Integrating Passengers' Routes in Periodic Timetabling: A SAT approach*

Philine Gattermann[1], Peter Großmann[2], Karl Nachtigall[3], and Anita Schöbel[4]

1 Institut für Numerische und Angewandte Mathematik, Georg August University Göttingen, Göttingen, Germany
p.gattermann@math.uni-goettingen.de
2 Chair for Traffic Flow Science, TU Dresden, Dresden, Germany
peter.grossmann@tu-dresden.de
3 Chair for Traffic Flow Science, TU Dresden, Dresden, Germany
karl.nachtigall@tu-dresden.de
4 Institut für Numerische und Angewandte Mathematik, Georg August University Göttingen, Göttingen, Germany
schoebel@math.uni-goettingen.de

Abstract

The periodic event scheduling problem (PESP) is a well studied problem known as intrinsically hard. Its main application is for designing periodic timetables in public transportation. To this end, the passengers' paths are required as input data. This is a drawback since the final paths which are used by the passengers depend on the timetable to be designed. Including the passengers' routing in the PESP hence improves the quality of the resulting timetables. However, this makes PESP even harder.

Formulating the PESP as satisfiability problem and using SAT solvers for its solution has been shown to be a highly promising approach. The goal of this paper is to exploit if SAT solvers can also be used for the problem of integrated timetabling and passenger routing. In our model of the integrated problem we distribute origin-destination (OD) pairs temporally through the network by using time-slices in order to make the resulting model more realistic. We present a formulation of this integrated problem as integer program which we are able to transform to a satisfiability problem. We tested the latter formulation within numerical experiments, which are performed on Germany's long-distance passenger railway network. The computation's analysis in which we compare the integrated approach with the traditional one with fixed passengers' weights, show promising results for future scientific investigations.

1998 ACM Subject Classification G.1.6 Optimization, G.2.2 Graph Theory, G.2.3 Applications

Keywords and phrases PESP, Timetabling, Public Transport, Passengers' Routes, SAT

Digital Object Identifier 10.4230/OASIcs.ATMOS.2016.3

1 Introduction

Since the introduction of the *Periodic Event Scheduling Problem (PESP)* in [26] this problem has been investigated and analyzed in numerous publications. Early contributions about the PESP in the context of timetabling include [16, 18, 19]. They show NP-hardness of even the feasibility problem and develop different formulations as integer programs. It

* This work was partially supported by DFG under grant SCHO1140/8-1

16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'16).
Editors: Marc Goerigk and Renato Werneck; Article No. 3; pp. 3:1–3:15
Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

turns out that formulations using cycle-bases are the most efficient ones leading to several further publications, e.g., [21, 20, 10, 22, 14, 11]. Success stories for timetabling in practice based on PESP models are presented in [7, 12] where the Dutch railway timetable and the timetable of Berlin Underground have been computed. Besides using integer programming, the modulo simplex [17, 2] is a heuristic approach for tackling PESP. Recently, [8, 4] showed that SAT solvers can be used successfully for solving PESP instances in the context of railway timetabling.

When using PESP models for timetabling, it is always assumed that the precise passengers' paths are known beforehand, i.e., which lines a passenger takes and at which stations he or she wants to transfer. As noted in [24, 23] this is not realistic since the passengers' behavior crucially depends on the timetable which still has to be determined. Recently, [1] showed that the error which can be made by this assumption can be arbitrarily large theoretically and present a case study which shows that allowing a re-routing of passengers can improve the transfer waiting time of periodic timetables by more than 20%.

Our contribution. In this paper we study an integrated problem of finding a timetable and passengers' routes in which we distribute the passengers temporally using *time-slices*. We propose a formulation as satisfiability problem and study its computational behavior.

2 Definition of the integrated problem

When integrating timetabling and passenger routing we want to find a solution which optimizes the travel quality of the passengers. The travel quality of the passengers is usually measured as the sum of all traveling times over all passengers. For technical reasons we use a slightly different measure here, namely the speedup compared to a maximal travel time that a passenger is going to accept. In order to determine a passenger's travel time, they are routed through the network on shortest paths (according to the actual timetable) as part of the optimization. To account for a more realistic distribution of passengers, an OD-pair is distributed to different *time slices*. The time slice a passengers is allotted to specifies in which part of the planning period his or her journey is supposed to start. Changing to a different time slice is allowed but penalized in order to account for much shorter travel times when starting earlier or later than planned. Every passenger whose travel time exceeds the maximal one for the OD-pair is supposed to use another mode of transportation and is not counted towards the objective function.

Our model only uses data which can be supplied when only the public transportation network (V, E), with its set of stations V and direct connections E between them, the line plan $\mathcal{L}$ and the planning period T are known. We especially need maximal and minimal driving times $L_e^{\text{drive}}, U_e^{\text{drive}}$ for all edges, minimal and maximal waiting times $L_v^{\text{wait}}, U_v^{\text{wait}}$ in each stop as well as minimal and maximal transfer times $L_v^{\text{trans}}, U_v^{\text{trans}}$ in each station to define feasibility of a timetable. As mentioned above, we need origin-destination data $C_{u,v}^t$ for each time-slice $t \in \{1, \ldots, T_{u,v}\}$, where $T_{u,v}$ is the number of time slices for OD-pair (u, v), and a penalty $P_{u,v}^{t,t'}$ for changing the start of a journey from time slot t to t' as well as a maximal traveling time $D_{u,v}$ for each passenger. The *Timetabling Problem with Passenger Routing* hence is:

▶ **Definition 1.** For the input data mentioned above, find a timetable such that the speedup of the passengers routed along their shortest paths according to travel time and time slice changing penalty, is maximized.

In the following we model this problem more formally as integer program and as satisfiability problem. To this end, we first have to introduce the extended event-activity network (*extended EAN*) as common basis for both formulations.

