Omega 116 (2023) 102798

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

A MILP model for quasi-periodic strategic train timetabling[☆,☆☆]

Giorgio Sartor[a,*], Carlo Mannino[a,b], Thomas Nygreen[c], Lukas Bach[d]

[a] SINTEF, Oslo, Norway
[b] University of Oslo, Norway
[c] Jernbanedirektoratet, Oslo, Norway
[d] Visma, Oslo, Norway

ARTICLE INFO

Article history:
Received 9 December 2021
Accepted 3 November 2022
Available online 11 November 2022

Keywords:
Mixed-integer programming
Train scheduling
Quasi-periodic timetabling

ABSTRACT

In railways, the long-term strategic planning is the process of evaluating improvements to the railway network (e.g., upgrading a single track line to a double track line) and changes to the composition/frequency of train services (e.g., adding 1 train per hour along a certain route). The effects of different combinations of infrastructure upgrades and updated train services (also called scenarios), are usually evaluated by creating new feasible timetables followed by extensive simulation. Strategic Train Timetabling (STT) is indeed the task of producing new tentative timetables for these what-if scenarios. Unlike the more classic train timetabling, STT can often overlook (or at least give less importance to) some complementary aspects, such as crew and rolling stock scheduling. On the other hand, the different scenarios are likely to lead to very different timetables, hindering the common and effective practice of using existing timetables to warm start the solution process. We introduce the concept of quasi-periodic timetables, that are timetables where certain subsets of trains need to start at almost (rather than precisely) the same minute of every period. The additional flexibility offered by quasi-periodic timetables turned out to be crucial in real-life scenarios characterized by elevated train traffic. We describe a MILP based approach for strategic quasi-periodic train timetabling and we test it on 4 different realistic what-if scenarios for an important line in Norway. The timetables produced by our algorithm were ultimately used by the Norwegian Railway Directorate to select 3 out of the 4 scenarios for phasing the progressive expansion of the JÕren line.

1. Introduction

As railway networks become busier and as the cost of improving/extending the infrastructures becomes higher, it is crucial to be able to exploit them to their fullest potential. To achieve this, it would be important to develop both an automatic rail-time dispatching system (which reduces load on train dispatchers, allowing them to handle more trains at the same time) and a more sophisticated optimization-based timetabling tool (which increases the utilization of the network). Sometimes, investments in the infrastructure are unavoidable if a relatively ambitious train service is desired. In this case, it is crucial to carefully investigate all available options and choose the most appropriate one. The most common way of evaluating such options is indeed to first create a tentative timetable that satisfies the desired train service requirements with the new infrastructure, and then use off-the-shelf simulation tools to establish a score that is based not only on the money expenditure (e.g., of building the new infrastructure or purchasing more trains) but also on several social aspects (passenger satisfaction, CO_2 emissions, etc.). This is commonly called strategic planning, and it is part of a three-level process that is used by almost all national railway authorities to create feasible timetables for the daily operations. Here is a brief summary of the different planning levels:

- **Operational** (daily, maximum precision): it consists of computing the exact times trains are actually planned to leave and arrive at stations with second-precision, taking into account all microscopic train operations that ultimately guarantee the safety of passengers (running times between signals, safety margins, exact routing in stations, speed profiles, etc.). An operational timetable is always a refinement of the current tactical timetable.

☆ This manuscript was processed by Associate Editor Kis. Edited by Prof. Benjamin Lev.

☆☆ This work was partially funded by the Norwegian Research Council (projects #267554, #237718) and by the Norwegian Railway Directorate (project StraTi)

* Corresponding author.
E-mail address: giorgio.sartor@sintef.no (G. Sartor).

https://doi.org/10.1016/j.omega.2022.102798
0305-0483/

- **Tactical** (once a year, good precision): based on the service requests of the train operators and on the planned maintenance/upgrades of the infrastructure (both dependent on previous strategic planning), the planners produce once a year a timetable that is as close as possible to being operational. Given the uncertainty regarding some of the inputs (e.g., weather conditions, weights of freight trains), the tactical timetable does not need to be as precise as the operational one and may require some small adjustments on a daily basis. The **public** timetable is directly derived from the tactical one and presented to the passengers once a year in a simplified fashion.
- **Strategic** (varying intervals and precision, but focused on the long-term): this consists of creating a tentative timetable used in the strategic planning and development of railway networks. While the operational timetable is an actual and detailed plan of operations and the public timetable is a promise to the customers, the strategic timetable is a tool for determining necessary investments in infrastructure and rolling stock, and for assessing the viability of service improvements. Strategic timetables are subject to more uncertainty and require less precision than the tactical ones. Consequently, they may initially overlook some of the microscopic aspects of the train operations. However, it is crucial that each strategic timetable can be refined, without significant changes, to a feasible operational timetable. Unlike tactical and operational, strategic timetables are often built from scratch and may be completely different from previous or existing ones, making the entire process quite different and the solution space much larger.

Jernbanedirektoratet (Norwegian Railway Directorate), the government agency responsible for managing the Norwegian railway sector on behalf of the Norwegian state, performs strategic planning with time horizons spanning from 4 to 15 years. In this paper, we study the task of creating strategic timetables based on the alternative train service requirements and infrastructure upgrades currently under evaluation by Jernbanedirektoratet. We focus on the JŌren line, a busy stretch of railway line in southern Norway that runs between Stavanger and Egersund accomodating around 150 trains a day (see Fig. 1). The section from Stavanger to Sandnes is double track, while the rest is single track. The current train service consists of 4 trains per hour from Stavanger to Sandnes, one every 15 minutes; of these, half continue onwards to NŌrbφ with a 30-minute headway; finally, only one train per hour continues all the way to Egersund. These are what we call *local trains*. Fig. 2 schematically summarizes how the (periodic) local trains' services should be distributed within every hour. The minutes on the dial are only for reference; in fact, the pattern can be "rotated" to achieve feasibility or satisfy other preferences (more on this later). A particular periodic pattern together with a possible infrastructure upgrade will constitute a test scenario.

This railway line also accommodates up to 16 long-distance trains a day and a few freight trains, whose schedules extend substantially after Egersund. We call these *non-local trains*. The planning of local trains happens usually at a different stage than the planning of non-local trains, but we decided to try to plan all the trains at the same time (although for the non-local trains we will consider only a partial schedule). As per request of Jernbanedirektoratet, we also slightly extended the investigated railway line of few stations after Egersund up to Sira.

Finding feasible and optimal solutions for each alternative scenario is a complex and time-consuming task, combining microscopic resource allocation and conflict solving on the one hand with macroscopic strategic decision-making on the other. To overcome the resulting complexity, the number of scenarios must be kept to a minimum, and the planners rely on experience to select the most promising avenues of exploration.

Fig. 1. The Jæren line.

A few infrastructure managers have been working (and are still working) on developing decision support tools for timetabling, both at strategic and tactical level (see, for example, [1]). However, to the best of our knowledge, we are not aware of any commercial tool that is able to *automatically* generate optimal or near-optimal strategic timetables from scratch for given conceptual service requirements and/or specific infrastructures. Currently available commercial tools such as TPS [2] or TRENOplus [3] mainly allow for manual interaction and they have little to no automatic decision support. A bit of an exception is the OptDis module of LUKS [4] that claims to be able to automatically produce a conflict-free timetable that minimizes the deviation from a given roughly planned timetable and maximizes robustness. However, this module is not listed in their website anymore and we are not aware of it being used in commercial applications. While this is definitely not an exhaustive list of railway planning tools, we believe it is a representative one in the sense that shows the lack of an automatic strategic timetabling tool.

Having an automatic decision support tool for generating strategic timetables would allow for the task of the planners to be shifted away from detailed conflict resolution towards higher-level strategic considerations. This change will in turn enable the planners to study scenarios that currently have to be left out due to time or human resource constraints. In this paper, we present an approach based on mixed-integer programming that provides the foundations for a tool useful to the Norwegian railways, and it represents another research step towards practical tools. Moreover, we present the results of a prototype that implements this approach and applies it to the JŌren line on the scenarios that are currently under consideration by Jernbanedirektoratet.

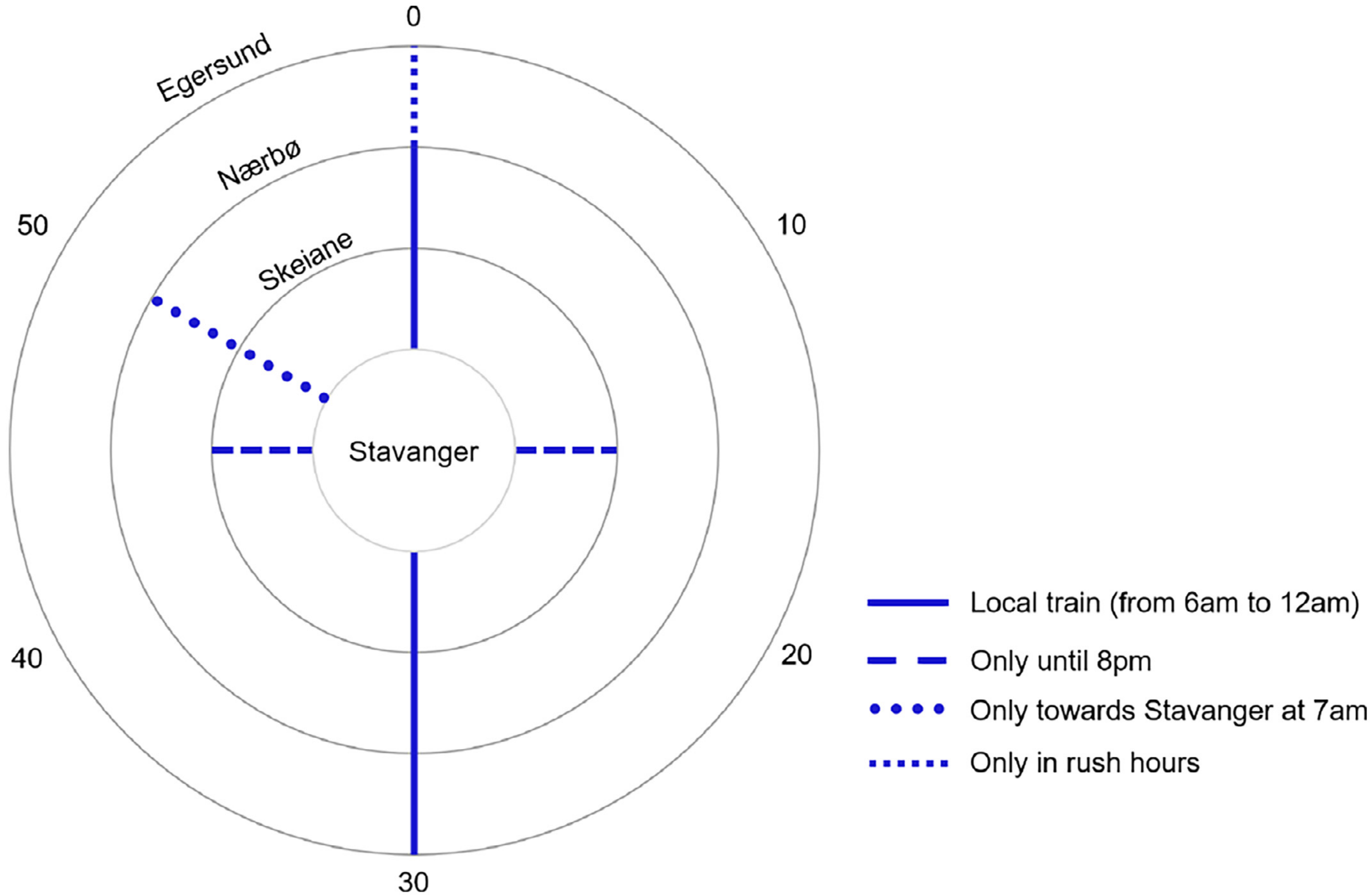

Fig. 2. The actual periodic pattern for 2020.

