Solving Periodic Event Scheduling Problems efficiently by reducing to SAT

Julian Ritter
Matrikelnr. 4365168


Freie Universität Berlin
Institut für Mathematik

Seminar »Periodic Timetabling and Passenger Routing«
Prof. Dr. Ralf Borndörfer, Dr. Marika Karbstein, Heide Hoppmann
This seminar talk is based on the papers


1 Introduction

The authors of [1] state that railway planning in Germany often is manual labour. Unsatisfied with the resulting sub-optimal timetables, their motivation is to work on versatile models and efficient solving techniques to help improving the German railway network. They present »A state-of-the-art realization of cyclic railway timetable computation«, as they name their paper. More explicitly, they indicate that the timetables they compute are used in timetabling studies to evaluate both infrastructural and operational concepts (»different routes, stopping patterns, vehicles and so on«([1], p.282)). With respect to this seminar’s title, »Periodic Timetabling and Passenger Routing«, their work does not deal with passenger routing, but focuses entirely on periodic timetabling. Moreover, they restrict their work to feasibility studies and propose the idea of timetable optimisation with regard to passenger demand only as a possible future extension of their work.

The main paper I worked with, [1] published in 2015, attempts to cover the entire process of timetable creation, describing mathematical details only here and there, often referring to a previous work of some of the authors, [2], published in 2012, which handles their main technique more thoroughly. I will roughly follow the step-by-step procedure presented in [1]. I will start by introducing the model used to describe a timetabling problem, after which I will present the authors’ proposed solving technique with mathematical rigorosity. The rest of the paper contains an overview of related literature rather than own findings, so I will not present it here.

2 The Model: A Periodic Event Network

This graphic (taken from [1], p.283) gives a visualisation of a small Periodic Event Network (PEN). It encodes information on trains to schedule as well as infrastructural and operational demands. Each train \(L \in \mathcal{L}\) serves a specified sequence of stations \(S \in S\). In a periodic timetable with period \(t^* \in \mathbb{N}\), each event happens periodically at the times \(t \equiv T_v \pmod{t^*}\). Each train can be implemented as a sequence of departure and arrival vertices at the respective stations. The pairs of succeeding vertices in the sequence are encoded as run- and dwell-arcs, respectively. Furthermore, interactions between different trains are taken into account as well: Headway arcs are used to encode minimum safe headways between different trains on identical or independent infrastructure. They can also be used to evenly distribute intervals between trains for a more robust timetable. Transfer times include the times required for passengers or staff to change from one train to another as well as for vehicles to end one service and start another. All those arcs are equipped with lower and upper bounds for the amount of time between the corresponding events.

The question »Is there a timetable that fulfills all these conditions?« is called the Periodic Event Scheduling Problem (PESP). It is »established as one of the most suitable problem formulations for periodic timetabling« ([1], p. 282) and the underlying Periodic Event Network (PEN) »permits flexible representation of almost all periodic timetable’s constraints« ([1], p. 282). Their formal definitions are the following:
Definition 2.1 (A PEN for Periodic Timetabling)

\[ \mathcal{L} = \{L_1, \ldots \} \] set of periodic trains
\[ \mathcal{S} = \{S_1, \ldots \} \] set of stations in the railway network
\[ \mathcal{V} = \mathcal{V}_{arr} \cup \mathcal{V}_{dep} \] vertex set, where the vertices are of 2 types
\[ \mathcal{V}_{arr} \ni (L, arr, S) \] vertices for the arrival of train L at station S
\[ \mathcal{V}_{dep} \ni (L, dep, S) \] vertices for the departure of train L from station S
\[ \mathcal{A} = \mathcal{A}_{run} \cup \mathcal{A}_{dwell} \cup \mathcal{A}_{headway} \cup \mathcal{A}_{transfer} \] arc set, where the arcs are of 4 types
\[ \mathcal{A}_{run} \ni ((L, dep, S_1), (L, arr, S_2)) \] arcs for running time of L from S_1 to S_2
\[ \mathcal{A}_{dwell} \ni ((L, arr, S), (L, dep, S)) \] arcs for dwell time of L at station S
\[ \mathcal{A}_{headway} \ni ((L_1, dep, S_1), (L_2, dep, S_2)) \] arcs for headway between L_1 and L_2 on a segment
\[ \mathcal{A}_{transfer} \ni ((L_1, arr, S), (L_2, dep, S)) \] arcs for transfer time from L_1 to L_2 at S