2.1 Extended EAN

The basis of both the integer programming (IP) and the satisfiability (SAT) formulation for the integrated problem is an event-activity network, similar to the one used in a standard PESP-formulation (see, e.g., [16, 11]). First we define the basic EAN $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$:

$$\mathcal{E}^0 = \mathcal{E}^0_{\text{arr}} \cup \mathcal{E}^0_{\text{dep}}$$

as the set of all arrival and departures of all lines at all stations,

$$\mathcal{E}^0_{\text{arr}} = \{(v, l, \text{arr})\colon v \in V, v \in l, l \in \mathcal{L}\}$$

$$\mathcal{E}^0_{\text{dep}} = \{(v, l, \text{dep})\colon v \in V, v \in l, l \in \mathcal{L}\}$$

$$\mathcal{A}^0 = \mathcal{A}^0_{\text{drive}} \cup \mathcal{A}^0_{\text{wait}} \cup \mathcal{A}^0_{\text{trans}}$$

links the events in $\mathcal{E}^0$ by driving, waiting and transfer activities,

$$\mathcal{A}^0_{\text{drive}} = \{((v_1, l, \text{dep}), (v_2, l, \text{arr}))\colon \{v_1, v_2\} \in l, l \in \mathcal{L}\}$$

$$\mathcal{A}^0_{\text{wait}} = \{((v, l, \text{arr}), (v, l, \text{dep}))\colon v \in l, l \in \mathcal{L}\}$$

$$\mathcal{A}^0_{\text{trans}} = \{((v, l_1, \text{arr}), (v, l_2, \text{dep}))\colon v \in l_1, v \in l_2, l_1, l_2 \in \mathcal{L}\}.$$

Moreover, headway activities (which are not relevant for the passengers' paths) are used to model security distances between trains. The upper and lower bounds on the duration of the activities are set according to the underlying edges E of the public transportation network.

$$L_a = \begin{cases} L^{\text{drive}}_{\{v_1,v_2\}}, & \text{if } a = ((v_1, l, \text{dep}), (v_2, l, \text{arr})) \\ L^{\text{wait}}_v, & \text{if } a = ((v, l, \text{arr}), (v, l, \text{dep})) \\ L^{\text{trans}}_v, & \text{if } a = ((v, l_1, \text{arr}), (v, l_2, \text{dep})) \end{cases}$$

$$U_a = \begin{cases} U^{\text{drive}}_{\{v_1,v_2\}}, & \text{if } a = ((v_1, l, \text{dep}), (v_2, l, \text{arr})) \\ U^{\text{wait}}_v, & \text{if } a = ((v, l, \text{arr}), (v, l, \text{dep})) \\ U^{\text{trans}}_v, & \text{if } a = ((v, l_1, \text{arr}), (v, l_2, \text{dep})) \end{cases}$$

Additionally, we need nodes and arcs representing the OD-pairs. These nodes need not be scheduled in the timetable. Thus we get $\bar{\mathcal{N}} = (\bar{\mathcal{E}}, \bar{\mathcal{A}})$ with

$$\bar{\mathcal{E}} = \mathcal{E}^0 \cup \mathcal{E}^0_{\text{OD}}$$

$$\mathcal{E}^0_{\text{OD}} = \{(u, v, t, t', \text{source}), (u, v, t, \text{target})\colon u, v \in V, t, t' \in \{1, \dots, T_{u,v}\}\}$$

source and target nodes for passenger paths

$$\bar{\mathcal{A}} = \mathcal{A}^0 \cup \mathcal{A}^0_{\text{time}} \cup \mathcal{A}^0_{\text{to}} \cup \mathcal{A}^0_{\text{from}}$$

$$\mathcal{A}^0_{\text{time}} = \{((u, v, t, t, \text{source}), (u, v, t, t', \text{source}))\colon u, v, \in V, t \neq t' \in \{1, \dots, T_{u,v}\}\}$$

arcs for changing the time slice

$$\mathcal{A}^0_{\text{to}} = \{((u, v, t, t', \text{source}), (u, l, \text{dep}))\colon u \in l, u, v, \in V, t, t' \in \{1, \dots, T_{u,v}\}\}$$

acrs to get from a source node into the network

$$\mathcal{A}^0_{\text{from}} = \{((v, l, \text{arr}), (u, v, t, \text{target}))\colon v \in l, u, v, \in V, t \in \{1, \dots, T_{u,v}\}\}$$

arcs to get from the network to a source node.

Here, a node $(u, v, t, t', \text{source}) \in \mathcal{E}^0_{\text{OD}}$ corresponds to the OD-pair traveling from u to v which was appointed to start in time slice t and actually starts in t'.

3 An IP formulation for the integrated problem

For the IP-formulation for the integrated problem, we combine a PESP-formulation for timetabling with an IP-formulation for passenger flow for each OD-pair and each time slice.

Integer variables π_i are used to model the time appointed to event i with corresponding modulo parameters z_a. For the passengers we are using binary variables $x^t_{u,v}$ to determine if there is there a path from u to v starting in time slice t which is used and variables $z^{u,v,t}_a$ to decide if arc a is used by the passengers going from u to v starting in time slice t.

$$\begin{aligned}
\max \sum_{u,v,\in V} \sum_{t=1}^{T_{u,v}} C^t_{u,v} \big(D_{u,v} \cdot x^t_{u,v} - &\sum_{a=(i,j)\in\mathcal{A}^0} z^{u,v,t}_a \cdot (\pi_j - \pi_i + z_a \cdot T) \\
&- \sum_{a=((u,v,t,t',\text{source}),\bullet)\in\mathcal{A}^0_{\text{time}}} P^{t,t'}_{u,v} \cdot z^{u,v,t}_a\big)
\end{aligned} \tag{1}$$

$$\begin{aligned}
\pi_j - \pi_i + z_a \cdot T \geq& L_a && \forall a = (i,j) \in \mathcal{A}^0 && (2)\\
\pi_j - \pi_i + z_a \cdot T \leq& U_a && \forall a = (i,j) \in \mathcal{A}^0 && (3)\\
x^t_{u,v} \geq& z^{u,v,t}_a && \forall u,v \in V, t \in \{1,\dots,T_{u,v}\}, && (4)\\
&&& a \in \bar{\mathcal{A}}, l \in \mathcal{L}: \\
&&& a = (\bullet, (\bullet, l, \bullet)), a = ((\bullet, l, \bullet), \bullet) \\
A^{u,v,t} \cdot (z^{u,v,t}_a)_{a\in\bar{\mathcal{A}}} =& b^{u,v,t} && \forall u,v,\in V, t \in \{1,\dots,T_{u,v}\} && (5)\\
\pi_i \geq& z^{u,v,t}_a \cdot L^{t'}_{u,v} && \forall u,v \in V, t,t' \in \{1,\dots,T_{u,v}\}, && (6)\\
&&& a = ((u,v,t,t',\text{source}), i) \in \mathcal{A}^0_{\text{to}} \\
\pi_i \leq& U^{t'}_{u,v} + M \cdot (1 - z^{u,v,t}_a) && \forall u,v \in V, t,t' \in \{1,\dots,T_{u,v}\}, && (7)\\
&&& a = ((u,v,t,t',\text{source}), i) \in \mathcal{A}^0_{\text{to}} \\
\pi_i \in& \{0, T-1\} && \forall i \in \mathcal{E}^0 \\
z_a \in& \mathbb{Z} && \forall a \in \mathcal{A}^0 \\
z^{u,v,t}_a \in& \{0,1\} && \forall u,v,\in v, t \in \{1,\dots,T_{u,v}\}, a \in \bar{\mathcal{A}} \\
x^t_{u,v} \in& \{0,1\} && \forall u,v,\in v, t \in \{1,\dots,T_{u,v}\}
\end{aligned}$$