Finally, it is important to remark that our model does not incorporate variables and constraints to allow for infrastructure decisions, such as, e.g., a new track or platform. The infrastructure is indeed specified by the input, and alternative configurations will correspond to different input scenarios.

1.1. Related work

Despite of the substantial lack of software tools to generate timetables, the literature on train timetabling is quite rich, as also testified by a conspicuous number of surveys and tutorials on the subject, e.g. [5–11]. Basically, all modern approaches are based on Mixed Integer Linear Programming (MILP) models, which are then solved by some standard available solver or by ad-hoc algorithms. For the main stream of works the reference model is the job-shop scheduling problem with no wait and blocking constraints introduced in [12], which also applies to the on-line version of the timetabling problem, namely the train dispatching problem [13]. The basic model is a disjunctive program, and different approaches differ in the way scheduling variables and disjunctive constraints are represented. In particular, the two main representations are the big-M formulation and the time-indexed (TI) formulation (see [5] and, for theoretical insights, see [14]). In the first approach, the scheduling variables are continuous variables and every disjunctive constraint is represented by a binary variable and two constraints containing a large coefficient (the big-M). In the second approach, the time horizon is discretized, where time variables are binary and disjunctive constraints are represented through packing constraints. TI formulations are generally preferred for timetabling problems, whereas big-M formulations are more often adopted for dispatching.

Time-indexed formulations are typically stronger and return better bounds then big-M formulations, but solving each relaxation is much more costly - which in turn significantly slows down the solution search (in [15] a direct comparison of the two formulations on certain on-line train scheduling instances shows a clear advantage for big-M formulations). Moreover, time discretization introduces approximations which may make the returned solutions impossible to implement in practice (see [16,17]). Nevertheless, probably due to their ductility in representing various constraints, many authors adopt TI formulations in their approaches to train timetabling (e.g. [16,18–23], among many others). On the other hand, big-M formulations are also quite exploited in the literature, for instance in [24–31]. These lists of references are far from being exhaustive and we refer the reader to the above mentioned surveys for more comprehensive discussions.

There is another relevant class of MILP models for timetabling, specifically introduced to represent and solve periodic scheduling, namely the Periodic Event Scheduling Problem (PESP) introduced in the seminal paper [32] by Serafini e Ukovich. In a periodic timetable train departures repeat during the time horizon after a period T. PESP formulations have been used to produce periodic timetables in several papers (and some real-life applications), as for instance in [33–37]. One interesting feature of PESP MILP formulation, described by Nachtigall in [38], is that it can be reformulated in a compact and effective way by some variable transformation. This transformation allows to formulate the problem by a compact family of constraints associated with a basis of cycles of the event graph. The reformulation can be strengthened further by several classes of strong inequalities [39]. Although elegant and quite effective, PESP models seem to have some major limits that make them not practical for our problem. In [33], it is claimed that the PESP model can only represent a subset of the constraints which are necessary to model trains dynamic and interaction (for a list of manageable constraints, see [35]). Indeed, we are not aware of any work that includes in the PESP model specific microscopic constraints and routing alternatives. In [33] this is tackled by adding a second stage after the solution of a PESP model, where a non-PESP model is solved to tackle the additional constraints. This second stage problem may be even infeasible and in [33] the authors try to mitigate this issue by introducing some flexibility in the first stage PESP model. Finally, we are not aware of PESP models that can handle the quasi-periodic timetables which are indeed relevant in the case described in this paper.

Finally, it is worth mentioning that a certain number of papers develop approaches to generate robust timetables in order to tackle the uncertainty inherent in any schedule. This is not the focus of this paper, so we refer the reader to the survey paper by Cacchiani and Toth [8]. Instead, we treat uncertainty in the same way it is currently treated by Jernbanedirektoratet, that is by simply considering suitable supplements in running times and buffer times between trains, which are provided as input data.

The model introduced in this work has its roots in [28], where a big-M formulation of the strategic train timetabling problem is paired with a decomposition approach and an effective row-and-column-generation solution method. The scope of [28] was limited to adding a single freight train at a time to an existing timetable, with some restrictions on the periodicity. Our work shares the decomposition approach and the solution method with [28], but it extends its scope and model in several significant ways. In particular, our model considers the planning of all train services, both passenger and freight, at the same time, without an existing timetable to start from. This requires additional constraints, and a more careful formulation of the periodicity requirements. In addition, we extend the concept of periodicity to the more generic and useful concept of quasi-periodicity, providing more flexibility in the construction of a timetable while guaranteeing a perfectly periodic timetable for the customers. *Our contribution.*

- We introduce the concept of quasi-periodic timetables and its mathematical formulation.
- We extend an already efficient and effective train scheduling MILP formulation to generate a quasi-periodic timetable from scratch for a full set of train services.
- We report results using real data on real strategic scenarios.
- We discuss how the timetables that we produced were used to select and validate some of these scenarios for the future developments of the JÕren line in Norway.

2. The quasi-periodic train scheduling problem

We plan a set T of trains on a line, which consists of an ordered set of stations S connected by tracks. Trains travel through the stations even according to the station ordering (or *direction*) or in reverse order (or opposite direction). Every pair of adjacent stations is connected by one or two tracks. In the first case (*single-track section*), trains in both directions will share the available track. In the second case (*double-track section*), one track is reserved for trains in one direction and the other track for the other direction.

In a given day, every train $t \in T$ runs through a sequence $S(t) \subseteq S$ of consecutive stations. The *timetabling problem* amounts to finding an arrival time a_t^s and a departure time d_t^s for all $t \in T$ and $s \in S(t)$, which are expressed in *minutes from midnight*. Note that, the arrival time at the first station - where the train is originated - and departure time from the last station - where the train is terminated - are not defined, and the corresponding variable should not be introduced. However, to simplify the notation, we just assume that in the first (and last) station of $S(t)$, the arrival time equals the departure time. Since the planning horizon H is one day, that is 1440 minutes, we have that $0 \leq a_t^s \leq d_t^s \leq 1440$, and we let $H = \{0, \ldots, 1440\}$. In the sequel, for a departure (arrival) time $d \in H$, we also use the notation hh.mm(d), where $hh = \lfloor d/60 \rfloor$ is the departure (arrival) hour, whereas $mm = (d \mod 60)$ is the departure (arrival) minute.

To be feasible, the schedule (a, d) must not imply conflicts in the use of rail resources, neither in the line tracks nor in the stations - the exact definition of conflict will be given later.

We can now introduce the concept of a *periodic subset* of trains. Given a *periodic identifier* $c \in \{1, \ldots, n\}$, all trains of a periodic subset $T_c \subseteq T$ share a non-empty set of stations S_c, and, for any station in S_c, the trains in T_c leave the station at the same minute but at different (not necessarily consecutive) hours (for instance train 1 at 10.22 and train 2 at 12.22). In other words, if $q, r \in T_c$ and $s \in S_c$, then $mm(d_s^q) = mm(d_s^r)$ for all $q, r \in T_c$. There exist in general several (not necessarily disjoint) periodic subsets $T_c \in \{T_1, T_2, \ldots, T_n\}$. We also let $\overline{mm}_c$ be the departure minute associated with T_c at $s \in S_c$, namely $\overline{mm}_c(s) = mm(d_s^q)$ for any $q \in T_c$. Finally, with each periodic subset T_c we associate a reference station $r_c \in S_c$ and for each train in $t \in T_c$ we define a reference hour $h_t^{r_c} \in \{0, 1, \ldots, 23\}$, which is the hour $h_t^c = hh(d_t^{r_c})$ at which the train must depart from r_c.

In Fig. 3 we show an example of periodic timetable. The blue shaded train schedules with the same periodic id depart at the same minute of each hour, while the orange shaded train schedules are not associated to any periodic id and they can start anytime during the hour. Note that train schedules with different routes may have the same periodic id.

In our approach, we extend the above definitions and introduce the concept of *quasi-periodic subset* of trains. In this case, trains in the same (quasi-)periodic subset are not bound to depart at exactly the same minute, but they deviate moderately. First, we redefine the departure minute at station $s \in S_c$ as $mm_c(s) = \min_{q \in T_c} mm(d_s^q)$. Then we introduce a *flexibility* constant $\gamma_c \geq 0$ which defines the maximum delay a train can depart from s w.r.t. $mm_c(s)$, namely $mm_c(s) + \gamma_c \geq mm(d_s^q)$, for all $q \in T_c$. If needed, γ_c can of course depend on s as well, but we avoid the extra notation for simplicity.

We can now state our *Quasi-Periodic Timetabling Problem* (QPTP) more formally:

Problem 1. Given a railway line with set of trains T and stations S. For $c = 1, \ldots, n$, we are given quasi-periodic subsets T_c and stations S_c, flexibility constant γ_c and reference station $r_c \in S_c$, and, for each $t \in T_c$, reference hour $h_t^{r_c} \in H$. Then we want to find a conflict-free quasi-periodic timetable.

In the next section we describe our MILP formulation for the QPTP in more detail. We can identify two blocks of constraints: the first corresponds to the basic train scheduling problem, which is the standard model also for the generic non-periodic train scheduling problem (see, for instance, [40]). The second group of constraints corresponds to the quasi-periodic requirements.

3. A MILP formulation for the QPTP

Train timetabling (periodic or non-periodic) is an off-line variant of the basic *train scheduling* problem (whereas *train dispatching* is the on-line version). Given a railway network and a set of trains, train scheduling consists in finding a schedule for all trains that satisfies all operational constraints (e.g., safety constraints, station capacity constraints, blocking constraints, etc.).

The versions of train scheduling problem differ for specific additional constraints and for the objective function. In train

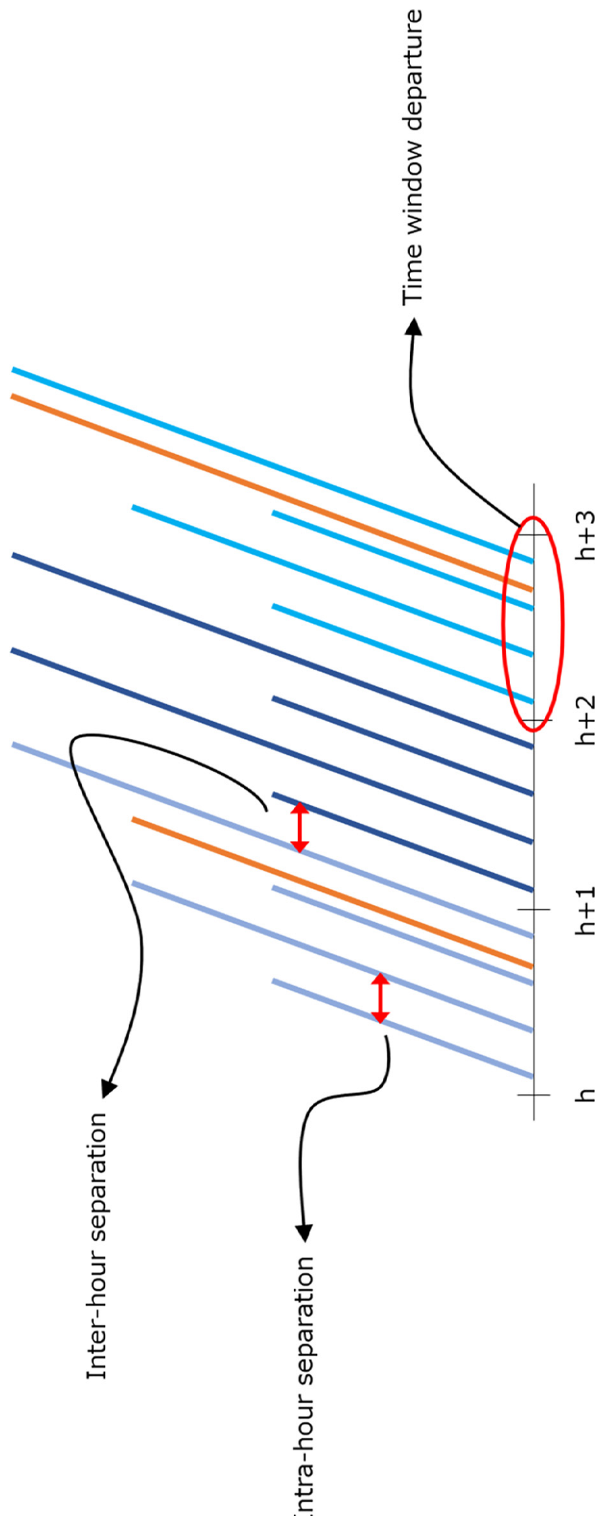

Fig. 3. A stylized mono-directional timetable. Periodic trains with shades of blue and non-periodic trains in orange. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

dispatching, given the current position of the trains and a reference schedule (i.e., the public timetable), the goal is to find a near-future (from now to the next few hours) schedule that minimizes a measure of the overall delay, i.e., a schedule that is (loosely speaking) as close as possible to the reference schedule. Train timetabling is somehow more intricate because one must produce a schedule that satisfies additional and more complicated constraints (e.g., there should be 4 trains per hour during peak hours and 2 per hour during non-peak hours).

