Reformulations for Integrated Planning of Railway Traffic and Network Maintenance

Tomas Lidén
Linköping University, Department of Science and Technology
Norrköping SE-601 74, Sweden
tomas.liden@liu.se
https://orcid.org/0000-0002-1643-6365

Abstract

This paper addresses the capacity planning 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, more instances can be solved to optimality within a given time limit and the optimality gap can be reduced quicker.

2012 ACM Subject Classification Applied computing → Transportation

Keywords and phrases Railway scheduling, Maintenance planning, Optimization

Digital Object Identifier 10.4230/OASIcs.ATMOS.2018.1

Funding This research is 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".

Acknowledgements The work was performed during a research visit to University of Newcastle, Australia, where Thomas Kalinowski and Hamish Waterer gave invaluable help, ideas and inspiration.

1 Introduction

Scheduling access to a railway infrastructure, commonly termed *capacity planning*, is the core tactical planning problem for all railway systems and can be seen as a resource planning for the infrastructure components (stations, lines, yards, tracks, switches, signalling blocks etc). Capacity planning includes producing a timetable for the train traffic and 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], there has been much less focus on maintenance planning [8]. 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, 11] 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 train services on a railway network. An example of how such plans

18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018).
Editors: Ralf Borndörfer and Sabine Storandt; Article No. 1; pp. 1:1–1:10
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

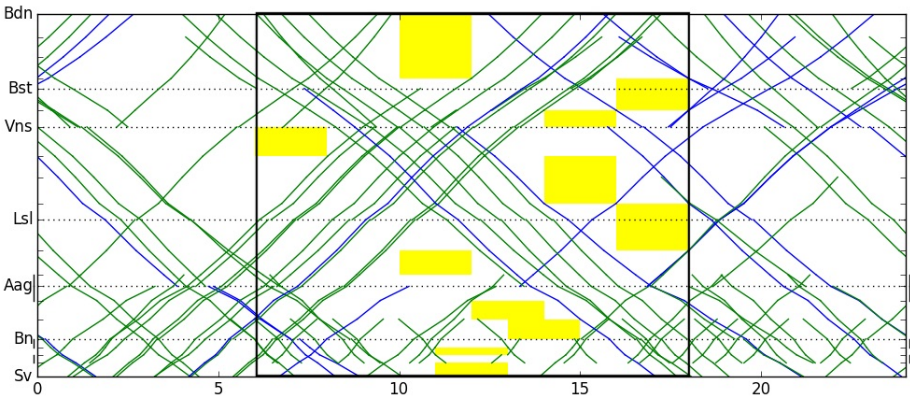

Figure 1 Train and work graph example.

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 [9]. 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 [10]. The latter model also has a stronger formulation for the train and maintenance window scheduling, but that paper only briefly summarizes these model improvements. In this paper we describe and compare the two formulations more closely.

The original model, which we here denote with ORG, uses cumulative train entry/exit variables (for each link and time period) and implicit link usage variables for the train scheduling. The improved model, which we denote with IMP, instead uses binary train entry/exit detection variables and explicit link usage variables. These changes increase the number of variables and decrease the number of constraints. The linear relaxation does not become tighter, but the MILP solver benefits from having more binary variables to branch on, a better linking of constraints and some possibilities for pruning due to the binary restrictions.

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 [12, section 11.4, pp 341–343] and mathematically analysed in [13]. These improvements do make the linear relaxation tighter and in addition the MILP solver presolve method is able to reduce the size more effectively.

The net effect of these model improvements is that the solution performance gets better, more instances can be solved to optimality within a given time limit and the optimality gap can be reduced quicker.

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

2 Mathematical formulation

2.1 Notation and model structure

The railway network is modelled by a link set L, where a subset of links $L^M \subseteq L$ shall have maintenance windows. The scheduling problem has a planning horizon of length H, divided into a sequence $T = \{0, \ldots, H-1\}$ of unit size time periods $t \in T$, each covering real-valued event times between t and $t+1$.

For each link $l \in L^M$, a required number of time periods shall have maintenance windows. The scheduling shall be done according to a set W_l of window options where each option $o \in W_l$ is defined by 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.

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 trains shall be done by selecting one route $r \in R_s$ and deciding entry and exit times for each link in that route, such that all event times are within the scheduling window defined by $T_s \subseteq T$.