As the model is non-linear, the objective function (1) has to be linearized by substituting

$$z^{u,v,t}_a \cdot (\pi_j - \pi_i + z_a \cdot T) = d^{u,v,t}_a$$

with

$$\begin{aligned}
&d^{u,v,t}_a \geq 0 \\
&d^{u,v,t}_a \geq \pi_j - \pi_i + z_a \cdot T - (1 - z^{u,v,t}_a) \cdot M'
\end{aligned}$$

where M' is sufficiently large, e.g. $M' \geq \max_{a\in\mathcal{A}^0} U_a$.
Constraints (2) and (3) are the standard timetabling constraints while constraint (4) makes sure that an activity can only be used by a passenger, if a path for this passengers is

chosen at all. The routing of passengers is modeled by constraint (5). Here, $A^{u,v,t}$ is a node-arc-incidence-matrix and $b^{u,v,t}$ the corresponding demand vector:

$$A^{u,v,t} \in \{0, 1, -1\}^{|\bar{\mathcal{E}}| \times |\bar{\mathcal{A}}|}$$

$$a^{u,v,t}_{i,a} = \begin{cases} 1, & \text{if } a = (i,j) \in \mathcal{A}^0_{\text{time}} \cup \mathcal{A}^0_{\text{to}}, i = (u,v,t,t',\text{source}) \\ -1, & \text{if } a = (j,i) \in \mathcal{A}^0_{\text{time}} \cup \mathcal{A}^0_{\text{to}}, j = (u,v,t,t',\text{source}) \\ 1, & \text{if } a = (i,j) \in \mathcal{A}^0_{\text{from}}, j = (u,v,t,\text{target}) \\ -1, & \text{if } a = (j,i) \in \mathcal{A}^0_{\text{from}}, i = (u,v,t,\text{target}) \\ 1, & \text{if } a = (i,j) \in \mathcal{A}^0 \\ -1, & \text{if } a = (j,i) \in \mathcal{A}^0 \\ 0, & \text{otherwise} \end{cases}$$

$$b^{u,v,t} \in \{0, 1\}^{|\bar{\mathcal{E}}|}$$

$$b^{u,v,t}_i = \begin{cases} x^t_{u,v}, & \text{if } i = (u,v,t,t,\text{source}) \\ -x^t_{u,v}, & \text{if } i = (u,v,t,\text{target}) \\ 0, & \text{otherwise} \end{cases}$$

Constraints (6) and (7) make sure that the first event of a path starting in time slice t' lies in the correct time slice. Here $L^{t'}_{u,v} = (t'-1) \cdot \frac{T}{T_{u,v}}$ and $U^{t'}_{u,v} = t' \cdot \frac{T}{T_{u,v}} - 1$. M has to be sufficiently large, e.g. $M = T$ is large enough.

In case that all $x^t_{u,v}$ variables are set to one, the objective function minimizes the traveling time $\sum_{a=(i,j)\in\mathcal{A}^0} z^{u,v,t}_a \cdot (\pi_j - \pi_i + z_a \cdot T)$ and the penalty for changing a time slice $\sum_{a=((u,v,t,t',\text{source}),\bullet)\in\mathcal{A}^0_{\text{time}}} P^{t,t'}_{u,v} \cdot z^{u,v,t}_a)$. For technical reasons we, however, need an upper bound $D_{u,v}$ on the length of a passengers' path, and hence allow that a passenger is not routed at all if his or her shortest path exceeds this length. Since the contribution of such an non-routed passengers to the objective function is only zero, the model tries to avoid non-routed passengers such that this happens only in exceptional cases.

4 A SAT formulation for the integrated problem

Now we model the same problem as a *partial weighted MaxSAT* problem. Therefore, we have to formulate all constraints in conjunctive normal form and convert the objective into a set of clauses with positive weight.

To emphasize the similarities of the problems, the variables we are using will be mostly the same. We can directly use the binary variables $x^t_{u,v}$ for the usage of paths and $z^{u,v,t}_a$ for the usage of arcs. Due to the definition of the satisfiability problem we cannot use the integer variables π_i but have to substitute them for binary variables π^k_i which determine if $\pi_i \le k$ holds.

4.1 Modeling feasibility

At first we show how to model the feasibility of the Timetabling Problem with Passenger Routing as a SAT problem by extending the timetabling SAT model proposed in [5]. We will discuss all sets of constraints in detail in the following paragraphs.

4.1.1 Timetabling

As shown in [5], the timetabling constraints can be modeled as conjunction of two sets of clauses. One set, which we call $\Omega_{\mathcal{N}^0}$, is used for modeling the variable encoding and another set, called $\Psi_{\mathcal{N}^0}$, for modeling the constraints.

4.1.2 Modeling passenger routes

For the passenger routes we simply model whether a path from u to v is used for time slice t by the variables $x^t_{u,v}$. If this is the case, we also have to model the corresponding passenger path.
Therefore, we first have to make sure that for each OD-pair u, v and each time slice t a path starts, if one is chosen at all. This can be realized either by moving to a different time slice or by starting at a specified event in the allotted time slice.

$$\begin{aligned} & x^t_{u,v} \Rightarrow \bigvee_{a=((u,v,t,t,\text{source}),\bullet)\in\mathcal{A}^0_{\text{time}}\cup\mathcal{A}^0_{\text{to}}} z^{u,v,t}_a \\ \iff & \underbrace{\neg x^t_{u,v} \vee \bigvee_{a=((u,v,t,t,\text{source}),\bullet)\in\mathcal{A}^0_{\text{time}}\cup\mathcal{A}^0_{\text{to}}} z^{u,v,t}_a}_{enc\_start(u,v,t)} \\ & \text{for all } u, v \in V, t \in \{1, \ldots, T_{u,v}\} \end{aligned}$$