We start by discussing a (classic) MILP model for the basic scheduling problem and then we present the additional constraints necessary to model the quasi-periodic timetabling.

3.1. The basic train scheduling constraints

In any classical train scheduling model, the movement of the train through the network is segmented into a sequence of atomic movements. For each atomic movement the minimum running time is given, and a feasible solution (*schedule*) is the time in which each train begins each of its atomic movements. The timetable (a, d) corresponds to the components of the solution associated with atomic movements of entering and leaving the stations. Note that the timetable is only a sub-vector of the entire solution, which must also specify the schedule of all other microscopic movements, for instance those inside the stations.

In order to build and solve our MILP model, we follow the decomposition approach presented in [41]. Since the approach is described in full detail in [41] and in other papers [28,40], we only present here the main features.

We consider the railway line as a sequence of alternating track and station resources and we decompose the problem into a *line problem* and a *station problem*. The optimal timetable (a^*, d^*) is computed by solving the MILP associated with the line problem. In the line problem, the micro-movements of the trains inside the stations are neglected and we only consider (explicitly) the potential conflicts on the tracks between successive stations. Specific feasibility cuts ensure that the timetable (a^*, d^*) is also feasible for the stations, namely that there exists a conflict-free schedule of the train movements inside the stations which is compliant with the timetable. As we will see, the number of these cuts can grow exponentially with the size of the problem, and so they are introduced iteratively by delayed row generation. The row generation problem is precisely the station problem, which amounts to find a schedule (and routing) for the trains in a station which is compatible with the arrival and departure times of trains - or prove that it does not exist.

Next, we describe the two problems in our decomposition, starting with the line problem and then the station problem. In particular, we will introduce and discuss the main variables and constraints, as they appear in our reference application context provided by Jernbanedirektoratet.

3.1.1. Line problem

A railway line is an alternating sequence of stations and tracks. Between each pair of consecutive stations we assume we have at most two tracks. When two tracks are available, one is reserved for the trains running in one direction while the other track is assigned to the opposite direction. If only one track is available, then trains in both directions alternate on the track. In any case, trains have no alternative routing options and the path through the line is fixed.

Minimum running time For each train $t \in T$ we let Δ_t^s be the minimum running time from $s \in S(t)$ to the next station and we let $f_t \in S(t)$ be the final station for t. The following constraints ensure that trains do not travel faster than they can:

$$a_t^{s+1} - d_t^s \geq \Delta_t^s, \qquad t \in T, s \in S(t) \setminus \{f_t\}, \tag{1}$$

where a_t^{s+1} denotes the arrival time of train t at the station next to s on the path of t.

Minimum dwell time If Ω_t^s is the minimum dwell time at station s for train t, we have:

$$d_t^s - a_t^s \geq \Omega_t^s, \qquad t \in T, s \in S(t). \tag{2}$$

Note that the dwell time also factors in the time necessary for the train to travel through the station.

Track conflict We say that we have a *conflict* (between two trains) whenever, according to some schedule, two trains are planned to occupy simultaneously the same track. Conflicts are not allowed and any schedule implying conflicts is infeasible. In other words, in any feasible schedule, for any two trains sharing a track, one must precede the other one on the track. Therefore, the first train must exit the track before the second train enters the track. To model this we introduce, for every ordered pair of trains $(q, r) \in T \times T$ with a common track e in their routes, variable $p_{q,r}^e$ which is 1 if and only if q precedes r on e. Then we have:

$$p_{q,r}^e + p_{r,q}^e = 1. \tag{3}$$

This equation and one of these two variables are of course redundant, since one could simply use $(1 - p_{rq}^e)$ in place of p_{qr}^e, but it helps the reading of subsequent constraints to have both of them explicitly. The same motivation applies to similar constraints that will appear later in the description of the model.

Assume[1] now that track e between two adjacent stations is travelled by train q from station u to station $u+1$, and by train r from station v to $v+1$. In double track sections, q and r can share a track e only if they run in the same direction, and thus station $u = v$ and $u+1 = v+1$. In single track sections, the trains may also run in opposite directions on the same track, in which case we have that $u = v+1$ and $u+1 = v$. In any case, if q precedes r on e, then q enters station $u+1$ before r leaves station v So, we have

$$d_r^v - a_q^{u+1} \geq -M p_{r,q}^e. \tag{4}$$

Indeed, if q precedes r on e, then $p_{rq}^e = 0$, which in turn implies $d_r^v \geq a_q^{u+1}$.

Station conflict We introduce now the constraints ensuring that a timetable (d, a) is feasible for the stations. To this end, we say that two trains $q, r \in T$ *meet* in $s \in S(q) \cap S(r)$ if they are simultaneously in s, i.e. $d_q^s \geq a_r^s$ and $d_r^s \geq a_q^s$. For all unordered pairs $\{q, r\} \subseteq T$, and $s \in S(q) \cap S(r)$, we introduce a binary *meet variable* $m_{\{q,r\}}^s$ which is 1 if q and r meet in s and 0 otherwise. Similarly, for all ordered pairs $(q, r) \in T \times T$, and $s \in S(q) \cap S(r)$ we introduce the binary variable $p_{q,r}^s$ which is 1 if and only if q arrives in s before r. Note that if trains q, r do not meet in station s, then either q leaves the station before r arrives, or r leaves the station before q arrives. This disjunctive requirement can be expressed by the following linear constraints:

$$\begin{aligned} & p_{q,r}^s + p_{r,q}^s = 1, \\ & d_r^s - a_q^s \geq -M\left(1 - m_{\{q,r\}}^s\right) \\ & d_q^s - a_r^s \geq -M\left(1 - m_{\{q,r\}}^s\right) \\ & a_r^s - d_q^s \geq -M\left(m_{\{q,r\}}^s + p_{r,q}^s\right) \\ & a_q^s - d_r^s \geq -M\left(m_{\{q,r\}}^s + p_{q,r}^s\right). \end{aligned} \tag{5}$$

In [41] it is showed that, under some mild assumptions, station feasibility only depends on the meeting vector m. Moreover, the feasibility constraints for a station s have the form

$$\sum_{\{q,r\} \in K} m_{\{q,r\}}^s \leq |K| - 1 \tag{6}$$

for $K \in \mathcal{K}^s$ where, for a station s, K is a set of train pairs such that at least one pair of trains cannot meet in s and $\mathcal{K}^s$ is the set of all such subsets for s.

Safety margin These constraints model a safety requirement for train operations. When two crossing trains meet in a station, the second train may be required to enter the station only some time after the tail of the first train has fully entered the station. This is usually called safety margin, and its value Γ_q^s depends both on the train and the station. Then, for every ordered pair of trains $(q, r) \in T \times T$ meeting in a station $s \in S$, we write the following constraints:

$$\begin{aligned} & a_r^s - a_q^s \geq \Gamma_q^s - M p_{r,q}^s \\ & a_q^s - a_r^s \geq \Gamma_r^s - M p_{q,r}^s. \end{aligned} \tag{7}$$

3.1.2. Station problem

The family $\mathcal{K}^s$ of infeasible subsets of train pairs for station $s \in S$ can (in principle) grow exponentially with the number of trains. So we generate the set and the associated family of constraints dynamically. In particular, given a station s and the arrival and departure times $(\bar{a}^s, \bar{d}^s)$ at s, along with the associated meet vector $\overline{m}^s$, the station problem for station s is the *separation problem* [42] for constraints of type (6). That is, it establishes if $\overline{m}_s$ is feasible for s or finds one set $\bar{K}$ of train pairs such that at least one pair of trains should not meet in s but $\sum_{\{q,r\} \in \bar{K}} \overline{m}_{q,r}^s = |\bar{K}|$.

Depending on the layout of the stations, one can apply different models to solve the station problem (see [41]). For our reference test instances, we adopted the *list coloring* model [41]. Stations are characterized by the set of platforms where trains can hold for embarking and disembarking passengers or just to wait for other trains to meet. Typically, in small stations, there is only one path from the incoming track to a given platform, and from a given platform to the outgoing track. In this situation, the route of the train through the station only depends on the platform assigned to that train (also if the train does not stop). We may also assume that the running time through the station is approximately the same for all routes. In general, not all platforms can be assigned to every train. Because two trains cannot occupy the same platform at the same time, the station problem reads as:

Given meet vector $\overline{m}^s$, assign to every train a feasible platform such that $q, r \in T$ get different platforms whenever $\overline{m}_{r,q}^s = 1$, or prove that such assignment does not exist.

In [41] we describe a MILP model for the station problem, and we also give sufficient conditions for when the separation of violated inequalities (6) can be carried out in polynomial time. Since the station problem is not the focus of the paper, and all details of our approach can be found in [41], we prefer to delve resolutely with the new contribution.

3.2. The quasi-periodic timetabling constraints

The previous section described how to model the (main) constraints of a basic train scheduling problem. In this section, we describe instead the additional constraints that characterize the quasi-periodic timetabling problem. These constraints are based on the desires of Jernbanedirektoratet for a feasible and ideal timetable. Many of them are implicitly derived from the specifics of a particular scenario (see, for example, Fig. 2), such as ensuring a “uniform” distribution of train services throughout each hour of the day. There are 4 basic types of constraints that we want to consider in a quasi-periodic timetable, some of which are highlighted in 3.

1. **Single train time window.** Each train $t \in T_c$, $c = 1, \ldots, n$, will depart from the reference station r_c in the one-hour time interval $[h_t^{r_c}, h_t^{r_c} + 1)$, where $h_t^{r_c} \in \{0, \ldots, 23\}$ is the given reference (starting) hour.

[1] With some abuse of notation, if s is a station on the path of a train, we denote by $s+1$ the next station.

2. **Intra-hour separation.** Trains travelling in the same direction within a particular reference hour should be scheduled at regular intervals across the hour. This is generally the preferred option for passengers and was required in our scenarios. This is achieved by requiring a certain time separation between pairs of trains. Note that different pairs of trains may have different separation requirements. For example, we may want a pair of long-distance trains to be separated by at least half an hour, while a pair of local trains to be separated by at least 15 minutes.
3. **Inter-hour separation.** The intra-hour separation rules should apply also to each pair of trains that travel in the same direction but on two consecutive reference hours. For a particular pair of trains, the desired separation is the minimum intra-hour separation required by each train in its corresponding reference hour.
4. **Quasi-periodicity.** All trains in the same periodic subset T_c must leave station s at (approximately) the same minute after the reference hour of the train, for all $s \in S_c$.