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^h_{lt}	number of train services traversing link l in direction h during time period t

The ORG formulation uses cumulative x variables, which we denote by $\bar{x}^+_{slt}, \bar{x}^-_{slt}$, and implicit u variables, while the IMP formulation uses binary x variables and explicit u. 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:

$$
\begin{aligned}
\text{minimize } & c(\mathbf{z}, \mathbf{e}, \mathbf{y}, \mathbf{v}) && (1)\\
\text{subject to } & \mathbf{A}(\mathbf{z}, \mathbf{e}, \mathbf{x}, \mathbf{u})^{\,route} && (2)\\
& \mathbf{A}(\mathbf{z}, \mathbf{e})^{\,trains} && (3)\\
& \mathbf{A}(\mathbf{w}, \mathbf{y}, \mathbf{v})^{\,maintenance} && (4)\\
& \mathbf{A}(\mathbf{u}, \mathbf{n}, \mathbf{y})^{\,capacity} && (5)\\
& \text{variable types and bounds}
\end{aligned}
$$

where $c(..)$ is the objective function and $\mathbf{A}(..)$ 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

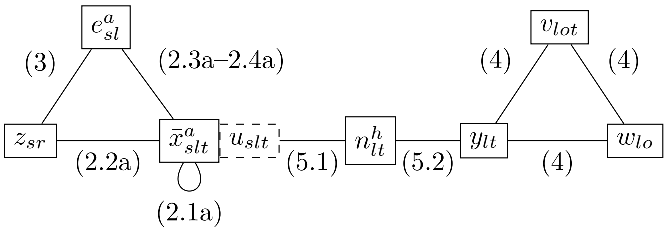

Figure 2 Variable and constraint graph – ORG formulation.

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 constraints (2) and (4), which will be described in the following sections. For the details regarding the objective function and other constraints, we refer to [10].

2.2 Train scheduling

The ORG formulation uses cumulative variables $\bar{x}^+_{slt}, \bar{x}^-_{slt}$, which takes value 1 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 $\bar{x}^a_{sl,t-1} = 0$ for $t = 0$.

The constraint set (2) for ORG is:

$$\bar{x}^a_{slt} \geq \bar{x}^a_{sl,t-1} \qquad \forall s \in S, l \in L_s, t \in T_s, a \in \{+,-\} \tag{2.1a}$$

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

$$e^a_{sl} \geq \sum_{t \in T_s} LB^a_t(\bar{x}^a_{slt} - \bar{x}^a_{sl,t-1}) \qquad \forall s \in S, l \in L_s, a \in \{+,-\} \tag{2.3a}$$

$$e^a_{sl} \leq \sum_{t \in T_s} UB^a_t(\bar{x}^a_{slt} - \bar{x}^a_{sl,t-1}) \qquad \forall s \in S, l \in L_s, a \in \{+,-\} \tag{2.4a}$$

where LB^a_t and UB^a_t 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 will be 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 correspond to cliques in the graph as indicated in the figure.

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

$$x^a_{slt} = \bar{x}^a_{slt} - \bar{x}^a_{sl,t-1}$$

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

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

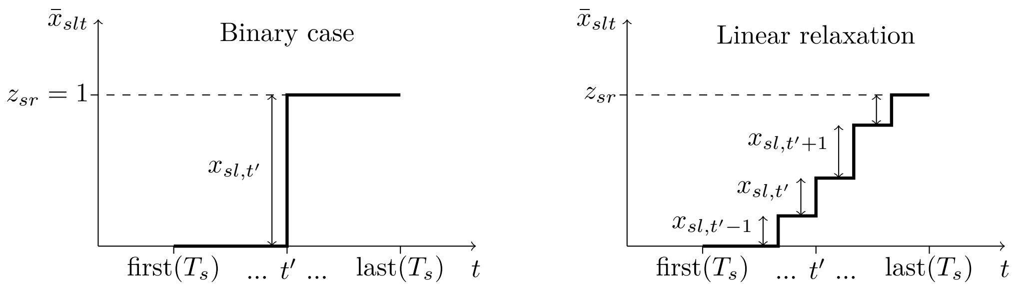

Figure 3 Relation between cumulative variables $\bar{x}$ and detection variables x.

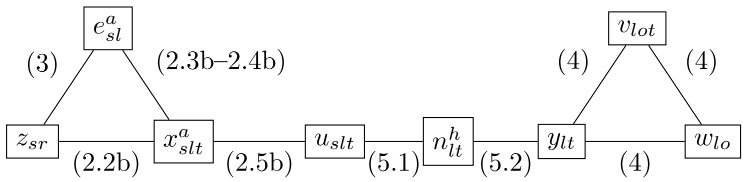

Figure 4 Variable and constraint graph – IMP formulation.

