar{x}_{slt}^+, \bar{x}_{slt}^-$, which take value one if train service s has entered/exited link l in time period t or earlier. The link usage is given by the implicit variables $u_{slt} := \bar{x}_{slt}^+ - \bar{x}_{sl,t-1}^-$, with the convention that $\bar{x}_{sl,t-1}^a = 0$ for $t = 0$, $a \in \{+, -\}$.

The constraint set (2) for ORG is:

$$\bar{x}_{slt}^a \geq \bar{x}_{sl,t-1}^a \quad \forall s \in S, l \in L_s, t \in T_s, a \in \{+, -\} \quad (2.1a)$$

$$\bar{x}_{sl, \text{last}(T_s)}^a = \sum_{r \in R_s: l \in L_r} z_{sr} \quad \forall s \in S, l \in L_s, a \in \{+, -\} \quad (2.2a)$$

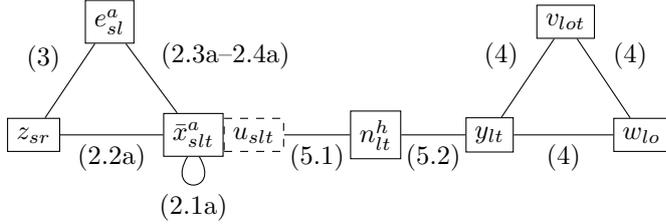
$$e_{sl}^a \geq \sum_{t \in T_s} LB_t^a (\bar{x}_{slt}^a - \bar{x}_{sl,t-1}^a) \quad \forall s \in S, l \in L_s, a \in \{+, -\} \quad (2.3a)$$

$$e_{sl}^a \leq \sum_{t \in T_s} UB_t^a (\bar{x}_{slt}^a - \bar{x}_{sl,t-1}^a) \quad \forall s \in S, l \in L_s, a \in \{+, -\} \quad (2.4a)$$

where LB_t^a and UB_t^a are the lower and upper bound time values for entry and exit in time period t . The constraints enforce: (2.1a) the cumulative property, (2.2a) that all links in the selected route are visited, and (2.3a–2.4a) correct lower and upper bounds for the event variables.

The structure of this model is illustrated in Figure 2 as a constraint (or co-occurrence) graph with vertices for the variables and edges connecting variables that occur in the same constraint. The constraints are cliques in the graph as indicated in the figure. In this case the (constraint) cliques are of size 1, 2 and 3, which means that one to three variable types will occur in the constraints.

Figure 2: Variable and constraint graph — ORG formulation

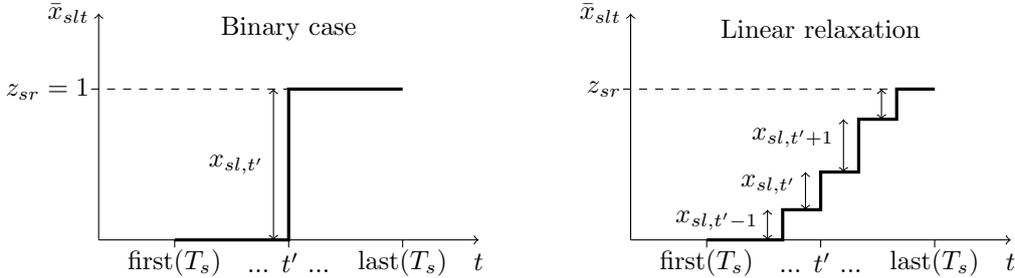


The IMP formulation uses binary detection variables x_{slt}^+, x_{slt}^- to track whether service s enters or exits link l in time period t or not. These variables correspond to the cumulative \bar{x} variables in the ORG formulation as follows

$$x_{slt}^a = \bar{x}_{slt}^a - \bar{x}_{sl,t-1}^a$$

This relation is illustrated in Figure 3, both for the binary case and for a linear relaxation.

Figure 3: Relation between cumulative variables \bar{x} and detection variables x .