Additionally, we have to make sure that if an arc $a = ((u, v, t, t', \text{source}), i) \in \mathcal{A}^0_{\text{to}}$ is used, the target event $i \in \mathcal{E}^0$ lies in the correct time slice.

$$\begin{aligned} & z^{u,v,t}_a \Rightarrow \pi_i \in \{(t'-1)\cdot\frac{T}{T_{u,v}}, \ldots, t'\cdot\frac{T}{T_{u,v}} - 1\} \\ \iff & \neg z^{u,v,t}_a \vee (\neg\pi_i^{(t'-1)\cdot\frac{T}{T_{u,v}}-1} \wedge \pi_i^{t'\cdot\frac{T}{T_{u,v}}-1}) \\ \iff & \underbrace{(\neg z^{u,v,t}_a \vee \neg\pi_i^{(t'-1)\cdot\frac{T}{T_{u,v}}-1})}_{enc\_slice\_1(a,u,v,t')} \wedge \underbrace{(\neg z^{u,v,t}_a \vee \pi_i^{t'\cdot\frac{T}{T_{u,v}}-1})}_{enc\_slice\_2(a,u,v,t')} \\ & \text{for all } u, v \in V, t, t' \in \{1, \ldots, T_{u,v}\}, a = ((u, v, t, t', \text{source}), i) \in \mathcal{A}^0_{\text{to}} \end{aligned}$$

Next we have to ensure that if a path is started, this path continues throughout the network. Let $a = (i, j) \in \mathcal{A}^0 \cup \mathcal{A}^0_{\text{to}} \cup \mathcal{A}^0_{\text{time}}$.

$$\begin{aligned} & z^{u,v,t}_a \Rightarrow \bigvee_{a'=(j,k)\in\mathcal{A}^0\cup\mathcal{A}^0_{\text{to}}\cup\mathcal{A}^0_{\text{from}}} z^{u,v,t}_{a'} \iff \underbrace{\neg z^{u,v,t}_a \vee \bigvee_{a'=(j,k)\in\mathcal{A}^0\cup\mathcal{A}^0_{\text{to}}\cup\mathcal{A}^0_{\text{from}}} z^{u,v,t}_{a'}}_{enc\_continue(a,u,v,t)} \\ & \text{for all } u, v \in V, t \in \{1, \ldots, T_{u,v}\}, a = (i, j) \in \mathcal{A}^0 \cup \mathcal{A}^0_{\text{to}} \cup \mathcal{A}^0_{\text{time}} \end{aligned}$$

We also have to make sure that the path ends at a node (u, v, t, target).

$$\begin{aligned} & x^t_{u,v} \Rightarrow \bigvee_{a=(k,(u,v,t,\text{target}))\in\mathcal{A}^0_{\text{from}}} z^{u,v,t}_a \iff \underbrace{\neg x^t_{u,v} \vee \bigvee_{a=(k,(u,v,t,\text{target}))\in\mathcal{A}^0_{\text{from}}} z^{u,v,t}_a}_{enc\_stop(u,v,t)} \\ & \text{for all } u, v \in V, t \in \{1, \ldots, T_{u,v}\} \end{aligned}$$

In the end we have to ensure that there are no nodes where multiple arcs are used. First we make sure that each node $i \in \bar{\mathcal{E}}$ has only one successor.

$$\neg(\bigvee_{\substack{a,a' \in \bar{\mathcal{A}}:\\ a=(i,j), a'=(i,j')}} z_a^{u,v,t} \wedge z_{a'}^{u,v,t}) \iff \bigwedge_{\substack{a,a' \in \bar{\mathcal{A}}:\\ a=(i,j), a'=(i,j')}} \underbrace{(\neg z_a^{u,v,t} \vee \neg z_{a'}^{u,v,t})}_{enc\_only\_one\_successor(a,a')}$$

for all $u, v \in V, t \in \{1, \ldots, T_{u,v}\}, i \in \bar{\mathcal{E}}$

Next we make sure that each node $j \in \bar{\mathcal{E}}$ has only one predecessor.

$$\neg(\bigvee_{\substack{a,a' \in \bar{\mathcal{A}}:\\ a=(i,j), a'=(i',j)}} z_a^{u,v,t} \wedge z_{a'}^{u,v,t}) \iff \bigwedge_{\substack{a,a' \in \bar{\mathcal{A}}:\\ a=(i,j), a'=(i',j)}} \underbrace{(\neg z_a^{u,v,t} \vee \neg z_{a'}^{u,v,t})}_{enc\_only\_one\_predecessor(a,a')}$$

for all $u, v \in V, t \in \{1, \ldots, T_{u,v}\}, j \in \bar{\mathcal{E}}$

To model the whole passenger behavior we get the following.

$$\begin{aligned}\Theta_{\mathcal{N}^0} : = \bigwedge_{u,v \in V} \bigwedge_{t \in \{1,\ldots,T_{u,v}\}} \Big(& enc\_start(u,v,t) \\ & \bigwedge_{a \in \mathcal{A}^0_{\text{to}}} enc\_slice\_1(a,u,v,t) \wedge enc\_slice\_2(a,u,v,t) \\ & \bigwedge_{a \in \mathcal{A}^0 \cup \mathcal{A}^0_{\text{to}} \cup \mathcal{A}^0_{\text{time}}} enc\_continue(a,u,v,t) \\ & \wedge\, enc\_stop(u,v,t) \\ & \bigwedge_{i \in \bar{\mathcal{E}}} \bigwedge_{\substack{a,a' \in \bar{\mathcal{A}}:\\ a=(i,j), a'=(i,j')}} enc\_only\_one\_successor(a,a') \\ & \bigwedge_{j \in \bar{\mathcal{E}}} \bigwedge_{\substack{a,a' \in \bar{\mathcal{A}}:\\ a=(i,j), a'=(i',j)}} enc\_only\_one\_predecessor(a,a') \Big)\end{aligned}$$

Together, the feasibility can be modeled as

$$\Omega_{\mathcal{N}^0} \wedge \Psi_{\mathcal{N}^0} \wedge \Theta_{\mathcal{N}^0},$$

i.e., by a conjunction of clauses. Thus, the feasibility of the Timetabling Problem with Passenger Routing can be modeled as a SAT problem.

Note that the number of clauses needed for passenger routing can be reduced in a preprocessing step. This process is described in more detail in the experimental evaluation in Section 5.