\[ t^* \in \mathbb{N} \] period
\[ t_{\min}, t_{\max} : \mathcal{A} \rightarrow \mathbb{Z}_{\geq 0} \] lower and upper bounds for each arc

Definition 2.2 (Timetable, Validity)
A timetable \( T \) for a given PEN \( \mathcal{P} = (\mathcal{V}, \mathcal{A}, t^*, t_{\min}, t_{\max}) \) is a function \( T : \mathcal{V} \rightarrow \mathbb{Z} \cap [0, t^* - 1] \)
A timetable \( T \) is valid \( \iff \forall (x, y) \in \mathcal{A} : T(y) - T(x) \in [t_{\min}(a), t_{\max}(a)] \mod t^* \)

Definition 2.3 (Periodic Event Scheduling Problem (PESP))

Input: \( \mathcal{P} = (\mathcal{V}, \mathcal{A}, t^*, t_{\min}, t_{\max}) \) Periodic Event Network
Output: YES / NO Answer to »Does a valid timetable \( T \) for \( \mathcal{P} \) exist?«

3 The Technique: Conversion into a Satisfiability Problem

The authors state that the PESP is NP-complete. Another NP-complete problem is the Satisfiability Problem (SAT). A main difference of the two: Very efficient solvers for the SAT problem are known. [2] finds that despite the effort to encode a PESP instance as a SAT instance, this detour outperforms any previously known approach to solve a PESP in terms of running time. We will define the SAT and describe how the translation of a PESP instance into a SAT instance works.

Definition 3.1 (SAT terminology)
Let \( X \) be a finite set of Boolean variables. An interpretation of \( X \) is a function \( I : X \rightarrow \{\text{true}, \text{false}\} \). A clause \( C \) over \( X \) is a set of variables and negations of variables in \( X \). It is satisfied by an interpretation \( (I \models C) \) iff at least one of its members is assigned »true«. A family of clauses \( F \) over \( X \) is satisfiable iff there is some interpretation simultaneously satisfying all of its clauses. Such an interpretation \( I \models F = \bigwedge_{C \in F} C \) is called a model for \( F \).

Definition 3.2 (Satisfiability Problem (SAT))

Input: \( F = \bigwedge_{C \in F} (\bigvee_{q \in C} q) \) Family of clauses
Output: YES / NO Answer to »Does a model \( J \) for the family of clauses \( F \) exist?«

We will now see how to encode a PEN as an instance of the Satisfiability Problem that is equivalent to the PESP instance for the PEN. Encoding the vertices of a PEN into Boolean variables is done by order encoding, which is not an invention of the authors but an established technique to translate problems into propositional logic. For each vertex \( v \in \mathcal{V} \) of the PEN, a number of \( t^* \) Boolean variables \( q_{v,i}, i \in \{0, \ldots, t^* - 1\} \) are introduced, where the logical value of \( q_{v,i} \) represents the answer to »Is \( T(v) \leq i \)?«, so that its negation means that \( T(v) > i \).