Using the expression

$$\bar{x}_{slt}^a = \sum_{t' \in T_s: t' \leq t} x_{slt'}^a$$

we transform constraints (2.2a–2.4a) and introduce explicit u variables to get the IMP formulation:

$$\sum_{t \in T_s} x_{slt}^a = \sum_{r \in R_s: l \in L_r} z_{sr} \quad \forall s \in S, l \in L_s, a \in \{+, -\} \quad (2.2b)$$

$$e_{sl}^a \geq \sum_{t \in T_s} LB_t^a x_{slt}^a \quad \forall s \in S, l \in L_s, a \in \{+, -\} \quad (2.3b)$$

$$e_{sl}^a \leq \sum_{t \in T_s} UB_t^a x_{slt}^a \quad \forall s \in S, l \in L_s, a \in \{+, -\} \quad (2.4b)$$

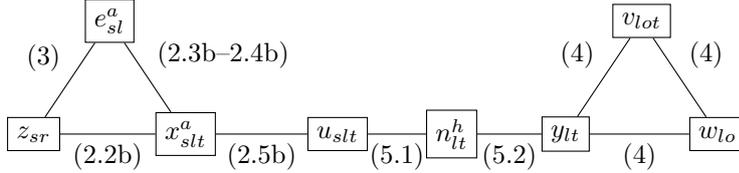
$$u_{slt} = \sum_{t' \in T_s: t' \leq t} x_{slt'}^+ - \sum_{t' \in T_s: t' \leq t-1} x_{slt'}^- \quad \forall s \in S, l \in L_s, t \in T_s \quad (2.5b)$$

Note that the cumulative constraints disappear, but that we have new constraints (2.5b) for the usage variables. The IMP formulation has $|S||L_s||T_s|$ additional variables but $|S||L_s||T_s|$ fewer constraints compared to ORG. Also the train counting constraints ($n_{lt}^h = \sum u_{slt}$) operate on the explicit u variables and hence there are fewer nonzeros in IMP compared to ORG.

The IMP formulation has one more set of variables, u , to branch on. We observe that branching on the x_{slt}^a variables in IMP leads to an unbalanced branch-and-bound tree in that the 1-branch is strong, but the 0-branch weak. Branching on the \bar{x}_{slt}^a variables in ORG is slightly better in this respect, while branching on the u_{slt} variables in IMP gives a more balanced tree.

The constraint graph for the IMP formulation is shown in Figure 4.

Figure 4: Variable and constraint graph — IMP formulation



2.3 Maintenance scheduling

In the following we study the formulation differences between ORG and IMP for the maintenance scheduling constraints (4). In the ORG formulation we have

$$\sum_{o \in W_l} w_{lo} = 1 \quad \forall l \in L \quad (4.1)$$

$$\sum_{t \in T} v_{lot} \geq \eta_o w_{lo} \quad \forall l \in L, o \in W_l \quad (4.2)$$

$$v_{lot} + 1 \geq y_{lt} - y_{l,t-1} + w_{lo} \quad \forall l \in L^M, o \in W_l, t \in T \quad (4.3a)$$

$$v_{lot} \leq w_{lo} \quad \forall l \in L^M, o \in W_l, t \in T \quad (4.4)$$

$$v_{lot} \leq y_{lt} \quad \forall l \in L^M, o \in W_l, t \in T \quad (4.5a)$$

$$v_{lot} \leq 1 - y_{l,t-1} \quad \forall l \in L^M, o \in W_l, t \in T \quad (4.6a)$$

$$\sum_{t'=t}^{t+\theta_o} y_{lt'} \geq \theta_o v_{lot} \quad \forall l \in L^M, o \in W_l, t \in T \quad (4.7a)$$

Constraint (4.1) ensures that exactly one window option is used, while (4.2) ascertain a sufficient number of maintenance windows. Constraints (4.3a–4.6a) ensure the correct coupling of work start variables (v_{lot}), window choice (w_{lo}) and work variables (y_{lt}), while constraint (4.7a) imposes the required maintenance window lengths.

The first reformulation in IMP is aggregation of the coupling constraints (4.3a,4.5a,4.6a). This is possible since (4.1) ensures that exactly one window option will be selected. Consequently, thanks to (4.4), only one v_{lot} variable for each l, t combination can be non-zero. Therefore we can use summations over the window options, as follows:

$$\sum_{o \in W_l} v_{lot} \geq y_{lt} - y_{l,t-1} \quad \forall l \in L^M, t \in T \quad (4.3b)$$

$$\sum_{o \in W_l} v_{lot} \leq y_{lt} \quad \forall l \in L^M, t \in T \quad (4.5b)$$

$$\sum_{o \in W_l} v_{lot} \leq 1 - y_{l,t-1} \quad \forall l \in L^M, t \in T \quad (4.6b)$$