we transform constraints (2.2a–2.4a) and introduce explicit u variables to get the IMP formulation of (2):

$$\sum_{t \in T_s} x^a_{slt} = \sum_{r \in R_s : l \in L_r} z_{sr} \qquad \forall s \in S, l \in L_s, a \in \{+,-\} \tag{2.2b}$$

$$e^a_{sl} \geq \sum_{t \in T_s} LB^a_t x^a_{slt} \qquad \forall s \in S, l \in L_s, a \in \{+,-\} \tag{2.3b}$$

$$e^a_{sl} \leq \sum_{t \in T_s} UB^a_t x^a_{slt} \qquad \forall s \in S, l \in L_s, a \in \{+,-\} \tag{2.4b}$$

$$u_{slt} = \sum_{t' \in T_s : t' \leq t} x^+_{slt'} - \sum_{t' \in T_s : t' \leq t-1} x^-_{slt'} \qquad \forall s \in S, l \in L_s, t \in T_s \tag{2.5b}$$

Note that the cumulative constraints disappear, but that we have new constraints (2.5b) for the usage variables. The IMP formulation will have $|S|\,|L_s|\,|T_s|$ more variables but $|S|\,|L_s|\,|T_s|$ less constraints as compared to ORG. Also the train counting constraints ($n^h_{lt} = \sum u_{slt}$) will operate on the explicit u variables and hence contain fewer elements in IMP as compared to ORG.

The increase in variables might be a drawback, but also gives the MILP solver an opportunity for more pruning (due to the binary restriction of u) and another set of variables to branch on during the branch and bound procedure.

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

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

$$\sum_{o \in W_l} w_{lo} = 1 \qquad \forall l \in L^M \tag{4.1}$$

$$\sum_{t \in T} v_{lot} \geq \eta_o w_{lo} \qquad \forall l \in L^M, o \in W_l \tag{4.2}$$

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

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

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

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

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

Constraint (4.1) ensures that exactly on 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 coupling constraints (4.3a,4.5a,4.6a) can be aggregated, by utilising the fact that exactly one window option must be selected. Hence, since only one v_{lot} variable for each l, t combination can be non-zero, IMP uses summations over the window options, as follows:

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

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

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

Next, we make use of a model for bounded on/off sequences, presented in [13], 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:

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

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

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

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

The constraints (4.7b)-(4.8b) enforce that each window should span exactly θ_o time periods. 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. If there would be requirements for larger separation between windows on each link, the LHS should be a suitable forward going sum over 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. Here we can only aggregate over the window options for constraints (4.7b) (and (4.9b)).

3 Computational results

The same set of synthetic test instances as in [9] 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 one (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.

The evaluations have been done in two steps. First, various alternatives for the train scheduling formulation have been tested. These tests were made with Gurobi 6.0.5 as MIP solver on a MacBook Pro with a 2,6 GHz Intel Core i5 processor, 8 GB 1600 MHz DDR3 memory and OSX 10.10.5. The formulation with the best performance, as presented in Section 2.2, was then used when evaluating various alternatives for the maintenance scheduling part. The latter tests were made with Gurobi 6.5 as MIP solver on a Dell PowerEdge R710 rack server with dual hex core 3.06GHz Intel Xeon X5675 processors and 96GB RAM running Red Hat Enterprise Linux 6. For all tests, a maximum computation time of 3600 seconds have been used. Most tests have a relative MIP gap tolerance of 0.001 (0.1%) while some of the smaller instances (L1–L5 and N1–N2) have 0.01%. All other options have been left at their default values.

The computational results are presented in Table 1, where the left part gives instance properties, the middle part lists solution statistics for the four alternatives

(a) ORG - the original formulation, run with Gurobi 6.0.5
(b) T60 - the improved train scheduling formulation with Gurobi 6.0.5
(c) T65 - the improved train scheduling formulation with Gurobi 6.5
(d) IMP - the complete improved formulation with Gurobi 6.5

and the right part lists the absolute improvement in initial LP value as compared to the ORG formulation.

There is a clear improvement in solving performance – both solution times and remaining MIP gaps are reduced. Also, we see that IMP is tighter since the initial LP value is increasing (the small increases for some T65 instances are caused by a more efficient pre-solve).

To further illustrate the improvements, performance profile plots are used which show the accumulated number of instances (on the vertical axis) reaching a certain level of quality measure (on the logarithmic horizontal axis) – normalised as a factor of the best outcome for all alternatives. Thus an alternative have better performance when being above (= more instances) and to the left (= better quality) of another curve. In Figure 5 the time for

Table 1 Instances, properties, performance (time (seconds) to reach optimality or else remaining gap after 3600 seconds) and improvement in initial LP value vs ORG.