4.1.3 Considering only one time slice

If only one time slice is considered, the index t is not needed for any of the variables. The arc set $\mathcal{A}^0_{\text{time}}$ reduces to the empty set. Additionally the clauses *enc_slice_1* and *enc_slice_2* are not needed anymore.

4.2 Objective function

It remains to show that the objective function can be written as a set of weighted clauses, such that the Timetabling Problem with Passenger Routing can be formulated as a partial weighted MaxSAT problem. We refer to the following theorem and its proof which can be found in the appendix.

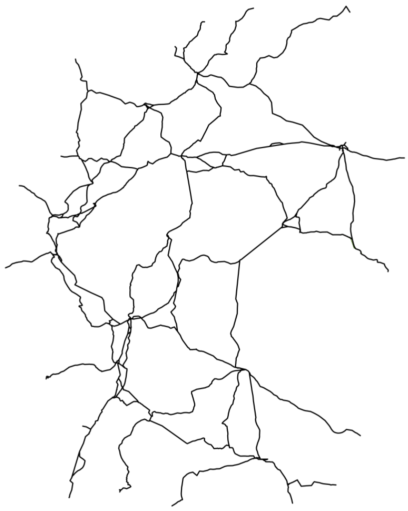

Figure 1 Line plan of Germany's inter city network.

▶ **Theorem 2.** *The Timetabling Problem with Passenger Routing can be formulated as a partial weighted MaxSAT problem.*

5 Experiments

The MaxSAT model introduced in Section 4 is implemented and the experiments are evaluated on an Intel® Core™ i7-4790K CPU with 32 GB RAM. However, the memory limit has never been reached for any instance. As MaxSAT Solver we apply the solver *open-wbo* [15]. As IP solver we use *Gurobi 6.0.3* [6] which is used with 4 CPU cores.

The line plan in our experiments is fixed as input. The periodic event-activity network is generated automatically from the given input data. This is necessary, as large timetabling problems can consist of up to one million activities and ten thousands of events, which cannot be calculated manually. The program automatically assigns an optimal route on the track layout to each train and calculates the running times within seconds. All minimum headways are calculated individually based on microscopical infrastructure data and are part of the PESP instance as well as symmetry constraints for pairwise connected train paths [13]. For details for encoding symmetry constraints we refer to the literature [3] that basically follows the same encoding as $enc\_rec(A)$, $A \in \zeta(a)$ shown in the previous sections.

In our instance, the German long-distance passenger railway network is examined, which in our scenario has 158 periodic lines and 181 stations. The macroscopic network is visualized in Figure 1. The PESP instance consists of 1 176 periodic events and 10 651 (periodic) activities. For comparison we also use the traditional approach in which

- first, the passengers are routed on shortest paths through the network (where the lower bounds L_a of the activities a are used as edge weights),
- second, a timetable is computed minimizing the weighted slacks on driving, waiting and transfer activities. (In our implementation the driving times are fixed to their lower bounds, i.e., $L_a = U_a$ for driving activities $a \in \mathcal{A}^0_{\text{drive}}$.) The resulting problem is then a traditional PESP including headway constraints which is solved by the above mentioned IP solver. The outcome is an optimal timetable π^*.

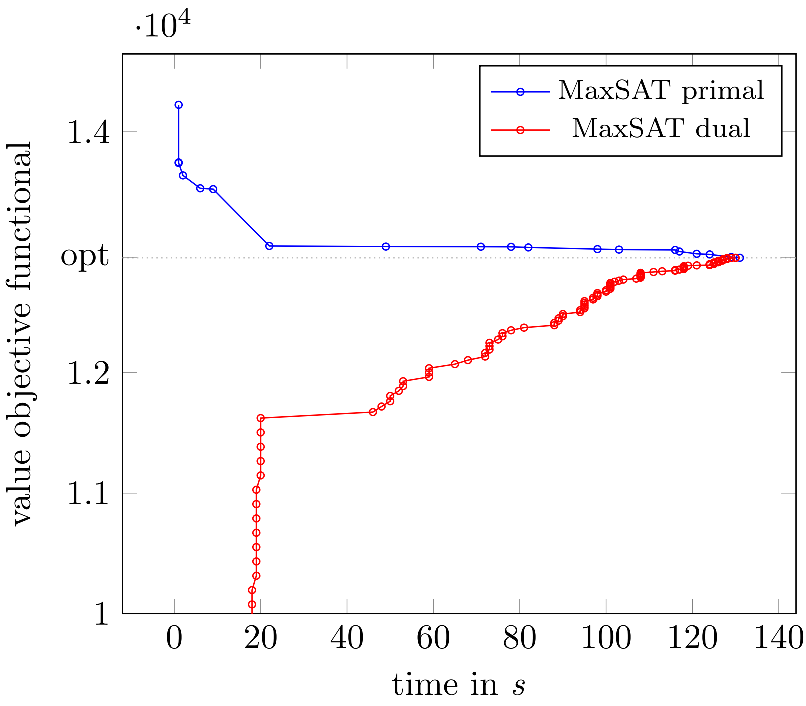

Figure 2 Graph of the computation for instance ic_1 of the integrated approach.

- Finally, for the evaluation of the optimal timetable π^* from the second step we proceed as follows: We use this timetable as given in the integrated formulation, i.e., we determine the best routes for the passengers with respect to this timetable and evaluate the sum of their traveling times.

For a fair comparison we use a single time slice such that $T_{u,v} = 1$ for all regarded OD-pairs (u, v).

In order to reduce the number of possible constraints we proceed as follows. For every OD-pair (u, v) we do the following: We search for the fastest path (again with respect to the lower bounds L_a) between u and v and then only add the constraints for paths that deviate at most by a given detour factor. For the experiments, we choose a maximum detour factor of 1.2. This seems reasonable, especially in terms of long-distance train networks.

The computational times contain both the encoding times and the solver times. Nevertheless, in Figure 2 and in Figure 3 just the solver times are shown, since the encoding times for large networks are neglectable. Note that the solvers formulate the problem as minimization problem and show the sum of weighted violated clauses which is displayed in the graphs.