Next, we make use of a model for bounded on/off sequences, presented in [15, Proposition 11.5], where the formulation describes the convex hull. Thus there is no tighter formulation for that set of

variables. We extend this model with the window option choice w_{lo} and can then replace (4.7a) with the following constraints:

$$\sum_{o \in W_l} \left[\sum_{t'=t+1-\theta_o}^t v_{lot'} \right] \leq y_{lt} \quad \forall l \in L^M, t \in T \quad (4.7b)$$

$$\sum_{t'=t+1-\theta_o}^t v_{lot'} + 1 \geq y_{lt} + w_{lo} \quad \forall l \in L^M, o \in W_l, t \in T \quad (4.8b)$$

$$\sum_{o \in W_l} v_{lot} \leq 1 - y_{l,t-1} \quad \forall l \in L^M, t \in T \quad (4.9b)$$

$$\sum_{t'=t+1}^{t+MS_o} v_{lot'} \geq w_{lo} - y_{lt} \quad \forall l \in L^M, o \in W_l, t = 1, \dots, H - MS_o \quad (4.10b)$$

The constraints (4.7b)-(4.8b) enforce that each window should span exactly θ_o time periods, where we note that (4.7b) dominates (4.5b). Constraint (4.9b) enforces at least one time period between two maintenance windows, but since it is precisely the same as (4.6b) and hence redundant, it is not needed. However, if there are requirements for larger minimum separation between windows on each link the constraint is needed, and the LHS of (4.9b) should have a corresponding forward going sum over time indices in v_{lot} . Constraint (4.10b) will make sure that the maximum separation (MS_o) between windows is respected. This constraint and (4.8b) will only be active for the chosen window option, thanks to the w_{lo} variable. Note that aggregation over the window options can only be used for constraints (4.7b) and (4.9b).

3 Computational results

The same set of synthetic test instances as in [12] has been used for evaluating the efficiency of the different formulations. The data instances are available as JSON files together with a set of Python parsers at <https://github.com/TomasLiden/mwo-data.git>. The test set consists of nine line instances (L1–L9) and five network instances (N1–N5), having a planning horizon of five hours to one week divided into 1 h periods and with 20 to 350 train services. All line instances except L4 are single track, while the network instances have a mixture of single and double track links. The trains are uniform with no runtime or cost differences — only the preferred departure times differ. These simplifications, which make the trains almost indistinguishable for the solver, are used in order to test the scalability and solvability of the models. Real-life instances will of course use more realistic settings for costs and runtimes. Each link is scheduled with a daily volume of 1–3 hours of maintenance windows.

In real-life cases, a typical link would have a length of 40–100 km with a couple of intermediate meet/pass loops. The line instances are therefore comparable to railway lines with an actual length of 100–1500 km. The network instances would translate to a geographic size of about 200 by 150 km. The daily number of train services are 40–80, which is roughly one to two trains per hour and direction. Thus the instances represent low density traffic cases, which is reasonable when scheduling 1–3 hour maintenance windows each day.

The computational results are presented in Table 1, where the left-most columns give the instance properties, the middle columns list solution statistics for the following alternatives:

1. ORG - the original formulation,
2. TS - the improved train scheduling formulation, and
3. IMP - the complete improved formulation;

and the right-most columns list the relative improvement in initial LP value (without and with pre-solve active) compared to the ORG formulation. The tests were run on a network server (12 CPU's, 24 GB memory and Ubuntu 16.04.3) with Gurobi 7.5.2 as solver. The number of threads were limited to 4, the relative optimality gap tolerance set to 0.1%, and the maximum computation time to 3600 seconds. All other solver options have been left at their default values.

Table 1: Instances, properties, performance (time (seconds) to reach optimality or else optimality gap after 3600 seconds) and relative improvement in initial LP value vs ORG (without / with pre-solve).

Case	Properties			Performance (time / gap)			LP improvement (%)	
	L	T	S	ORG	TS	IMP	TS	IMP
L1	4	5	20	50	3	3	0 / 0	0 / 0
L2	4	5	20	12	3	3	0 / 0	0 / 0
L3	4	12	40	2912	51	54	0 / 0	1.08 / 0.74
L4	4	12	40	1	2	1	0 / 0	0.68 / 0.49
L5	9	24	40	0.22%	3555	769	0 / 0	1.33 / 0.91
L6	9	48	80	0.47%	0.15%	1568	0 / 0	0.78 / 0.58
L7	18	24	80	3.82%	2.75%	0.60%	0 / 0	1.21 / 0.82
L8	18	96	160	9.46%	13.4%	5.60%	0 / 0	0.36 / 0.17
L9	25	168	350	46.8%	69.9%	86.0%	0 / 0.00	0.50 / 0.38
N1	9	5	20	7	2	2	0 / 0	0.80 / 0
N2	9	24	50	10	3	4	0 / 0.05	0.50 / 0.21
N3	9	48	100	157	212	14	0 / -0.01	0.67 / 0.51
N4	9	96	200	357	240	219	0 / 0.00	0.16 / 0.07
N5	9	168	350	0.11%	2453	630	0 / 0.00	0.20 / 0.14