As discussed in Section 2, our model in principle allows for trains belonging to the same periodic subset T_c to run through different sets of stations. However, they all must share a set of stations $S_c \neq \emptyset$ from which the railway infrastructure manager can choose a reference station $r_c \in S_c$. For trains that are not required to be quasi-periodic, the reference station will simply be the first station in their route. For instance, in the JÕrbane, in the direction “away” from Stavanger, all local and long-distance trains depart from Stavanger. In the direction “towards” Stavanger, all local and long-distance trains stop in Skeiane.

All the constraints above can be modeled using the variables introduced in the previous section plus a few extra ones. In the following, for ease of description, we represent variables and constraints only associated with trains travelling in one direction (e.g., away from Stavanger). In fact, the constraints described below are relevant only for trains travelling in the same direction. Moreover, we assume that departure and arrival times at stations, as well as all other relevant time constants, are expressed in seconds.

Single train time window For a train $t \in T_c$ with given reference (departure) hour $h_t^{r_c} \in \{0, \dots, 23\}$ at its reference station $r_c \in S_c$, we can write:

$$3600\, h_t^{r_c} \leq d_t^{r_c} \leq 3600\left(h_t^{r_c}+1\right)-1 \qquad t \in T_c, c=1,\dots,n. \tag{8}$$

Intra-hour separation For a fixed hour $h \in \{0, \dots, 23\}$ and a station $s \in S$, let T_h^s be the set of trains with reference hour h at station s. Recall that with every ordered pair of (distinct) trains $(q,r) \in T_h^s \times T_h^s$, we associate a binary variable $p_{q,r}^s$ which is 1 if and only if q precedes r in station s.

The separation between trains with same reference hour ensure a certain distribution within the hour. Our objective is to distribute local trains along the entire reference hour and place non-local trains between them. For ease of explanation, consider the partition $T_h^s = L_h^s \cup N_h^s$, where L_h^s, N_h^s are the subsets of local and non-local trains, respectively.

Local trains. For $h \in H$ and $s \in S$, consider the *local wavelength* $\Lambda_h^s = \lfloor \frac{3600}{|L_h^s|} \rfloor$. Let $(q,r) \in L_h^s \times L_h^s$ be an ordered pair of distinct trains, where s is a relevant shared stop. Then we have:

$$d_r^s - d_q^s \geq \Lambda_h^s - l_{q,r} - M p_{r,q}^s \tag{9}$$

where $0 \leq l_{q,r} \leq \bar{l}_{q,r}$ is a (bounded) slack variable which measures the intra-hour separation constraint violation, and M is a large constant. Note that the inequality holds only when r follows q in s, i.e. when $p_{r,q}^s = 0$.

Non-local trains. Since there is usually only one non-local train per hour, we are not interested in suggesting a specific intra-hour separation between non-local trains. However, we would prefer them to be equally spaced between a pair of local trains. Since the local wavelength is Λ_h^s, we can simply define the *non-local wavelength* as $\frac{\Lambda_h^s}{2}$ and apply it only to pairs of local/non-local trains. Then, for each pair of ordered trains $(q,r) \in N_h^s \times L_h^s \cup L_h^s \times N_h^s$ we have:

$$d_r^s - d_q^s \geq \frac{\Lambda_h^s}{2} - l_{q,r} - M p_{r,q}^s. \tag{10}$$

Inter-hour separation We consider now two adjacent hours h and $h+1$. The inter-hour margin is defined as the minimum of the two associated wavelengths, i.e., $\Psi_{h,h+1}^s = \min\{\Lambda_h^s, \Lambda_{h+1}^s\}$ for local trains and $\Theta_{h,h+1}^s = \min\{\frac{\Lambda_h^s}{2}, \frac{\Lambda_{h+1}^s}{2}\}$ for non-local trains. Then, for every pair of train $(q,r) \in L_h \times L_{h+1}$ and a relevant stop we have

$$d_r^s - d_q^s \geq \Psi_{h,h+1}^s - l'_{q,r} \tag{11}$$

where l'_{qr} is a non-negative slack variable. Similarly, for every pair of train $(q,r) \in N_h \times L_{h+1} \cup L_h \times N_{h+1}$ and a relevant stop we have

$$d_r^s - d_q^s \geq \Theta_{h,h+1}^s - l'_{q,r}. \tag{12}$$

Quasi-periodicity (Quasi-)Periodicity is enforced for a specific periodic id, for a specific station, on a specific direction. For each periodic id $c \in C$, and for each station $s \in S_c$, we introduce a variable $z_c^s \in \mathbb{R}$ to represent the base time of this specific *periodic pattern*, i.e., the minute $\overline{mm_c}(s)$ of the hour in which each train $t \in T_c$ is supposed to leave from every station $s \in S_c$. Then, for each $t \in T_c$, the constraints that enforce the periodicity pattern associated with $c \in C$ can be written as follows:

$$\begin{aligned} z_c^s &\leq d_t^s - h_t \\ z_c^s &\geq d_t^s - h_t - \rho_{c,t}^s \end{aligned} \tag{13}$$

where $h_t \in H$ is the starting hour of train t, and $\rho_{c,t}^s \in \mathbb{R}_+$ is a slack variable that allows for a deviation from the base time. The slack variable cannot exceed the flexibility constant γ_c introduced in Section 2, namely $\rho_{c,t}^s \leq \gamma_c$. In our instances, we let $\gamma_c = 5$ minutes. Note that $\rho_{c,t}^s$ is always non-negative. For example, if the base time is h:15, then all the corresponding periodic trains can start at $[h{:}15, h{:}20]$, but never before h:15. Indeed, it is usually acceptable for a train to be delayed a couple of minutes from its base time. It is not acceptable the opposite.

3.3. Objective function

The (linear) objective function consists of the minimization of a linear combination of the following terms:

1. A penalty $\delta_t^s \in \mathbb{R}_+$ for the extra time spent in track (if more than the minimum running time), which can be modelled by complementing the constraints in (1) with:
$$a_t^{s+1} - d_t^s \leq \Delta_t^s + \delta_t^s, \qquad t \in T, s \in S(t) \setminus \{f_t\}; \tag{14}$$
2. A penalty $\omega_t^s \in \mathbb{R}_+$ for the extra time spent in station (if more than the dwell time), which can be modelled by complementing the constraints in (2) with:
$$d_t^s - a_t^s \leq \Omega_t^s + \omega_t^s, \qquad t \in T, s \in S(t); \tag{15}$$
3. A penalty $l_{q,r}$ for not satisfying the intra-separation constraints in (9) and (10);
4. A penalty $l'_{q,r}$ for not satisfying the inter-separation constraints in (11) and (12);
5. A penalty $\rho_{c,t}^s$ for not satisfying the periodicity constraints in (13).

Therefore, the objective function will be (omitting the indices' sets for simplicity):

$$\min \sum \delta_t^s + \sum \omega_t^s + \sum l_{q,r} + \sum l'_{q,r} + \sum \rho_{c,t}^s \tag{16}$$

Each of these components (or even each single variable) can of course be weighted according to the preferences of the railway infrastructure manager or, in this particular case, of Jernbanedirektoratet.

A final remark concerns the relation of this model with the one presented in [28], with which we share the decomposition approach and the solution method, as mentioned in the introduction. Indeed, the model presented in paper[28] also allows for some deviation from periodicity. However, in [28] the modelling of such relaxation - namely constraints (10) and (11) (in [28]) - is much looser and allows for timetables that are non-quasi-periodic (in the sense defined in the current paper). This is because the model in [28] does not include the definition of what in the current paper is called *base time* (minute), which must be the same for every train in a periodic class. Indeed, the constraints (10) and (11) were introduced in [28] for ensuring that successive trains (intra-hour periodicity) were close to be periodic. But, by only constraining pairs of successive trains, small deviations could sum up and the schedules of two arbitrarily distant trains in a same periodic class may deviate too much. It must be said that the goal in [28] was the insertion of a number of freight trains in a given passenger-train periodic timetable, and so the risk of accumulating deviations was somehow mitigated. By extending constraints (10) and (11) to all pairs of trains in a same periodic class (and not only to successive trains), we could enforce quasi-periodicity as defined in this paper, but the number of these constraints would grow as the square of the number of trains in the periodic class. In contrast, only a linear number of constrains (13) appear in the current model.

4. Solution algorithm

As we have seen in the previous sections, the number of constraints of the model may grow very large even for relatively small problems. However, one can efficiently exploit the structure of the model to dynamically generate all the necessary constraints through an ad-hoc delayed-row-generation scheme built on top of a MILP solver, as also described in [40] and [28].

In particular, we apply this treatment to the constraints in (4), (5), (6) and (7).

In fact, these are the ones that, loosely speaking, give rise to the combinatorial complexity of the model. Typically, only few of those are really necessary to prove optimality (or infeasibility).

So, in order to solve the model P defined by constraints (1) to (15) and objective function (16), we solve a sequence of increasingly large MILPs $P^0, P^1, \ldots, P^q$, where P^0 is obtained by the original full problem P by removing some constraints, and P^i is obtained from P^{i-1} by re-inserting some of the initially removed constraints. In our approach, P^0 is obtained from P by dropping all constraints of type (4), (5), (6), and (7). Note that all the problems in the sequence are relaxations of the full problem P.

At the i-th iteration we solve problem P^i to optimality.

1. If P^i is infeasible, then P is infeasible (since P^i is a relaxation), and we are done. Otherwise, let $y^i = (a^i, d^i, p^i, m^i, z^i)$ be the current solution.
2. If y^i violates some of the inequalities of type (4) or (7) not included in P^i, we add these to P^i (generating P^{i+1}) and iterate.
3. If y^i violates some of the inequalities of type (6) not included in P^i, we add these to P^i (generating P^{i+1}) and iterate.
4. If y^i does not violate any missing inequality then y^i is also feasible and thus optimal for P, and we are done.

The identification (i.e., *separation*) of violated inequalities (4) and (7) at Step 2 can be easily done by inspection. Informally, this corresponds to checking whether the current timetable (a^i, d^i) implies a conflict between two trains in some tracks between two stations, or any of the safety margins are violated. The separation of violated inequalities (6) at Step 3 is carried out by solving a station problem, for all stations in the line.

Note that we do not necessarily need to solve each P^i to optimality before separating new constraints. We could simply separate them every time a MILP solution algorithm finds an integer feasible solution, that is in the feasible leaves of the branch-and-bound tree generated by the algorithm. However, we found that is usually more convenient to wait until a “good enough” solution has been found because commercial MILP solvers tend to generate many non-optimal feasible solutions during the beginning of the search (for example, through quick heuristics), which would lead to the separation of constraints that are less likely to be binding in the final solution. Here the term “good enough” is left vague on purpose and it stands for a solution that it likely to be close to the optimal one. In fact, due to the poor dual bounds provided by this type of big-M formulations, we found that it is not uncommon to find what is going the be the actual optimal solution at the beginning of the search tree, and then perhaps spend few millions nodes to prove its optimality. So, an idea to speed up the solution algorithm is to look for an optimal solution only after finding a feasible solution that is *good-enough* (for example, if the MIP gap is less than 10% or more than 100,000 nodes have been explored since the last incumbent). Whenever we find a good-enough solution that does not violate any missing inequalities (i.e., feasible but not necessarily optimal), then we solve again the problem to the optimum. If the incumbent did not change or the timetable associated with the new incumbent is feasible, then we found the optimal solution. Otherwise, we add the violated inequalities and we start again looking for a good-enough solution. This basically creates a sequence of feasible good-enough solutions, until one of these is proven to be optimal. In our experiments with slightly smaller problems than the ones considered in Section 5, we found that in many cases the first good-enough solution found by algorithm was indeed also the optimal one.