The number of OD-pairs in the first run (ic_1) is 38, in order to experimentally validate the method. These are the most important OD-pairs in Germany. The value of the objective function (1) in minutes of passenger traveling time is 1 219 which in this case (accidentally) equals the sum of weighted lower bounds, i.e., the theoretically best travel times for all given OD-pairs. Thus, no better timetable is possible. If we compare this to the traditional approach, which has an objective value of 1 279, we can conclude that the integrated approach results as expected in better results. Regarding only the travel times without the weights results in an improvement of 60 min from the traditional approach to the integrated one.

We also compared the computation times (encoding and optimization):

- For the integrated model using the SAT formulation the computation time was 133 s.

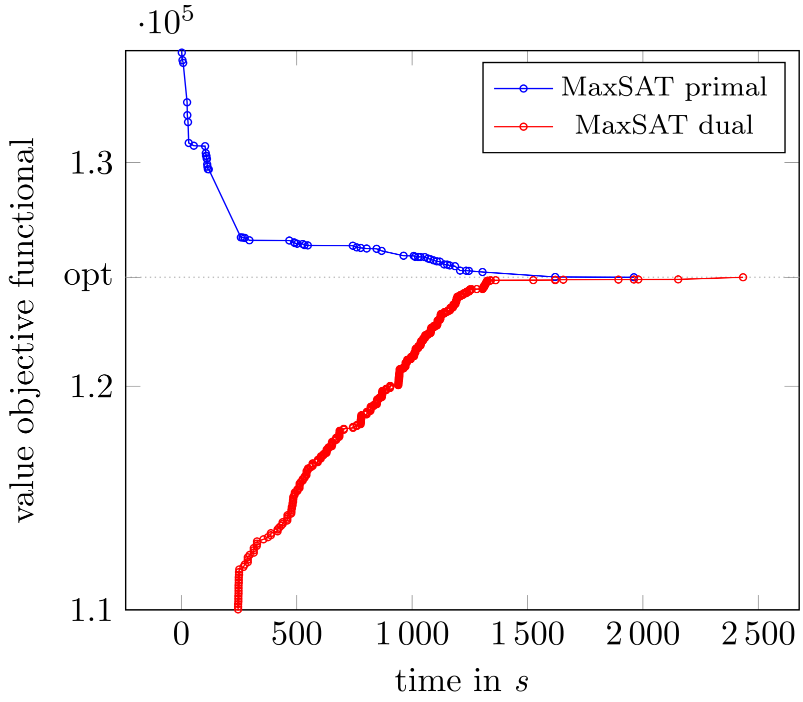

Figure 3 Graph of the computation for instance ic_2 of the integrated approach.

Table 1 PESP instances and their results.

	OD-pairs	objective value slack		objective value traveling time	
instance		integrated	traditional	integrated	traditional
ic_1	38	0	60	1 219	1 279
ic_2	192	141	572	13 515	13 946

For the traditional approach using an IP solver the computation time was 3 075 s. Hence, comparing the computational times for this instance, we get even better results with the integrated approach. This is probably due to the fact that SAT solvers perform better on PESP instances than integer programming solvers, see [5]. This promising result directly leads to the question whether more OD-pairs may be considered, which is investigated in the sequel.

In the second run (ic_2) we had a total number of 192 OD-pairs, which are yet again the top most important OD-pairs in Germany. The computation times are 2 453 s and 620 s for the integrated and traditional method, respectively. Evaluating the objective functional's value (1) and then computing the sum of traveling times for all OD-pairs yields 13 515 for the new and 13 946 for the traditional approach. The absolute lower bound, i. e., the sum of weighted minimum travel times for the OD-pairs is 13 374. As usual we hence evaluate the sum of slack times on all passengers' paths, i.e., the differences of absolute lower bound and the objective values. These are 141 for the integrated approach and 572 for the traditional approach. Cutting off the passenger weights and comparing the difference in travel time for both approaches conducts in an improvement of 301 min.

The results of the computations for both instances is provided in Table 1. Note that both instances were solved to optimality, so the duality gap is zero (and hence not listed). It can be concluded that at least for these first experiments the new approach leads always to better objective values compared to the traditional approach. Looking at the slack times the passengers may save, these reductions are rather large.

Nevertheless, the computational times for the traditional approach seem to vary depending on the instance. It should be mentioned that increasing the number of OD-pairs heavily increases the optimization times and hence, further methods or possible encoding improvements should be applied. Possible variable reduction methods are shortly discussed in Section 6.

6 Conclusion

In this paper, we provided an integer programming and a satisfiability formulation for the integrated problem of finding a periodic timetable and optimal passengers' paths simultaneously. We use time slices to distribute the passengers temporally.

The results on a restricted set of OD-pairs clearly show that the newly introduced, integrated method yields better objective values compared to the traditional approach. This is a promising position for regarding more OD-pairs. However, currently the computational times increase drastically with the number of OD-pairs such that no good objective value can be found in a reasonable time.

Nevertheless, the computational experiments have shown that there exists high potential for improvements. Firstly, we reduced the number of variables by reducing the possible path constraints with a detour factor. Secondly, by reducing the upper bounds of the transfer activities, the number of variables in the SAT formulation can be reduced as well.

Furthermore, we suggest the following possibilities for handling more OD-pairs: Currently, the lower and upper bounds of the constraints are coded in minutes which yields many binary variables in the resulting constraints in the SAT formulation. This leaves a lot of room for cutting off variables in two ways. On the one hand, we can reduce the search space – and not the solution space – by applying constraint propagation [16] and eliminating all variables that are no longer part of a constraint. This technique can even be applied for the possible routes with their possible detours. This results in better constraints' lower bounds for the path search which eventually results in fewer constraints per OD-pair [9]. On the other hand, from an engineering perspective we could also reduce the solution space by cutting off solutions that seem to be irrelevant in real-world scenarios. Therefore, we suggest a scaled variable encoding for the constraints to be optimized, which has a high granularity on the lower parts and a coarse-grained granularity on the higher parts of the constraints feasible areas. However, each variable's weight has to be adopted in the objective (1). The reasoning is that flows that are already badly fulfilled, e. g. contain transfer times above 60 min, are for better primal bounds not important, since the resulting solutions will be avoided anyway.

In future work we will implement the integrated approach as IP model and compare the computation times of state-of-the-art IP solvers to the state-of-the-art MaxSAT solvers. Also the number and distribution of the time-slices are subject of further experiments.

Finally, the SAT formulation provided in this paper can be easily extended to also include planning the lines (for a survey on line planning, see [25]), i.e., for modeling the problem of integrated line planning and timetabling. To this end, all potential lines from a given line pool have to be included in the formulation, and decision variables determine if a line is used (and should hence get a timetable) or not. We currently work on an implementation of this integrated formulation to make a step forward to integrated planning in public transportation.

