A multi-objective framework for strategic railway timetabling: integration of ant colony optimisation and mixed integer programming

Nicola Coviello\textsuperscript{a*}, Giorgio Medeossi\textsuperscript{b}, Thomas Nygreen\textsuperscript{c}, Paola Pellegrini\textsuperscript{d}, Joaquin Rodriguez\textsuperscript{e}

\textsuperscript{a} Trenolab SRLS, Via Maniacco 7/A, 34170 Gorizia, Italy. Email: n.coviello@trenolab.com. ORCID: https://orcid.org/0000-0002-7434-803X. * Corresponding author.

\textsuperscript{b} Trenolab SRLS, Via Maniacco 7/A, 34170 Gorizia, Italy. Email: g.medeossi@trenolab.com. ORCID: https://orcid.org/0000-0003-3042-9361

\textsuperscript{c} The Norwegian Railway Directorate, Postboks 16, Oslo, Norway, NO-0101. Email: thomas.nygreen@jernbanedirektoratet.no

\textsuperscript{d} COSYS-LEOST, Univ. Gustave Eiffel, IFSTTAR, Univ. Lille, F-59650 Villeneuve d’Ascq, France. Email: paola.pellegrini@univ-eiffel.fr. ORCID: https://orcid.org/0000-0002-6087-651X

\textsuperscript{e} COSYS-ESTAS, Univ. Gustave Eiffel, IFSTTAR, Univ. Lille, F-59650 Villeneuve d’Ascq, France. Email: joaquin.rodriguez@univ-eiffel.fr

Keywords
Train Timetabling Problem, ACO, MILP, Multi Objective

Abstract
This paper presents an algorithmic framework to perform automatic timetabling, developed within the project “Tools for mathematical optimisation of strategic railway timetable models” funded by the Norwegian Railway Directorate (Jernbanedirektoratet) and carried out by Trenolab SRLS and Gustave Eiffel University. The aim of this project is to develop a prototype tool to automatically generate timetable drafts. This will help planners to perform such tasks as capacity studies and strategic timetable planning. The paper describes the algorithmic core of the tool, called Automatic Timetabler with Multiple Objectives (ATMO). It implements a Pareto multi-objective approach, returning the user Pareto Optimal Set (POS) of timetables optimised according to up to 4 objectives. The framework is composed by a Multi-Objective Ant Colony Optimisation (MOACO) algorithm integrated with a Mixed Integer Linear Programming (MILP) formulation. MOACO performs a fast-but-coarse exploration of the solution space, populating and maintaining at the same time a POS of timetables. In MOACO, the TTP as a two-layer graph problem: on the one hand, a clique on a Layer 1 graph defines the set of trains scheduled; on the other hand, paths on Layer 2 graphs define the travel characteristics of these trains. The timetables generated by MOACO are then fed to a MILP formulation which performs a further refinement exploring only a neighbourhood of the input solution, finally providing feasible, high-quality timetables. The paper describes the framework from an algorithmic perspective, explaining how the proposed approach represents a novel contribution to the field of solution methods for the Train Timetabling Problem. Finally, a series of application tests are discussed, based on case studies driven from real practice in Norway.

1 Introduction

To maximise the benefit of passengers and rail freight customers, the Norwegian Railway Directorate (Jernbanedirektoratet) and other railway agencies rely heavily on strategic timetables to identify necessary new or upgraded infrastructure and rolling stock. When converting a set of conceptual railway service requirements into a feasible timetable, the planners must balance several conflicting objectives, such as travel time, capacity utilisation and robustness. The timetabling process is time- and resource-consuming, and it often must be carried out for a significant number of different concepts, studying multiple alternatives for each concept. Automationisation (at least partial) of this process will therefore significantly improve planners’ productivity in performing such
tasks as capacity studies and strategic timetable planning.

This paper presents an algorithmic framework to perform automatic timetabling, developed within the project “Tools for mathematical optimisation of strategic railway timetable models” funded by the Jernbanedirektoratet and carried out by Trenolab SRLS and Gustave Eiffel University. The aim of this project is to develop a prototype tool to automatically generate timetable drafts. The algorithmic core of the tool, called Automatic Timetable Maker with Multiple Objectives (ATMO) framework, rather than aggregating the different objective functions in a final objective function by means of weights, implements a multi-objective approach. It provides the user with a set of timetables representing an approximation of a Pareto-Optimal Set (POS) in the objective function (hyper-) space. The POS represents the best-found ways to exploit the available capacity, and is constituted by a set of non-dominated solutions, each solution being a timetable.