But we can do even more. Since current commercial MILP solvers do not support column generation within the branch-and-cut algorithm, postponing the separation of constraints actually gives us the opportunity to speed up the solution process significantly by a couple of simple observations: the variables p^e appear only in (3) and (4), while variables m appear only in (5)-(7); moreover, constraints (3) and (5) are necessary for the correctness of the formulation only in conjunction with the constraints (4), (6), (7) that are included in the aforementioned row generation approach. Therefore, not only we can delay the generation of the constraints (3) and (5) until the corresponding constraints (4), (6), (7) have been generated, but we can also delay the generation of the corresponding decision variables p^e, p^s and m. In most of the cases, these simple considerations significantly reduce the dimension of the initial problem P^0.

5. Real life experiments and future developments

In this section, we describe the results on the experiments that we carried out together with Jernbanedirektoratet on real data and real what-if scenarios. The base line is given by the current local train services (as of year 2020) summarized in Fig. 2. Note that the non-local train services do not change in our test scenarios and the current relevant railway infrastructure consists of double tracks between Stavanger and Sandnes and single tracks afterwards (with 4 passenger platforms in Stavanger). The what-if scenarios that Jernbanedirektoratet wanted to investigate were as follows (see also Table 1):

- **Scenario 1**: Original railway infrastructure but the 2 short-distance trains per hour between Stavanger and Sandnes would be extended to Ganddal (see Fig. 4).

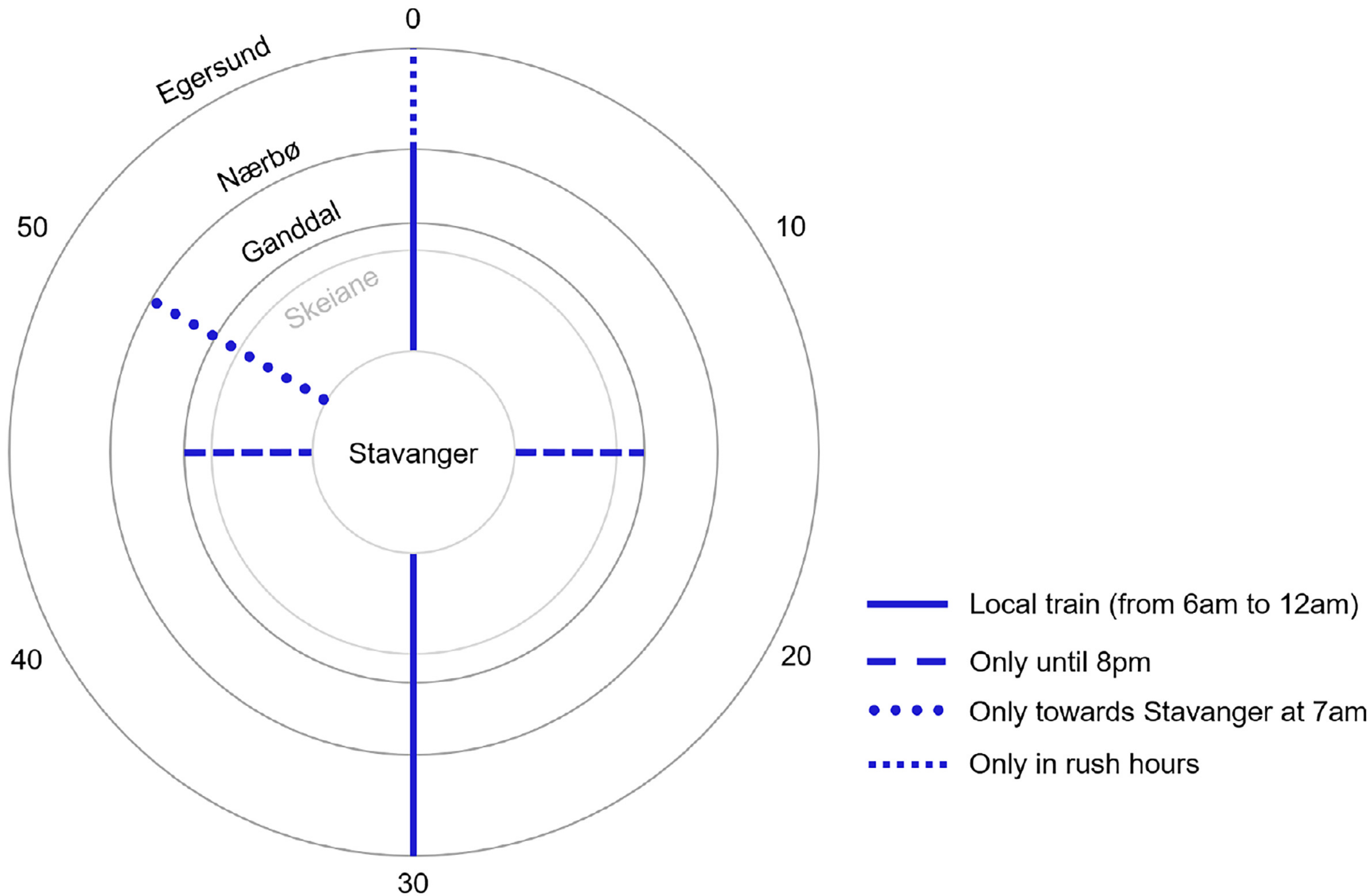

Fig. 4. The periodic pattern of the first test scenario. Compared to the original scenario, two local trains per hour are extended from Skeiane to Ganddal.

Table 1
A brief description of the characteristics of the different scenarios.

	Infrastructure		Train services per hour form STV			
	Plat. in STV	Double track to	To SAS	To GAN	To NB	To EGS
Current	4	SAS	4	2	2	1
Scenario 1	4	SAS	4	4	2	1
Scenario 2	6	SAS	6	4	2	1
Scenario 3	4	NB	4	4	4	1
Scenario 4	6	NB	6	6	6	1

- **Scenario 2**: Similar to Scenario 1 but we increase the turnaround capacity in Stavanger and we keep the existing short-distance StavangerSandnes trains when adding the new short-distance trains between Stavanger and Ganddal (see Fig. 6).
- **Scenario 3**: Both the double track and the two short-distance trains per hour between Stavanger and Sandnes are extended to NŐrbꝕ (see Fig. 8).
- **Scenario 4**: Mixing Scenario 2 and Scenario 3, we get increased turnaround capacity in Stavanger, double track to NŐrbꝕ, and the 2 short-distance trains per hour between Stavanger and Sandnes become 4 trains per hour between Stavanger and NŐrbꝕ (see Fig. 10).

The model and the algorithm described in Sections 3 and 4 have been implemented in.NET and run on an Intel i7-7700HQ @ 2.8 GHz with 32 GB of memory using IBM CPLEX 12.8. At first we tried to solve the entire model as is, but we quickly found it to be particularly challenging, especially due to the fact that trains within the same hour do not have a fixed order. This flexibility adds an enormous amount of complexity and makes for very poor dual bounds, since each particular order of trains is likely to generate a completely different set of conflicts (each order almost creating an independent subproblem). In practice, strategic planners usually have already a good idea of the particular order of train services they want to enforce, having to take into consideration other specific customer preferences. However, we decided to keep this flexibility and we tried to tackle the issue by mimicking a two stage approach that is also common in the manual timetabling process. In the first stage, we solve a simpler problem with only local and non-local trains (i.e., without freight trains). While in the second stage, we solve the problem with all trains, but we constrain the base time variables of the local and non-local trains $z_c^s \in [\bar{z}_c^s - 3 \text{ minutes}, \bar{z}_c^s + 3 \text{ minutes}]$ where $\bar{z}_c^s$ is the base time found during the first stage. This way, the order of periodic trains in the second stage is fixed, but their departure times are still flexible. The main reason for following this approach has to do with the fact that freight trains are not particularly relevant in the strategic timetabling process because: (1) they don't have to be periodic, (2) their departure time is much more flexible, and (3) they have lower priority compared to passenger trains. In other words, we just need to make sure that a certain scenario is able to accommodate a certain number of freight trains.

Feasibility is indeed the main purpose of strategic timetabling, making sure that there exists a conflict-free timetable satisfying certain basic requirements, such as periodicity or number of train services per hour. Notice that the objective of our model contains only preferences, while all these basic requirements are encoded in the model as hard constraints. This means that any feasible solution could already be considered as acceptable.

Taking into account these considerations, we solved the first stage with a timelimit of 12 hours, assuming a solution to be good-enough (see end of Section 4) if the gap is smaller than 10% or the incumbent did not change during the last 10,000 explored nodes. As mentioned above, these problems suffer from poor dual bounds and they are very difficult to solve to the optimum. For example, the first stage model for scenario 1 produced a feasible timetable after 1 h, but then spent the remaining 11 hours (in which CPLEX explored around 1 million nodes) to prove a gap of only 56%. However, the incumbent had a small objective value and during this time never changed. Our experience with these problems tells us that in these conditions the solution we found was likely to be either the optimal solution or close to the optimal one. Similar results were obtained in the other scenarios as well.

The second stage tells a different story, once the order of trains is fixed. We set again a time-limit of 12 hours, assuming a solution to be good-enough if the gap is smaller than 10% or the incumbent did not change during the last 100,000 explored nodes. Depending on the difficulty of the scenario, we were able to solve the problem either to optimality (Scenario 1) or to a gap of 10% (Scenario 4), with the other scenarios in between. On average, each the second stage problem explored a total of about 3 million nodes within 100 iterations, solving about 1000 conflicts. The size of the models at the first iteration was 20 to 30 thousand rows, and 13 to 21 thousands columns. In the last iteration, due to row and column generation, the number of rows was about 20% higher, while the number of columns about 10% higher.

The best feasible timetables are presented in Figs. 5, 7, 9, and 11. Note that the only available alternative, namely manual timetabling, is not even close to being competitive in terms of the time required to create a feasible timetable, usually ranging from several days to several weeks.

These strategic timetables were used in the following process to determine which scenarios to select for the future expansion of the JŐren line. The timetables were exported in the widely used railML format, fed into Jernbanedirektoratet's passenger transport model, and used to perform a preliminary cost-benefit analysis of the different scenarios. The worst-performing scenario (Scenario 3) was discarded, and some manual improvements were made to the remaining scenarios before re-running the transport model and updating the cost-benefit analysis. Even though the timetables were near-optimal, these manual modifications improved the cost-benefit ratio. This is to be expected, since the objective function of our model does not take into consideration those aspects that may require a complex simulation (e.g., passenger satisfaction). In the end Scenarios 1, 2, and 4 were recommended for different time horizons (i.e., few years, several years, and >10 years). In other words, the timetables produced by our algorithm were successfully used to select and validate future scenarios for the development of the JŐren line.