Run with Gurobi 7.5.2 on a network server with 12 CPU's, 24 GB memory and Ubuntu 16.04.3.

Figure 5: Performance profile — solution time.

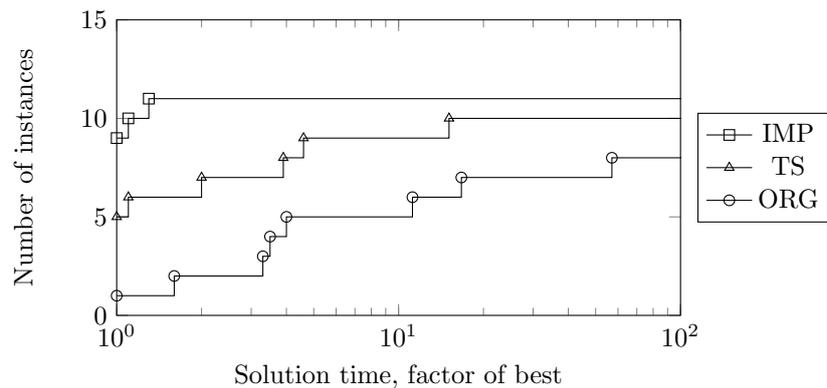
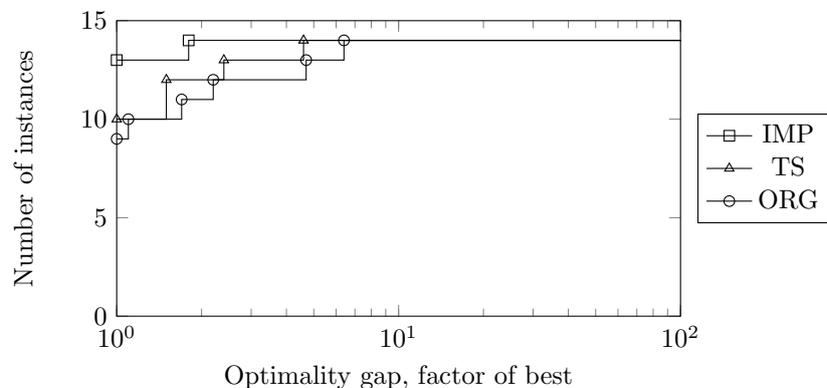


Figure 6: Performance profile — optimality gap.



There is a clear improvement in solving performance — both the solution times of the solved instances and the optimality gap of the unsolved instances are reduced, except for instance L9. Also we see that IMP is tighter, since the initial LP values improve, both without and with pre-solve active. It can be noted that the reformulations were initially developed and tested with Gurobi 6.0.5 and 6.5, and with these solver versions the performance improvements are slightly greater (as reported in [10]).

To illustrate the improvements, performance profile plots are used which show the accumulated number of instances (on the vertical axis) exceeding a value for the particular quality measure (on the logarithmic horizontal axis) — normalised as a factor of the best outcome for all alternatives. Thus an alternative has better performance when it is above (= more instances) and to the left (= better quality) of another curve. In Figure 5 the solution time is the quality measure, while Figure 6 shows the optimality gap for those instances that have not been solved to optimality. The improvements obtained for the reformulations are clear — especially for the solution times.

The net result is that three more instances (L5, L6 and N5) are solved to optimality, the optimal solutions are obtained faster (with a speed up between 2 and 10 times) and one more instance (L7) reaches an optimality gap that is less than 1.0%, which can be considered an acceptable solution quality for the cost factors being used.

4 Concluding remarks

We have investigated and found reformulations that substantially improve the solving performance for an optimization model that jointly schedules train services and network maintenance windows. The reformulations include the removal of cumulative variables, making implicit variables explicit, using aggregation where appropriate, but most importantly to use a tighter model for bounded up/down sequences (according to [15]).