Definition 3.3 (encV)
For \( v \in \mathcal{V} \) vertex of a PEN \( \mathcal{P} = (\mathcal{V}, \mathcal{A}, t^*, t_{\min}, t_{\max}) \), we define \( \text{encV}(\mathcal{P}, v) := q_{v,t^*-1} \bigwedge_{i \in [1,t^*-1]} (\neg q_{v,i-1} \lor q_{v,i}) \)
Lemma 1
Let $\mathcal{P} = (\mathcal{V}, A, t^*, t_{\min}, t_{\max})$ be a PEN, $v \in \mathcal{V}$ and $J$ an interpretation of $\{q_{v,i} : v \in \mathcal{V}, i \in [0, t^* - 1]\}$. Then:

$$J \models \text{encV}(\mathcal{P}, v) \iff \exists! k \in [0, t^* - 1]: (i) \forall i \in [0, k - 1]: J \not\models q_{v,i} \text{ and } (ii) \forall j \in [k, t^* - 1]: J \models q_{v,j}$$

Proof. 1. "$\Rightarrow$": Let $J$ be an interpretation of $\{q_{v,i} : v \in \mathcal{V}, i \in [0, t^* - 1]\}$ with $J \models \text{encV}(\mathcal{P}, v)$. Then $J \models q_{v,t^* - 1}$, so $\{i \in [0, t^* - 1] : J \models q_{v,i}\} \neq \emptyset$. Now we can safely define $k := \min\{i \in [0, t^* - 1] : J \models q_{v,i}\}$. As $J \models \bigwedge_{i \in [1, t^* - 1]}(\neg q_{v,i - 1} \lor q_{v,i})$, it follows that $J \models q_{v,i}$ for $i \geq k$. As $k$ was chosen to be minimal, $J \not\models q_{v,i}$ for $i < k$ and $k$ is unique with these two properties.

2. "$\Leftarrow$": Let $J$ be an interpretation of $\{q_{v,i} : v \in \mathcal{V}, i \in [0, t^* - 1]\}$ and $k \in [0, t^* - 1]$ an integer fulfilling (i) & (ii). (i) yields $J \models \bigwedge_{i \in [1,k]}(\neg q_{v,i - 1})$, so $J \models \bigwedge_{i \in [1,k]}(\neg q_{v,i - 1} \lor q_{v,i})$. (ii) yields $J \models \bigwedge_{i \in [k,t^* - 1]}(q_{v,i})$, so $J \models q_{v,t^* - 1} \bigwedge_{i \in [1,t^* - 1]}(\neg q_{v,i - 1} \lor q_{v,i})$. Together, we get $J \models q_{v,t^* - 1} \bigwedge_{i \in [1,t^* - 1]}(\neg q_{v,i - 1} \lor q_{v,i}) \Rightarrow J \models \text{encV}(\mathcal{P}, v)$

\(\square\)

Definition 3.4 ($J_T$)
Let $T$ be a timetable for $\mathcal{P} = (\mathcal{V}, A, t^*, t_{\min}, t_{\max})$. We define an interpretation $J_T$ of $\{q_{v,i} : v \in \mathcal{V}, i \in [0, t^* - 1]\}$:

$$\forall v \in \mathcal{V} \forall i \in [0, t^* - 1]: J_T(q_{v,i}) := \begin{cases} false, & i < T(v) \\ true, & i \geq T(v) \end{cases}$$

Definition 3.5 ($T_J$)
Let $J$ be an interpretation of $\{q_{v,i} : v \in \mathcal{V}, i \in [0, t^* - 1]\}$ with $J \models \bigwedge_{v \in \mathcal{V}} \text{encV}(\mathcal{P}, v)$. We define a function $T_J$:

$$T_J : \mathcal{V} \to \mathbb{Z}_{\geq 0}, T_J(v) = \min\{i \in [0, t^* - 1] : J \models q_{v,i}\}$$

We can now make some Observations:

1. As $T$ is a timetable for $\mathcal{P}$, $J_T$ is well-defined. Furthermore, it meets the requirements of Lemma 1 with $k = T(v)$, so $\forall v \in \mathcal{V} : J_T \models \text{encV}(\mathcal{P}, v)$.