The Train Timetable Problem (TTP) is a classical problem in the field of Operations Research. It is notoriously NP-hard, meaning that for large instances exact methods likely fail to return the optimal solution in a reasonable time. No guarantee exists to find even high-quality feasible solutions quickly. Here, the size of practical instances is typically large. Furthermore, exact methods such as those based on an integer linear programming formulation tackled by commercial solvers cannot manage Paretian multi-objective optimisation within a single algorithm run. To this purpose, they need to be run repeatedly, involving significant time consumption.

Metaheuristics are algorithmic principles that can be instantiated to tackle virtually any optimisation problem. They have proven to be effective in tackling combinatorial NP-hard problems such as the TTP: they can provide rather good solutions within a reasonable computation time. Furthermore, they can be extended to perform multi-objective optimisation, and in particular to search for a POS of solutions.

In our approach, we exploit both the metaheuristic and exact perspectives, by combining a Multi Objective Ant Colony Optimisation (MOACO) algorithm and a Mixed Integer Linear Programming (MILP) formulation tackled by a commercial solver. We adopt Ant Colony Optimisation (ACO) for three main reasons. First, it suits well the resolution of the TTP since it builds solutions incrementally, thus mimicking real-practice timetables planning. This will foster the understanding and acceptance of the way the ATMO works by practitioners. Second, it features effective exploration capabilities thanks to the blended exploitation of memory from past iterations (pheromone trails) and local information. On the one hand, pheromone trails can capture the implicit bonds between different choices eliminating the need for modelling them explicitly. On the other hand, local information can be provided by well-known key performance indicators (KPIs) for timetables. Third, ants generate and maintain populations of solutions, which fits particularly well Paretian multi-objective optimisation. A consolidated literature background describes how to extend ACO to this purpose.

In this paper, we present a novel method for solving the TTP with a Paretian multi-objective approach. We will first introduce a novel MOACO algorithm for the TTP which provides a fast-but-coarse exploration of the solution space, populating and maintaining at the same time a Pareto Optimal Set of timetables with regard to five objective KPIs. It combines the state of the art on ACO applications for solving minimum cost clique and path problems. Indeed, we model the TTP as a two-layer graph problem: on the one hand, a clique on a Layer 1 graph defines the set of trains scheduled; on the other hand, paths on Layer 2 graphs define the travel characteristics of these trains. To the best of our knowledge, this is the first approach proposed in the literature combining the two problem solutions. The timetables generated by MOACO are then fed to a MILP formulation which performs a further refinement exploring only a neighbourhood of the input solution, finally providing feasible, high-quality timetables.

The paper is structured as follows: Section 2 reports a short analysis of the literature, focusing on the practically applicable approaches. Section 3 describes the adopted data model and the relevant formulation for the TTP. Section 4 introduces the algorithmic principles of ATMO, in particular Section 4.1 presents the MOACO algorithm for the TTP and Section 4.2 describes the integration with the MILP formulation, which is reported in Appendix B. Numerical experiments are presented and discussed in Section 5 and conclusions are drawn in Section 6, proposing some possible future
developments.

2 Literature review

The Train Timetabling Problem (Hansen and Pachl, 2014) consists in defining the arrival and departure times of trains in stations and in selecting their routing across the network. The Real-Time Train Dispatching Problem (RTTDP) can be seen as a variant of the TTP in which a new feasible timetable (both in routing and scheduling) has to be reconstructed after a disruption. The TTP and the RTTDP normally share the same core formulation (with some minor differences) and can be solved with similar methods.

Several approaches have been pursued to solve the TTP, as MILP methods or meta-heuristic techniques. Caccianì et al. (2016), Caimi et al. (2017) and Lusby et al. (2018) provide extensive and up-to-date reviews on this topic. We refer the interested reader to these reviews for a complete analysis of the available literature. We discuss here the few papers dealing with the actual deployability of this literature.

Indeed, many issues limit the actual applicability of existing methods to real case studies, characterised by big-sized problem instances. Some major issues are reported by Bešinović (2018), which describes how existing integrated timetabling approaches tend to lack in efficiency, stability, feasibility or robustness of solutions. In general, as stated by Lamorgese (2017) exact solution of TTP models is often impossible, and even finding feasible solutions in a reasonable time is a difficult task. These authors agree that the only promising pathway for effectively solving the TTP is to adopt decomposition techniques in combination with heuristics.

In a particularly interesting study, Jordi et al. (2019) experiment with different approaches proposed in literature to test their quality and performance. The aim of this activity is to find the best solution to implement an integrated capacity planning tool for the Swiss Railways. The study highlights how the capability to solve big-sized problem instance (e.g., the whole Swiss network) is one of the main problems encountered. The authors focus on MILP-based methods, and on the one hand they confirm that solving routing and scheduling in the same MILP instance is often unfeasible for practical size problems. On the other hand, they point out that a MILP model remains a benchmark for solving smaller instances exactly. The authors conclude that a feasible approach is to reduce the size of the instances, for example by means of (meta-)heuristics, before solving them exactly.