All in all, it can be concluded that the introduced, integrated approach provides a promising scientific outlook that could highly improve travel times for passengers in periodic public railway transport networks in real-world scenarios.

References

1 R. Borndörfer, H. Hoppmann, and M. Karbstein. Timetabling and passenger routing in public transport. In *Proceedings of the 13th Conference on Advanced Systems in Public Transport (CASPT) 2015*, 2015.

2 M. Goerigk and A. Schöbel. Improving the modulo simplex algorithm for large-scale periodic timetabling. *Computers and Operations Research*, 40(5):1363–1370, 2013.

3 P. Großmann. Polynomial reduction from PESP to SAT. Technical Report 4, Technische Universität Dresden, Germany, October 2011.

4 P. Großmann, J. Opitz, R. Weiß, and M. Kümmling. On resolving infeasible periodic event networks. In *Proceedings of the 13th Conference on Advanced Systems in Public Transport (CASPT) 2015*. Erasmus University, 2015.

5 Peter Großmann, Steffen Hölldobler, Norbert Manthey, Karl Nachtigall, Jens Opitz, and Peter Steinke. Solving periodic event scheduling problems with SAT. In *Advanced Research in Applied Artificial Intelligence*, pages 166–175. Springer, 2012.

6 Z. Gu, E. Rothberg, and R. Bixby. *Gurobi 6.0.3*. Gurobi Optimization, Inc., Houston, TX, May 2015.

7 L.G. Kroon, D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maroti, A. Shrijver, A. Steenbeek, and R. Ybema. The new Dutch timetable: The OR Revolution. *Interfaces*, 39:6–17, 2009.

8 M. Kümmling, P. Großmann, K. Nachtigall, J. Opitz, and R. Weiß. A state-of-the-art realization of cyclic railway timetable computation. *Public Transport*, 7(3):281–293, 2015.

9 M. Kümmling, J. Opitz, and P. Großmann. Combining cyclic timetable optimization and traffic assignment. In *20th Conference of the International Federation of Operational Research Societies (IFORS)*, Barcelona, Spain, presentation, 2014.

10 C. Liebchen. Finding short integral cycle bases for cyclic timetabling. In *Proceedings of European Symposium on Algorithms (ESA) 2003*, pages 715–726, 2003.

11 C. Liebchen. *Periodic Timetable Optimization in Public Transport*. dissertation.de – Verlag im Internet, Berlin, 2006.

12 C. Liebchen. The first optimized railway timetable in practice. *Transportation Science*, 42(4):420–435, 2008.

13 C. Liebchen and R. Möhring. The modeling power of the periodic event scheduling problem: railway timetables — and beyond. In *Algorithmic Methods for Railway Optimization*, number 4359 in Lecture Notes on Computer Science, pages 3–40. Springer, 2007.

14 C. Liebchen and R. Rizzi. A greedy approach to compute a minimum cycle basis of a directed graph. *Information Processing Letters*, 94(3):107–112, 2005.

15 R. Martins, V. Manquinho, and I. Lynce. Open-WBO: A Modular MaxSAT Solver. In Carsten Sinz and Uwe Egly, editors, *Theory and Applications of Satisfiability Testing – SAT 2014*, volume 8561 of *Lecture Notes in Computer Science*, pages 438–445. Springer International Publishing, 2014.

16 K. Nachtigall. *Periodic Network Optimization and Fixed Interval Timetables*. PhD thesis, University of Hildesheim, 1998.

17 K. Nachtigall and J. Opitz. Solving periodic timetable optimisation problems by modulo simplex calculations. In *Proc. ATMOS*, 2008.

18 K. Nachtigall and S. Voget. A genetic approach to periodic railway synchronization. *Computers Ops. Res.*, 23(5):453–463, 1996.

19 M. A. Odijk. A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research*, 30B:455–464, 1996.

20 L. Peeters. *Cyclic Railway Timetabling Optimization.* PhD thesis, ERIM, Rotterdam School of Management, 2003.

21 L. Peeters and L. Kroon. A cycle based optimization model for the cyclic railway timetabling problem. In S. Voß and J. Daduna, editors, *Computer-Aided Transit Scheduling*, volume 505 of *Lecture Notes in Economics and Mathematical systems*, pages 275–296. Springer, 2001.

22 L. Peeters and L. Kroon. A variable trip time model for cyclic railway timetabling. *Transportation Science*, 37(2):198–212, 2003.

23 M. Schmidt. *Integrating Routing Decisions in Public Transportation Problems*, volume 89 of *Optimization and Its Applications.* Springer, 2014.

24 M. Schmidt and A. Schöbel. Timetabling with passenger routing. *OR Spectrum*, 37:75–97, 2015.

25 A. Schöbel. Line planning in public transportation: models and methods. *OR Spectrum*, 34(3):491–510, 2012.

26 P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematic*, 2:550–581, 1989.

A Objective function of the SAT formulation

▶ **Theorem 2.** *The Timetabling Problem with Passenger Routing can be formulated as a partial weighted MaxSAT problem.*

Proof. We already showed that the feasibility of the Timetabling Problem with Passenger Routing can be modeled by SAT constraints. Thus it remains only to show that the objective can be expressed as set of clauses with positive weight.

At first we need auxiliary variables $\tau_a^k \in \{0,1\}$ for all activities $a = (i,j) \in \mathcal{A}^0, k \in \{0, \dots, U_a + 1\}$ which determine if $\pi_j - \pi_i + z_a T \geq k$ holds. Here $z_a \in \mathbb{Z}$ is the corresponding modulo parameter.

We need to make sure that τ_a^k is consistent for all $k \in \{1, \dots, U_a\}$, i.e., that it really models a $\geq$-constraint. We encode this similar to the encoding *enc* of the variables π_i:

$$enc' \colon a \mapsto (\tau_a^0 \wedge \neg\tau_a^{U_a+1} \bigwedge_{k \in \{1,\dots,U_a+1\}} (\neg\tau_a^k \vee \tau_a^{k-1})).$$

It remains to ensure that τ_a^k is true if $\pi_j - \pi_i + z_a \cdot T \geq k$. For $k \leq L_a$ we already know this due to the timetabling constraints. Therefore, we consider the following:

$$\begin{aligned} &(\pi_j - \pi_i + z_a \cdot T \geq k) \Rightarrow \tau_a^k \\ \iff\ &\neg(\pi_j - \pi_i + z_a \cdot T \geq k) \vee \tau_a^k \\ \iff\ &\underbrace{\pi_j - \pi_i + z_a \cdot T \in [L_a, k-1]}_{F_2} \vee \tau_a^k \end{aligned}$$

for all $a \in \mathcal{A}^0, k \in \{L_a + 1, \dots, U_a\}$.