2. As $\forall v \in \mathcal{V} : J \models \text{encV}(\mathcal{P}, v)$, by Lemma 1, the set $\{i \in [0, t^* - 1] : J \models q_{v,i}\}$ is non-empty and $T_J$ is well-defined, complying with the definition of a timetable for any PEN with vertex set $\mathcal{V}$ and period $t^*$.

3. Note that for a timetable $T$, an interpretation $J \models \text{encV}(\mathcal{P}, v)$ and $v \in \mathcal{V}$, both $T_J = T$ and $J_{T_J} = J$:

- $T_{J_T}(v) = \min\{i \in [0, t^* - 1] : J_T \models q_{v,i}\} = \min\{i \in [0, t^* - 1] : i \geq T(v)\} = T(v)$
- $\forall i \in [0, t^* - 1]: J_{T_J} \models q_{v,i} \iff i \geq T_J(v) = \min\{j \in [0, t^* - 1] : J \models q_{v,j}\} \iff J \models q_{v,i}$

We have succeeded in translating timetables for the PEN into interpretations of our SAT clauses and vice versa. In order to complete our translation process, we now need to implement the arc constraints of the PEN into the SAT instance, enabling us to handle the required validity of a timetable in the SAT instance. We do so by considering each single arc $a = (x, y)$ of the PEN, and generating clauses to exclude all invalid combinations of values $T(x), T(y)$ with $(T(y) - T(x)) \not\in [t_{\min}(a), t_{\max}(a)]$ (mod $t^*$). Using a clause for each invalid combination would result in a number of clauses quadratic in $t^*$. In [2], the authors state that a smaller number of clauses seems to decrease the running time of SAT solvers, so it is advantageous to exclude multiple invalid combinations per clause. For a single arc $a$, we can visualise valid and invalid combinations on a grid.
Definition 3.6 (encR)
For \((x, y) = a \in A\) arc of a PEN and \(R = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \subseteq [0, t^* - 1]^2\), we define \(\text{encR}(P, R, x, y)\):

\[
\text{encR}(P, R, x, y) := \begin{cases} 
q_x x_{\min} - 1 \lor \neg q_x x_{\max} \lor q_y y_{\min} - 1 \lor \neg q_y y_{\max} & 0 < x_{\min} < y_{\min} \\

\neg q_x x_{\max} \lor q_y y_{\min} - 1 \lor \neg q_y y_{\max} & 0 = x_{\min} < y_{\min} \\

\neg q_x x_{\min} - 1 \lor \neg q_x x_{\max} \lor \neg q_y y_{\max} & 0 = y_{\min} < x_{\min} \\

\neg q_x x_{\max} \lor \neg q_y y_{\max} & 0 = x_{\min} = y_{\min} 
\end{cases}
\]

Lemma 2
Let \((x, y) = a \in A\) be an arc of a PEN \(P = (V, A, t^*, t_{\min}, t_{\max})\), \(J\) an interpretation of \(\{q_{v,i} : v \in V, i \in [0, t^* - 1]\}\) with \(J \models \bigwedge_{v \in V} \text{encV}(P, v)\) and \(R = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \subseteq [0, t^* - 1]^2\). Then:

\[J \models \text{encR}(P, R, x, y) \iff (T_J(x), T_J(y)) \not\in R.\]

Proof. The equivalence of the negations follows quite straightforwardly from the definitions:

\[(T_J(x), T_J(y)) \in R \iff x_{\min} - 1 < T_J(x) \leq x_{\max} \text{ and } y_{\min} - 1 < T_J(y) \leq y_{\max} \quad \text{(Def. R)}\]

\[J \models \neg q_x x_{\min} - 1 \land q_x x_{\max} \land \neg q_y y_{\min} - 1 \land q_y y_{\max} \quad \text{(Def. } \text{encR})\]

\[J \not\models \text{encR}(P, R, x, y) \iff (T_J(x), T_J(y)) \not\in R. \quad \text{(Obs. 3: } J_T = J)\]

(*) If \(x_{\min}\) or \(y_{\min}\) are zero, the corresponding condition is omitted, the proof remains valid.