Interestingly, Jernbanedirektoratet was not certain that it would have been possible to produce a feasible timetable for Scenario 4. Even with the upgraded infrastructure, the mix of freight and frequent passenger trains between NŐrbꝕ and Orstad saturates the line. The concern proved somewhat right. In fact, we were not able to find a feasible timetable for Scenario 4 when enforcing perfect periodicity. The complexity and the high level of utilization of the railway line in Scenario 4 made it impossible to run perfectly periodic trains (i.e., the corresponding instance was proven to be infeasible). However, inspired by the challenge of producing

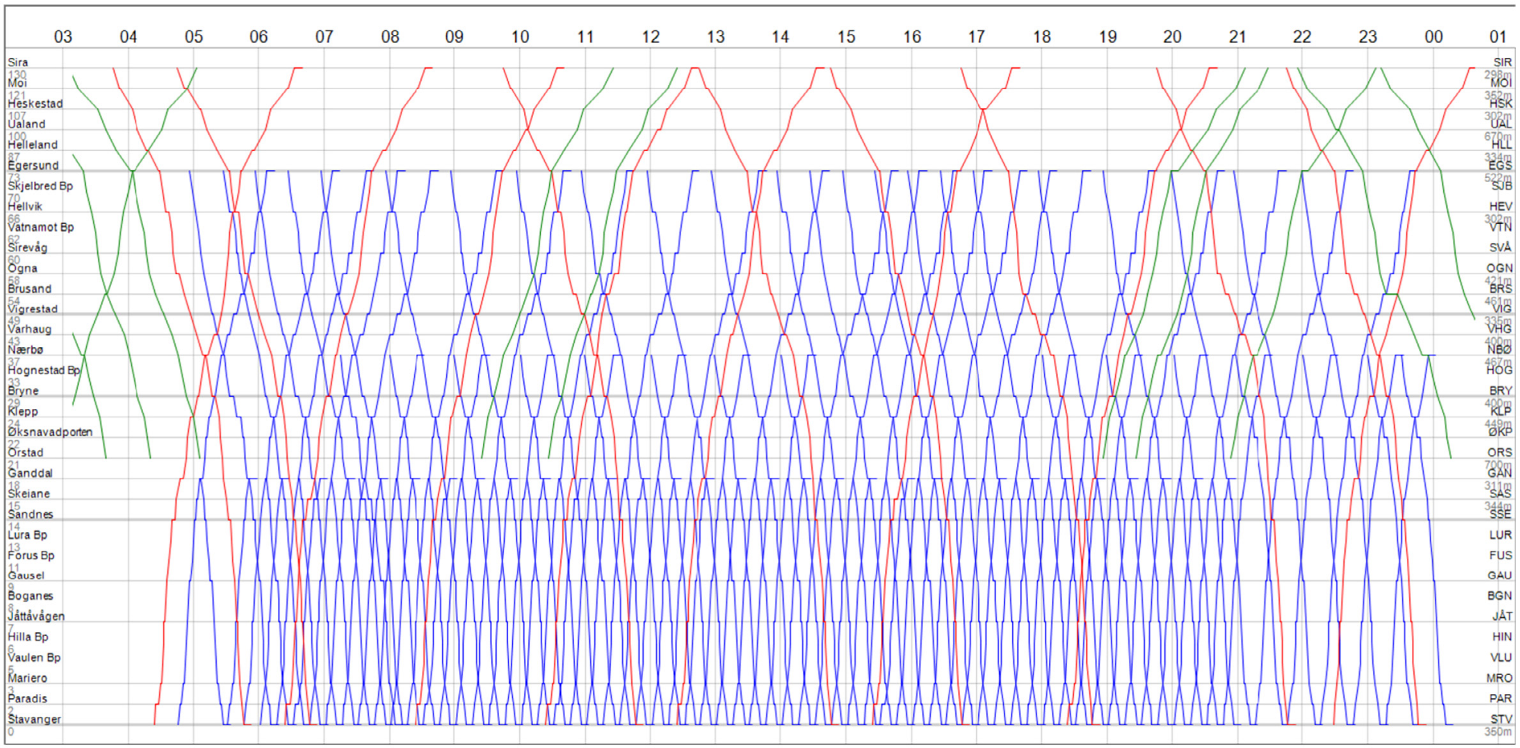

Fig. 5. The quasi-periodic (conflict-free) timetable for the first test scenario.

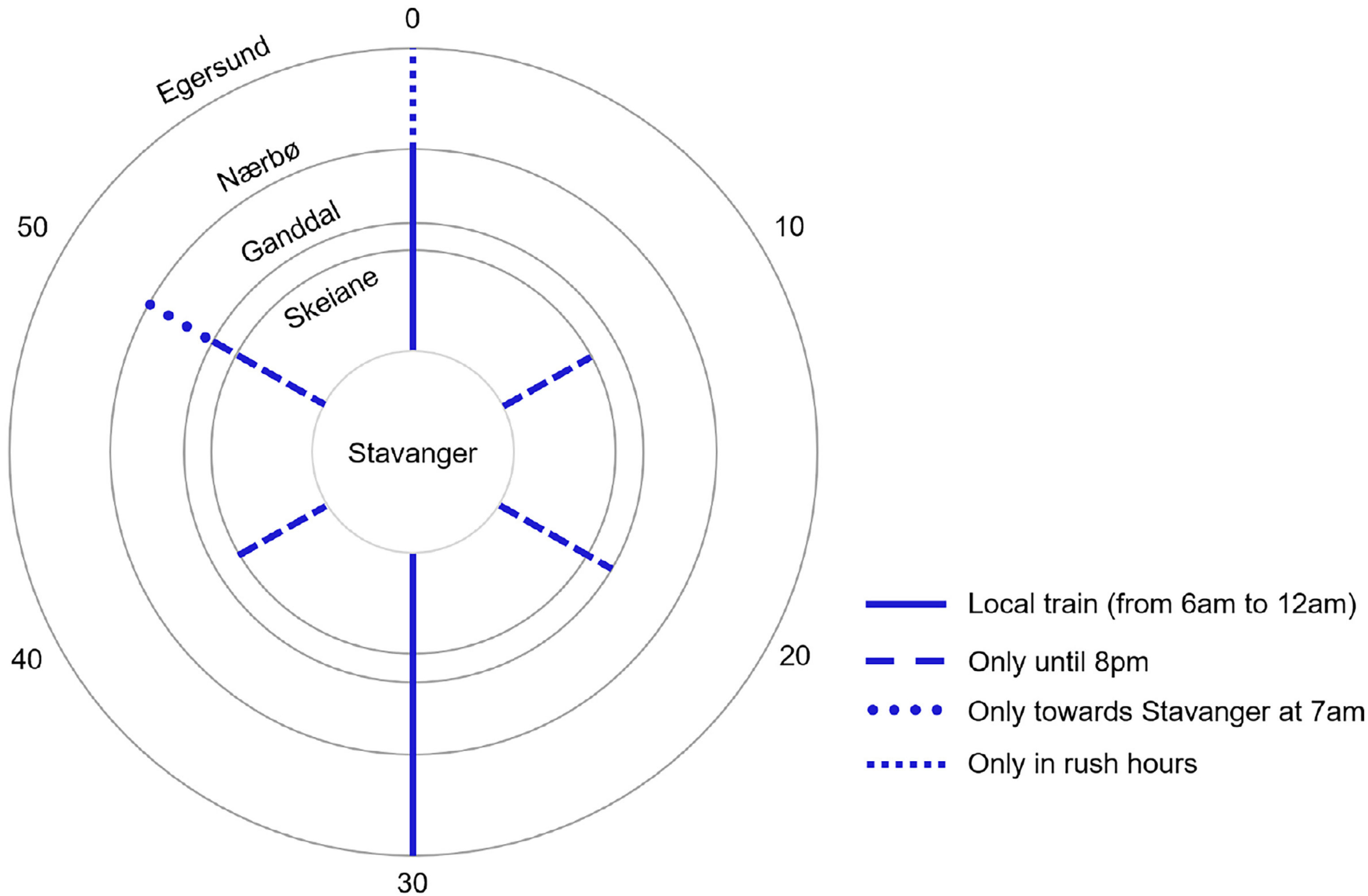

Fig. 6. The periodic pattern of the second test scenario. Compared to the original scenario, there are two extra local trains per hour from Stavanger to Ganddal. Morover, the turnaround capacity of Stavenger has been raised from 4 to 6 local trains.

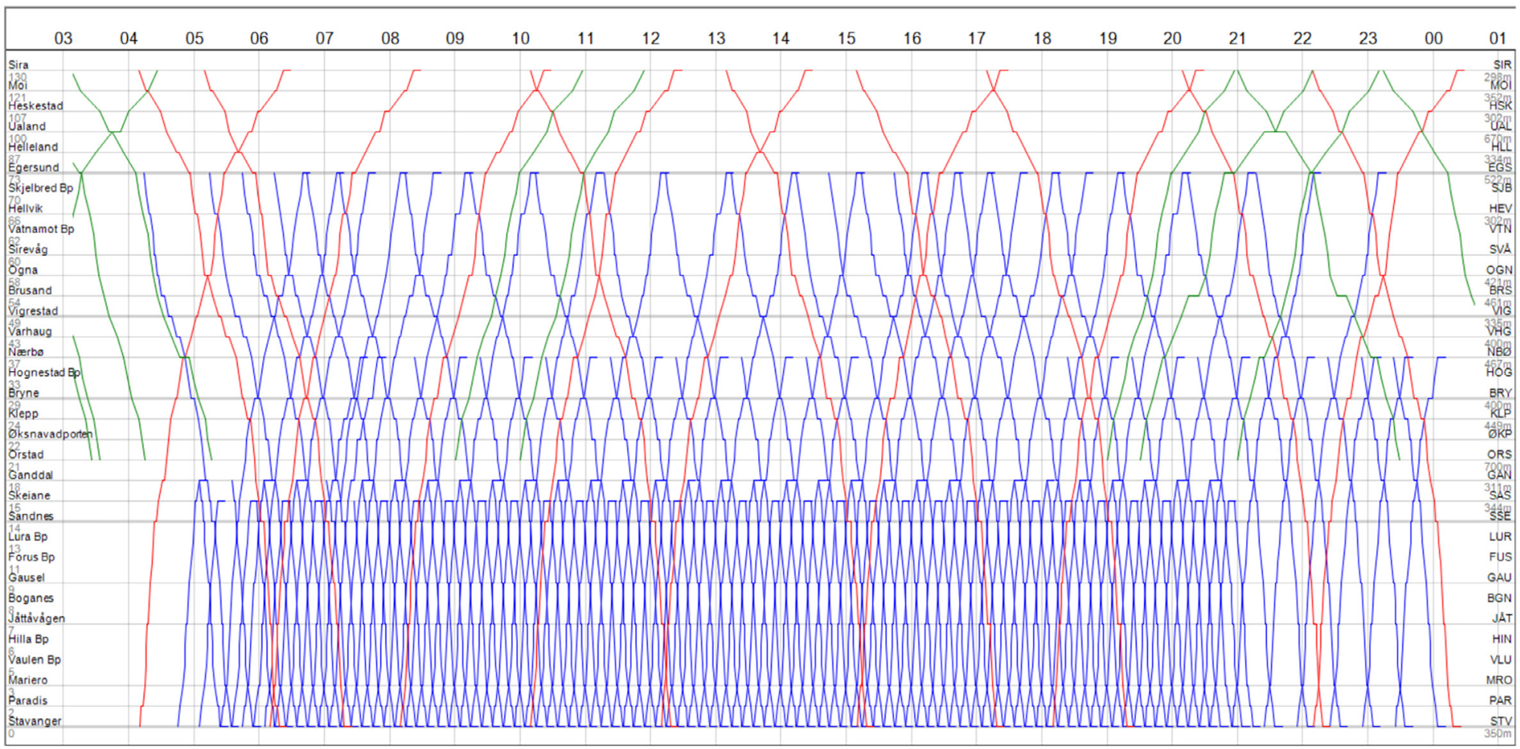

Fig. 7. The quasi-periodic (conflict-free) timetable for the second test scenario.