As F_2 can be encoded in the same way as any other timetabling constraint, we again get conjunction of clauses here.

Now we can express the length of an activity as the sum of τ_a^k variables.

$$\pi_j - \pi_i + z_a \cdot T = \sum_{k=1}^{U_a} \tau_a^k$$

Now we can formulate the objective function using only binary variables:

$$
\begin{aligned}
&\max \sum_{u,v\in V}\sum_{t=1}^{T_{u,v}} C_{u,v}^t \cdot (D_{u,v}\cdot x_{u,v}^t - \sum_{a\in\mathcal{A}^0} z_a^{u,v,t}\cdot(\sum_{k=1}^{U_a}\tau_a^k)\\
&\quad - \sum_{a=((u,v,t,t',\text{source}),\bullet)\in\mathcal{A}^0_{\text{time}}} P_{u,v}^{t,t'}\cdot z_a^{u,v,t})\\
\iff &\max \sum_{u,v\in V}\sum_{t=1}^{T_{u,v}} C_{u,v}^t \cdot (D_{u,v}\cdot x_{u,v}^t + \sum_{a\in\mathcal{A}^0}(-z_a^{u,v,t}\cdot U_a + z_a^{u,v,t}\cdot\sum_{k=1}^{U_a}\neg\tau_a^k)\\
&\quad \underbrace{- \sum_{a=((u,v,t,t',\text{source}),\bullet)\in\mathcal{A}^0_{\text{time}}} P_{u,v}^{t,t'}}_{\text{fixed}}\\
&\quad + \sum_{a=((u,v,t,t',\text{source}),\bullet)\in\mathcal{A}^0_{\text{time}}} P_{u,v}^{t,t'}\cdot\neg z_a^{u,v,t})\\
\iff &\max \sum_{u,v\in V}\sum_{t=1}^{T_{u,v}} C_{u,v}^t\cdot D_{u,v}\cdot x_{u,v}^t\\
&\quad + \sum_{u,v\in V}\sum_{t=1}^{T_{u,v}}\sum_{a\in\mathcal{A}^0} -C_{u,v}^t\cdot U_a\cdot z_a^{u,v,t}\\
&\quad + \sum_{u,v\in V}\sum_{t=1}^{T_{u,v}}\sum_{a\in\mathcal{A}^0}\sum_{k=1}^{U_a} C_{u,v}^t\cdot z_a^{u,v,t}\cdot\neg\tau_a^k\\
&\quad + \sum_{u,v\in V}\sum_{t=1}^{T_{u,v}}\sum_{a=((u,v,t,t',\text{source}),\bullet)\in\mathcal{A}^0_{\text{time}}} C_{u,v}^t\cdot P_{u,v}^{t,t'}\cdot\neg z_a^{u,v,t}\\
\iff &\max \sum_{u,v\in V}\sum_{t=1}^{T_{u,v}} C_{u,v}^t\cdot D_{u,v}\cdot x_{u,v}^t\\
&\quad + \underbrace{\sum_{u,v\in V}\sum_{t=1}^{T_{u,v}}\sum_{a\in\mathcal{A}^0} -C_{u,v}^t\cdot U_a}_{\text{fixed}}\\
&\quad + \sum_{u,v\in V}\sum_{t=1}^{T_{u,v}}\sum_{a\in\mathcal{A}^0} C_{u,v}^t\cdot U_a\cdot\neg z_a^{u,v,t}\\
&\quad + \sum_{u,v\in V}\sum_{t=1}^{T_{u,v}}\sum_{a\in\mathcal{A}^0}\sum_{k=1}^{U_a} C_{u,v}^t\cdot z_a^{u,v,t}\cdot\neg\tau_a^k\\
&\quad + \sum_{u,v\in V}\sum_{t=1}^{T_{u,v}}\sum_{a=((u,v,t,t',\text{source}),\bullet)\in\mathcal{A}^0_{\text{time}}} C_{u,v}^t\cdot P_{u,v}^{t,t'}\cdot\neg z_a^{u,v,t}
\end{aligned}
$$

As we want to maximize, we can substitute $z_a^{u,v,t} \cdot \neg\tau_a^k$ by a variable $y_a^{u,v,t,k}$ which is set to 0 if either $\neg\tau_a^k = 0$ or $z_a^{u,v,t} = 0$.

$$\begin{aligned}
\iff \max \sum_{u,v \in V} \sum_{t=1}^{T_{u,v}} C_{u,v}^t \cdot D_{u,v} \cdot x_{u,v}^t \\
+ \sum_{u,v \in V} \sum_{t=1}^{T_{u,v}} \sum_{a \in \mathcal{A}^0} C_{u,v}^t \cdot U_a \cdot \neg z_a^{u,v,t} \\
+ \sum_{u,v \in V} \sum_{t=1}^{T_{u,v}} \sum_{a \in \mathcal{A}^0} \sum_{k=1}^{U_a} C_{u,v}^t \cdot y_a^{u,v,t,k} \\
+ \sum_{u,v \in V} \sum_{t=1}^{T_{u,v}} \sum_{a=((u,v,t,t',\text{source}),\bullet) \in \mathcal{A}^0_{\text{time}}} C_{u,v}^t \cdot P_{u,v}^{t,t'} \cdot \neg z_a^{u,v,t}
\end{aligned}$$

From the substitution we get the following clauses:

$$\begin{aligned}
\neg\neg\tau_a^k \Rightarrow \neg y_a^{u,v,t,k} &\iff \neg\tau_a^k \vee \neg y_a^{u,v,t,k} \\
\neg z_a^{u,v,t} \Rightarrow \neg y_a^{u,v,t,k} &\iff z_a^{u,v,t} \vee \neg y_a^{u,v,t,k}
\end{aligned}$$

We see that the objective is to maximize a sum of weighted booleans. This can modeled in a partial weighted MaxSAT problem, where all the clauses appearing in the objective get their respective weight from the objective function and all other clauses which are needed to model the constraints get weight infinity, i.e., they have to be fulfilled anyway.

◀