We now know how we can encode a forbidden rectangle into a clause. If we can find a way to cover the forbidden pairs by rectangles, we can add a clause for each such rectangle for each arc and any interpretation that fulfills them all will be a valid timetable. The authors propose a covering of the infeasible region with rectangles, but their detailed approach in [2] seems to lack the handling of some cases and is consequently not mentioned in the subsequent paper [1]. Their rectangle covering can be replaced with one based on stripes. Firstly, we split the grid into those columns with one infeasible region and those with two infeasible regions, before we define the infeasible regions on each column:

Definition 3.7 (\(\mathcal{R}(P, a)\))
Let \((x, y) = a \in A\) be an arc of the PEN \(P = (V, A, t^*, t_{\min}, t_{\max})\). We define:

\[
S_1(P, a) := \{i \in [0, t^* - 1] : i + t_{\min}(a) \equiv i + t_{\max}(a) \pmod{t^*}\} \\
S_2(P, a) := \{i \in [0, t^* - 1] : i + t_{\min}(a) \equiv i + t_{\max}(a) \pmod{t^*}\} \\
\mathcal{R}(P, a) := \{\{i\} \times [i + t_{\max}(a) \pmod{t^*} + 1, i + t_{\min}(a) \pmod{t^*} - 1] : i \in S_1(a)\} \\
\cup \{\{i\} \times [i + t_{\max}(a) \pmod{t^*} + 1, i + t_{\min}(a) \mod{t^*} - 1] : i \in S_2(a)\} \\
\cup \{\{i\} \times [0, i + t_{\min}(a) \pmod{t^*} - 1] : i \in S_2(a)\}
\]

Definition 3.8 (encA)
For \((x, y) = a \in A\) arc of a PEN \(P = (V, A, t^*, t_{\min}, t_{\max})\), we define \(\text{encA}(P, a) := \bigwedge_{R \in \mathcal{R}(P, a)} \text{encR}(P, R, x, y)\)

The family of clauses \(\text{encA}\) contains a clause for each of the stripes defined above. It is constructed so that any interpretation \(J\) of \(\{q_{v,i} : v \in V, i \in [0, t^* - 1]\}\) satisfying this family does not contain any combination in any of the infeasible stripes, so that the constraint \(T_J(y) - T_J(x) \in [t_{\min}(a), t_{\max}(a)] \pmod{t^*}\) of the PEN \(P\) is satisfied by \(J\) and its induced timetable. We will prove this in our third and final lemma.

Lemma 3
Let \((x, y) = a \in A\) be an arc of a PEN \(P = (V, A, t^*, t_{\min}, t_{\max})\), \(J \models \bigwedge_{v \in V} \text{encV}(P, v)\). Then:

\[J \models \text{encA}(P, a) \iff (T_J(y) - T_J(x)) \in [t_{\min}(a), t_{\max}(a)] \pmod{t^*}\]
Proof. The definition of $\text{encA}(P, a)$ and Lemma 2 ensure the following:

$$J \models \text{encA}(P, a) \iff \forall R \in \mathcal{R}((x, y)) : J \models \text{encR}(P, R, x, y) \iff \forall R \in \mathcal{R}((x, y)) : (T_J(x), T_J(y)) \notin R.$$ 

We fix $i = T_J(x)$ and distinguish two cases:

1. $i \in S_1(a) [i + t_{\min}(a) (\mod t^*) > i + t_{\max}(a) (\mod t^*)]$:
   $$\forall R \in \mathcal{R}((x, y)) : (T_J(x), T_J(y)) \notin R$$
   $$\iff T_J(y) \notin [i + t_{\max}(a) (\mod t^*) + 1, i + t_{\min}(a) (\mod t^*) - 1]$$
   $$\iff T_J(y) \in [0, i + t_{\max}(a) (\mod t^*)] \cup [i + t_{\min}(a), t^* - 1] (\mod t^*)$$
   $$\iff T_J(y) \in [i + t_{\min}(a), i + t_{\max}(a)] (\mod t^*)$$
   $$\iff T_J(y) - i = T_J(y) - T_J(x) \in [t_{\min}(a), t_{\max}(a)] (\mod t^*)$$