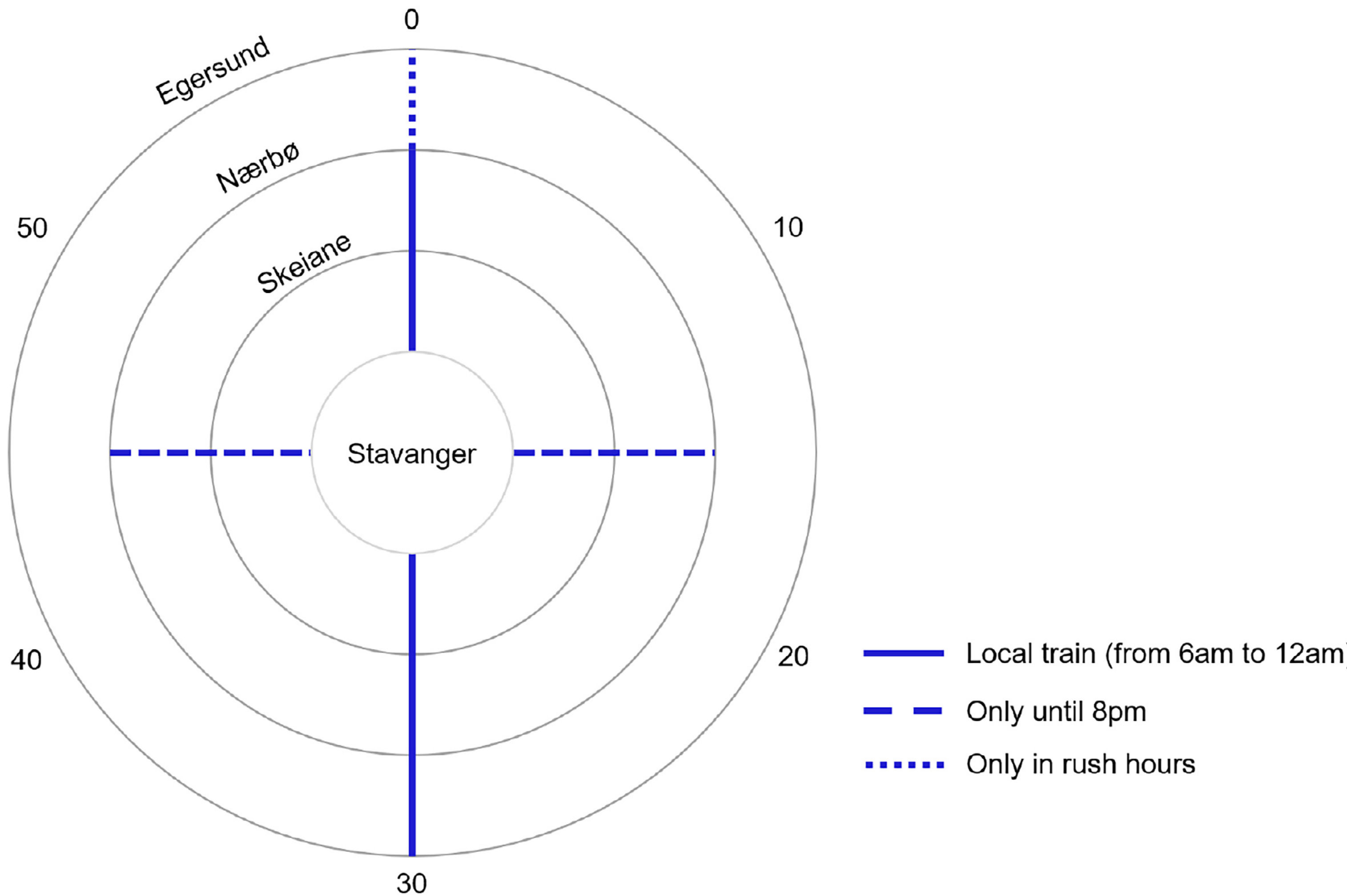

Fig. 8. The periodic pattern of the third test scenario. Compared to the original scenario, the two short local trains per hour have been extended from Skeiane all the way to Nærbø. Moreover, the double track has been extended from Skeiane to Nærbø.

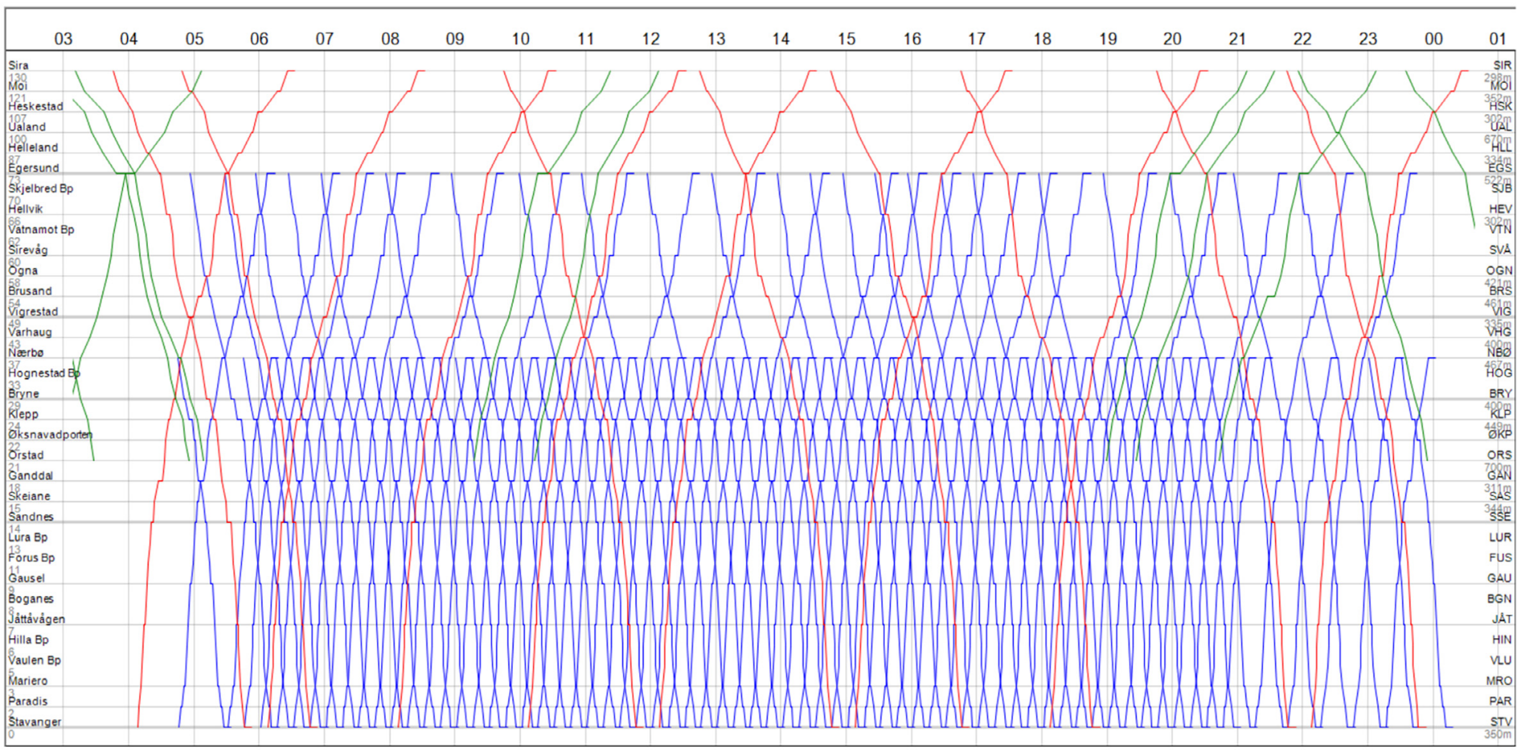

Fig. 9. The quasi-periodic (conflict-free) timetable for the third test scenario.

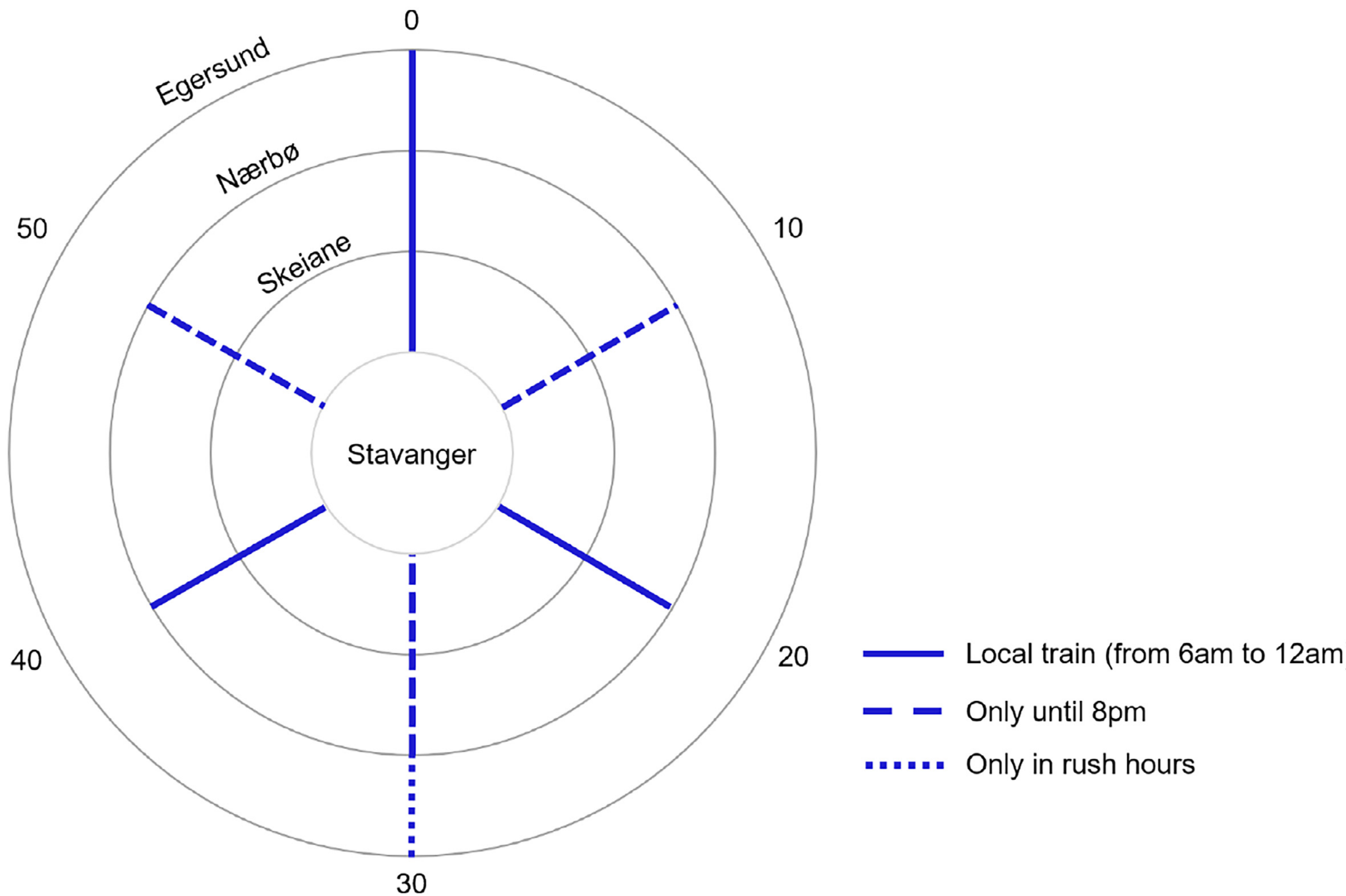

Fig. 10. The periodic pattern of the fourth test scenario. This is a mix of the second and third scenarios. There are now 4 local trains per hour running from Stavanger up to NÖrbφ (plus 2 to Egersund). The double track has been extended until NÖrbφ, and Stavanger now has turnaround capacity for 6 trains instead of just 4. The feasibility of this scenario was not certain, even with the upgraded infrastructure.