A large number of other formulation alternatives have also been investigated and tested, including various combinations of aggregated and disaggregated constraints, special ordered sets etc, but the ones presented have performed the best.

These improvements have made it possible to extend the model with maintenance resource considerations (see [13]), and in recent work the models have also been applied to real-world problems of realistic size [11]. In the latter work cyclic scheduling is used, which unfortunately destroys the integral properties of the previously mentioned model for bounded up/down sequences. Hence, the mathematical properties of cyclic on/off sequences are currently being studied with the aim of finding methods for strengthening such models in [8].

Acknowledgments

The work of Tomas Lidén is performed as part of the research project “Efficient planning of railway infrastructure maintenance”, funded by the Swedish Transport Administration with the grant TRV 2013/55886 and conducted within the national research program “Capacity in the Railway Traffic System”, during a research visit to University of Newcastle, Australia.

Hamish Waterer is supported in part by the Australian Research Council and Aurizon Network Pty Ltd under the grant LP140101000.

We also thank Thomas Kalinowski for support, inspiration and invaluable contributions with other research subjects.

References

- [1] A. R. Albrecht, D. M. Pantou, and D. H. Lee. Rescheduling rail networks with maintenance disruptions using problem space search. *Computers & Operations Research*, 40(3):703–712, March 2013. doi:10.1016/j.cor.2010.09.001.
- [2] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63:15–37, May 2014. doi:10.1016/j.trb.2014.01.009.

- [3] V. Cacchiani and P. Toth. Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727–737, June 2012. doi:10.1016/j.ejor.2011.11.003.
- [4] A. Caprara, L. Kroon, M. Monaci, M. Peeters, and P. Toth. Passenger Railway Optimization (ch 3). In *Handbooks in Operations Research and Management Science*, volume 14, pages 129–187. Elsevier, 2007. doi:10.1016/S0927-0507(06)14003-7.
- [5] A. Caprara, L. Kroon, and P. Toth. Optimization problems in passenger railway systems. *Wiley Encyclopedia of Operations Research and Management Science*, 2011. doi:10.1002/9780470400531.eorms0647.
- [6] F. Corman and L. Meng. A Review of Online Dynamic Models and Algorithms for Railway Traffic Management. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1274–1284, 2014. doi:10.1109/TITS.2014.2358392.
- [7] M. Forsgren, M. Aronsson, and S. Gestrelus. Maintaining tracks and traffic flow at the same time. *Journal of Rail Transport Planning & Management*, 3(3):111–123, August 2013. doi:10.1016/j.jrtpm.2013.11.001.
- [8] T. Kalinowski, T. Lidén, and H. Waterer. Tight MIP formulations for bounded length cyclic sequences. *ArXiv e-prints*, September 2018. arXiv:1809.01773.
- [9] T. Lidén. Railway Infrastructure Maintenance — A Survey of Planning Problems and Conducted Research. *Transportation Research Procedia*, 10:574–583, 2015. doi:10.1016/j.trpro.2015.09.011.
- [10] T. Lidén. Reformulations for Integrated Planning of Railway Traffic and Network Maintenance. In *18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018)*, volume 65 of *OpenAccess Series in Informatics (OASISs)*, pages 1:1–1:10, Dagstuhl, Germany, 2018. doi:10.4230/OASISs.ATMOS.2018.1.
- [11] T. Lidén. Coordinating maintenance windows and train traffic — a case study. *Submitted for journal publication*, July 2019.
- [12] T. Lidén and M. Joborn. An optimization model for integrated planning of railway traffic and network maintenance. *Transportation Research Part C: Emerging Technologies*, 74:327–347, January 2017. doi:10.1016/j.trc.2016.11.016.
- [13] T. Lidén, T. Kalinowski, and H. Waterer. Resource considerations for integrated planning of railway traffic and maintenance windows. *Journal of Rail Transport Planning & Management*, 8.1:1–15, February 2018. doi:10.1016/j.jrtpm.2018.02.001.
- [14] X. Luan, J. Miao, L. Meng, F. Corman, and G. Lodewijks. Integrated optimization on train scheduling and preventive maintenance time slots planning. *Transportation Research Part C: Emerging Technologies*, 80:329–359, July 2017. doi:10.1016/j.trc.2017.04.010.
- [15] Y. Pochet and L. Wolsey. *Production planning by mixed integer programming*. Springer series in operations research and financial engineering. Springer, 2006. doi:10.1007/0-387-33477-7.