2. $i \in S_2(a) [i + t_{\min}(a) (\mod t^*) \leq i + t_{\max}(a) (\mod t^*)]$:
   $$\forall R \in \mathcal{R}((x, y)) : (T_J(x), T_J(y)) \notin R$$
   $$\iff T_J(y) \notin [i + t_{\max}(a) (\mod t^*) + 1, t^* - 1] \cup [0, i + t_{\min}(a) (\mod t^*) - 1]$$
   $$\iff T_J(y) \in [i + t_{\min}(a), i + t_{\max}(a)] (\mod t^*)$$
   $$\iff T_J(y) - i = T_J(y) - T_J(x) \in [t_{\min}(a), t_{\max}(a)] (\mod t^*)$$

$\square$

**Definition 3.9 (enc(P))**

For a PEN $P = (V, A, t^*, t_{\min}, t_{\max})$, we define $\text{enc}(P) := \bigwedge_{v \in V} \text{encV}(P, v) \bigwedge_{a \in A} \text{encA}(P, a)$.

We have now encoded all the information of a PEN $P$ as a family of clauses $\text{enc}(P)$. Let us now take a look at the SAT instance associated with this family of clauses. We are prepared to prove that the PESP instance asking for a valid timetable for a PEN $P$ and its SAT translation, asking for a model of $\text{enc}(P)$, are equivalent. Thus, we can use SAT solvers to tackle our initial PESP, outperforming the previously used PESP solvers significantly.

**Theorem 1**

Let $P = (V, A, t^*, t_{\min}, t_{\max})$ be a PEN and $\text{enc}(P)$ be the encoded family of clauses of $P$. Then:

$$\exists J : J \models \text{enc}(P) \iff \exists T : T \text{ is a valid timetable for } P.$$ 

**Proof.**

1. "$\Rightarrow$":
   
   $J \models (\bigwedge_{v \in V} \text{encV}(P, v) \land \bigwedge_{a \in A} \text{encA}(P, a))$ (Def. $\text{enc}(P)$)
   
   $\Rightarrow \forall v \in V : J \models \text{encV}(P, v)$
   and $\forall a \in A : J \models \text{encA}(P, a)$
   
   $\Rightarrow \forall v \in V : J \models \text{encV}(P, v)$
   and $\forall (x, y) = a \in A : (T_J(y) - T_J(x)) \in [t_{\min}(a), t_{\max}(a)] (\mod t^*)$ (Lemma 3)
   
   $\Rightarrow T_J$ is a timetable for $P$
   and $T_J$ is valid for $P$ (Observation 2)
   
   $\Rightarrow T_J$ is a valid timetable for $P$. (Def. Validity)

2. "$\Leftarrow$":
   
   $T = T_{J_P}$ is a valid timetable for $P$. (Observation 3)
   
   $\Rightarrow T$ is a timetable for $P$
   and $\forall (x, y) = a \in A : (T_{J_P}(y) - T_{J_P}(x)) \in [t_{\min}(a), t_{\max}(a)] (\mod t^*)$ (Def. Validity)
   
   $\Rightarrow \forall v \in V : J_T \models \text{encV}(P, v)$
   and $\forall a \in A : J_T \models \text{encA}(P, a)$ (Observation 1)
   
   $\Rightarrow J_T = (\bigwedge_{v \in V} \text{encV}(P, v) \land \bigwedge_{a \in A} \text{encA}(P, a))$ (Lemma 3)
   
   $\Rightarrow J_T \models \text{enc}(P)$ (Def. $\text{enc}(P)$)

$\square$