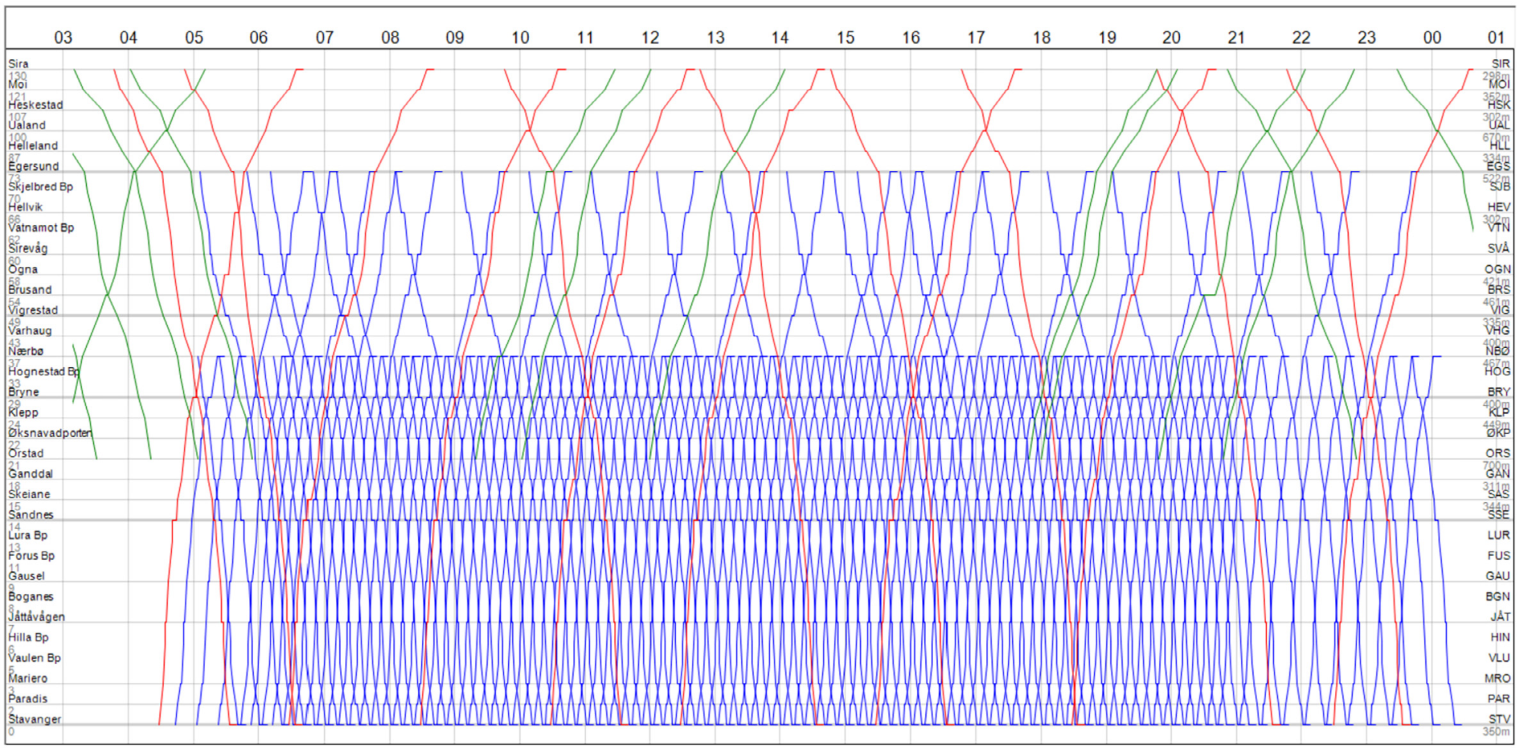

Fig. 11. The quasi-periodic (conflict-free) timetable for the fourth test scenario. Note that it did not exist a perfectly-periodic timetable for this scenario.

a feasible timetable for this scenario, we developed the concept of quasi-periodicity, which we ultimately applied to all scenarios with a maximum deviation of 3 minutes. Thanks to that, it is still possible to present a perfectly periodic timetable to the public while running the trains in a slightly less periodic fashion. It is interesting to note that the quasi-periodicity slack variables were almost all equal to zero in all scenarios. This is of course not surprising for Scenarios 1–3, since perfect periodicity is feasible. In Scenario 4, we identified only a particular set of periodic trains for which all the quasi-periodicity slack variables associated with the Bryne station had a value between 19 and 29 seconds. Considering the complexity of the train interactions in such a dense timetable, it is hard to say exactly why this was the case, though.

In any case, this result demonstrates that while perfect periodicity creates very passenger-friendly timetables, it may come at the cost of capacity. In highly saturated areas, strict periodicity will leave gaps between trains that cannot be used. By relaxing the periodicity constraints and adding a penalty for slight variations in the intervals between trains, we try to balance these conflicting goals to provide the most benefit for the passengers. Note, however, that the introduction of quasi-periodicity slightly hinders the computational performance. For example, the second stage of Scenario 1 was solved to optimality in about 2 hours with quasi-periodicity, but only in half an hour with perfect periodicity enforced (everything else the same).

While quasi-periodicity was a crucial addition to complement our timetabling model, there are a few aspects we want to improve upon. First, on the algorithmic side, it would be interesting to consider new decomposition approaches in order to tackle larger instances, so as to coordinate the timetables of several lines at the same time (see, for example, [43]). Next, on the modelling side, it would be interesting to develop effective ways to model uncertainty so as to produce timetables that are more robust against unwanted delays or disturbances (see, for example, [8]). As a final remark, we also would like to report that the success of the prototype tool for strategic timetabling discussed in this paper, among other things, inspired Jernbanedirektoratet to pursue the idea of contributing to a commercial tool for strategic planning by issuing a public tender aimed at financing the development of such tool.

Authors statement

All authors contributed equally to the development of the model and the writing of the paper.

Data availability

The authors do not have permission to share data.

References

[1] Kroon L, Huisman D, Abbink E, Fioole P-J, Fischetti M, Maróti G, et al. The new dutch timetable: the or revolution. Interfaces (Providence) 2009;39(1):6–17.
[2] Hacon. TPS Suite. https://www.hacon.de/en/solutions/tpssuite/.
[3] TRENOlab. TRENOplus. https://www.trenolab.com/tools/trenoplus/.
[4] VIA Consulting & Development. LUKS. https://www.via-con.de/en/development/luks/.
[5] Harrod SS. A tutorial on fundamental model structures for railway timetable optimization. Surv Oper Res Manag Sci 2012;17(2):85–96.
[6] Cacchiani V, Toth P. Nominal and robust train timetabling problems. Eur J Oper Res 2012;219(3):727–37.
[7] Caimi G, Kroon L, Liebchen C. Models for railway timetable optimization: applicability and applications in practice. J Rail Transp Plan Manag 2017;6(4):285–312.
[8] Cacchiani V, Toth P. Robust train timetabling. In: Handbook of Optimization in the Railway Industry. Springer; 2018. p. 93–115.
[9] Galli L, Stiller S. Modern challenges in timetabling. In: Handbook of Optimization in the Railway Industry. Springer; 2018. p. 117–40.
[10] Lusby RM, Larsen J, Ehrgott M, Ryan D. Railway track allocation: models and methods. OR Spectrum 2011;33(4):843–83.
[11] Kümmling M, Großmann P, Nachtigall K, Opitz J, Weiß R. A state-of-the-art realization of cyclic railway timetable computation. Public Transp 2015;7(3):281–93.
[12] Mascis A, Pacciarelli D. Job-shop scheduling with blocking and no-wait constraints. Eur J Oper Res 2002;143(3):498–517.
[13] Lamorgese L, Mannino C, Pacciarelli D, Krasemann JT. Train dispatching. Handbook Optim Rail Ind 2018:265–83.
[14] Queyranne M, Schulz AS. Polyhedral approaches to machine scheduling. Citeseer; 1994.
[15] Mannino C, Mascis A. Optimal real-time traffic control in metro stations. Oper Res 2009;57(4):1026–39.
[16] Harrod S. Modeling network transition constraints with hypergraphs. Transp Sci 2011;45(1):81–97.
[17] Boland NL, Savelsbergh MW. Perspectives on integer programming for time-dependent models. Top 2019;27(2):147–73.
[18] Cacchiani V, Furini F, Kidd MP. Approaches to a real-world train timetabling problem in a railway node. Omega (Westport) 2016;58:97–110.
[19] Cacchiani V, Caprara A, Toth P. Non-cyclic train timetabling and comparability graphs. Oper Res Lett 2010;38(3):179–84.
[20] Caprara A, Fischetti M, Toth P. Modeling and solving the train timetabling problem. Oper Res 2002;50(5):851–61.
[21] Caimi G, Burkolter D, Herrmann T, Chudak F, Laumanns M. Design of a railway scheduling model for dense services. Netw Spat Econ 2009;9(1):25–46.
[22] Talebian A, Zou B. Integrated modeling of high performance passenger and freight train planning on shared-use corridors in the us. Transp Res Part B 2015;82:114–40.
[23] Hao W, Meng L, Corman F, Long S, Jiang X. A train timetabling and stop planning optimization model with passenger demand. In: RailNorrköping 2019. 8th International Conference on Railway Operations Modelling and Analysis (ICROMA), Norrköping, Sweden, June 17th–20th, 2019, vol. 69. Linköping University Electronic Press; 2019. p. 390–406.
[24] Yang L, Gao Z, Li K. Passenger train scheduling on a single-track or partially double-track railway with stochastic information. Eng Optim 2010;42(11):1003–22.
[25] Castillo E, Gallego I, Ureña JM, Coronado JM. Timetabling optimization of a single railway track line with sensitivity analysis. Top 2009;17(2):256–87.
[26] Liu SQ, Kozan E. Scheduling trains as a blocking parallel-machine job shop scheduling problem. Comput Oper Res 2009;36(10):2840–52.
[27] Forsgren M, Aronsson M, Kreuger P, Dahlberg H. The maraca: a tool for minimizing resource conflicts in a non-periodic railway timetable. RailRome 2011; 2011.
[28] Lamorgese L, Mannino C, Natvig E. An exact micro–macro approach to cyclic and non-cyclic train timetabling. Omega (Westport) 2017;72:59–70.
[29] Blanco V, Conde E, Hinojosa Y, Puerto J. An optimization model for line planning and timetabling in automated urban metro subway networks. a case study. Omega (Westport) 2020;92:102165.
[30] Liu R, Li S, Yang L. Collaborative optimization for metro train scheduling and train connections combined with passenger flow control strategy. Omega (Westport) 2020;90:101990.
[31] Kang L, Zhu X, Sun H, Wu J, Gao Z, Hu B. Last train timetabling optimization and bus bridging service management in urban railway transit networks. Omega (Westport) 2019;84:31–44.
[32] Serafini P, Ukovich W. A mathematical model for periodic scheduling problems. SIAM J Discrete Math 1989;2(4):550–81.
[33] Caimi G, Fuchsberger M, Laumanns M, Schüpbach K. Periodic railway timetabling with event flexibility. Networks 2011;57(1):3–18.
[34] Liebchen C. The first optimized railway timetable in practice. Transp Sci 2008;42(4):420–35.
[35] Herrigel S, Laumanns M, Szabo J, Weidmann U. Periodic railway timetabling with sequential decomposition in the pesp model. J Rail Transp Plan. Manag 2018;8(3–4):167–83.
[36] Yan F, Bešinović N, Goverde RM. Multi-objective periodic railway timetabling on dense heterogeneous railway corridors. Transp Res Part B 2019;125:52–75.
[37] Liebchen C, Möhring RH. The modeling power of the periodic event scheduling problem: railway timetablesand beyond. In: Algorithmic methods for railway optimization. Springer; 2007. p. 3–40.
[38] Nachtigall K. A branch and cut approach for periodic network programming. Inst für Mathematik; 1994.
[39] Borndörfer R, Hoppmann H, Karbstein M, Lindner N. Separation of cycle inequalities in periodic timetabling. Discrete Optim 2020;35:100552.
[40] Lamorgese L, Mannino C. An exact decomposition approach for the real-time train dispatching problem. Oper Res 2015;63(1):48–64.
[41] Lamorgese L, Mannino C, Piacentini M. Optimal train dispatching by benders–like reformulation. Transp Sci 2016;50(3):910–25.
[42] Wolsey L.A.. Integer programming. series in discrete mathematics and optimization. 1998.
[43] Luan X, De Schutter B, Meng L, Corman F. Decomposition and distributed optimization of real-time traffic management for large-scale railway networks. Transp Res Part B 2020;141:72–97.