	Properties			Performance (sol. time / rem. gap)				LP improvement		
Case	$\lvert L\rvert$	$\lvert T\rvert$	$\lvert S\rvert$	ORG	T60	T65	IMP	T60	T65	IMP
L1	4	5	20	66	15	3	4	0	0	0
L2	4	5	20	181	18	4	4	0	0	0
L3	4	12	40	0.15%	255	102	46	0	0	0.6
L4	4	12	40	2	4	2	1	0	0	0.4
L5	9	24	40	0.16%	2878	3415	1796	0	0	0.75
L6	9	48	80	0.77%	0.50%	0.21%	0.13%	0	0	0.95
L7	18	24	80	31.0%	1.24%	0.90%	0.53%	0	0	1.35
L8	18	96	160	32.0%	17.6%	1.62%	1.14%	0	0	0.55
L9	25	168	350	265%	201%	170%	64.4%	0	0	2.76
N1	9	5	20	7	9	5	3	0	0.02	0.02
N2	9	24	50	42	25	6	5	0	0.12	0.27
N3	9	48	100	97	179	193	23	0	0	0.98
N4	9	96	200	1209	535	435	264	0	0.01	0.28
N5	9	168	350	0.14%	0.13%	3600	907	0	0	0.92

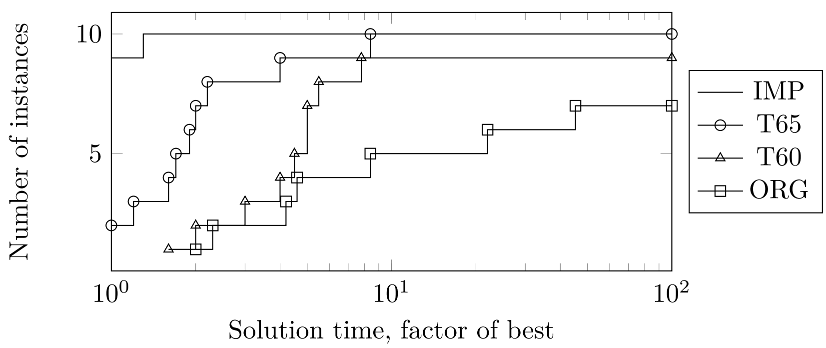

Figure 5 Performance profile – time to reach optimality.

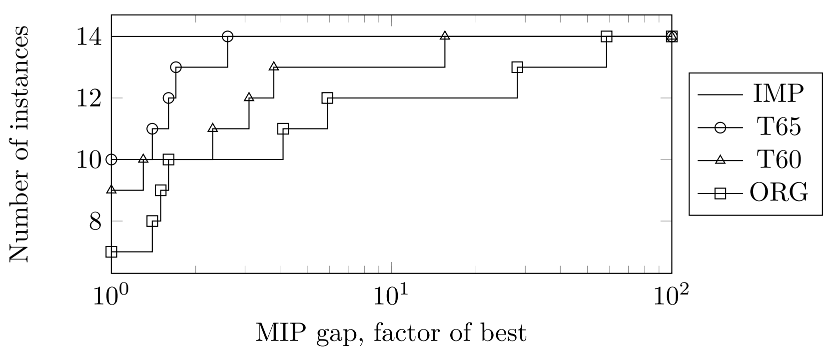

Figure 6 Performance profile – final MIP gap.

reaching optimality is the quality measure, while Figure 6 shows the remaining MIP gap for those instances that have not been solved to optimality. The improvement obtained for each step is clear – including the performance gain when changing solver version and computer platform.

The net result is that three more instances (L3, L5, N5) are solved to optimality, the optimal solutions are reached quicker (with a speed up between 2 and 10 times) and two more instances (L7, L8) are solved to a MIP gap $< 1.5\%$, 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 [13]).

These improvements have made it possible to extend the model with maintenance resource considerations (see [10]), and in recent work the models have also been applied to real world problems of realistic size – which will be reported in the presentation. 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.

References

1 A. R. Albrecht, D. M. Panton, and D. H. Lee. Rescheduling rail networks with maintenance disruptions using problem space search. *Computers & Operations Research*, 40(3):703–712, 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, 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, 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. Gestrelius. Maintaining tracks and traffic flow at the same time. *Journal of Rail Transport Planning & Management*, 3(3):111–123, 2013. doi:10.1016/j.jrtpm.2013.11.001.

8 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.

9 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, 2017. doi:10.1016/j.trc.2016.11.016.

10 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, 2018. doi:10.1016/j.jrtpm.2018.02.001.

11 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, 2017. doi:10.1016/j.trc.2017.04.010.

12 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.

13 M. Queyranne and L. Wolsey. Tight MIP formulations for bounded up/down times and interval-dependent start-ups. *Mathematical Programming*, 164(1-2):129–155, 2017. doi:10.1007/s10107-016-1079-2.