Focusing on large instances without completely abandoning the consideration of microscopic capacity constraints, Lamorgese et al. (2017) adopt a Benders-like decomposition in which the original TTP is split into a master and a slave problem. In principle, the master problem deals with train scheduling in line stretches connecting consecutive stations, while the slave one manages the station tracks allocation only. Once the master problem is solved to optimality, a set of additional constraints is generated for the slave one. If the latter is still feasible (and hence an optimal solution exists), then the master’s solution is feasible too and the combination of the two constitutes an optimal solution for the original TTP. Otherwise, a set of new constraints is generated for the master problem and the process is iterated. This approach has been successfully applied, as presented in Borndörf er et al. (2017).

With the same aim, Goverde et al. (2016) present a performance-based timetabling framework that decomposes the problem into three layers, namely a macroscopic, microscopic and fine tuning one. Each layer optimises a set of timetable KPIs, which altogether quantify timetable performance. The macroscopic layer produces the network routing of courses with aggregated capacity constraints macroscopic elements as nodes or lines. The microscopic layer generates feasible timetables within each macroscopic element, while the fine tuning one performs a final optimisation of trains’ speed profiles. By iterating across the layers, the framework actually produced optimised timetables with regard to the considered KPIs.

Even if these approaches consider different timetable KPIs as objectives (e.g., travel times, stability, energy consumptions, etc.) to be optimised during the timetabling process, they do not perform multi-objective optimisation in a strict sense. In fact, they aggregate the KPIs in a final objective function by means of weights. This approach presents a major known drawback, being that the
produced timetables significantly dependent on the weights’ values.

The method proposed by Yan et al. (2019) actually performs Pareto multi-objective optimisation, returning Pareto Optimal Sets of timetables with regard to a set of objective KPIs. It exploits a MILP formulation integrated with an \(\epsilon\)-constraint method to construct Pareto fronts representing the optimal trade-off between four different timetable KPIs, further developing the approaches proposed by Burdett (2015) and Binder et al. (2017). These studies point out how the utilisation of Pareto fronts can effectively depict all the possible ways in which railway capacity can be exploited to optimality. This knowledge is of interest in case of performing strategic timetabling, where is of main interest for the planner to be able to assess different timetable alternatives.

In our work, we overcome the computational limitation pointed out for many academic approaches, by exploiting the power of meta-heuristics to quickly provide good solutions to big size problems. Moreover, we supply the user with a POS of solutions, concurrently considering up to four objective functions. Finally, we show the applicability of our framework on real-world case studies provided by Jernbanedirektoratet.

3 Train Timetabling Problem

In this work, we model the TTP as the problem of transforming a service concept into a feasible timetable. A service concept is a set of conceptual railway service requirements. It is basically a raw timetable draft which provides the number and type of trains that have to figure in the output timetable, respecting given technical and operational constraints. The former make the output timetables feasible, i.e., ensure that in real operation trains can respect their schedule, at least in absence of traffic perturbations (Goverde and Hansen, 2013). The latter foster the compliance with commercial or organisational needs (passenger connections and transfer times, crew and rolling stock rostering, etc.).

The problem we deal with consists in finding a number of feasible timetables that respect a service concept and optimise a set of objective functions. In particular, we are interested in minimizing total weighted travel time and energy consumption, and maximizing timetable stability and weighted number of schedules trains.

3.1 Infrastructure model

We adopt a macroscopic model for infrastructure and operations, based on a multigraph \(G = (V, E)\). Nodes represent timing locations (stations, junctions, halts) in the rail network. For each node \(l \in V\) we define:

- A set of tracks \(t \in T_l\), one of which is identified as the main one;
- An additional running time \(\Delta r_{rs}^t\) if a track \(t\) different from the main one is used by a rolling stock \(rs\) (0 in case of using the main track);
- For each track, a minimum separation time between the end of the occupation by a train and the beginning of the occupation by the following one \(t\) providing that each track can be used by at most one train at a time.

Edges are line stretches connecting consecutive timing locations. Edges can be mono- or bi-directional, and more than one edge can connect the same pair of consecutive stations. Any edge \(e \in E\) is represented by \(e = (i, j, e)\), where the origin and destination of \(e\) are in the set of nodes \((i, j) \in E\) and \(j\) is an index in \([1, |e|]\) to distinguish different edges connecting the same pair of nodes (for example to represent the presence of multiple-track line). A set of rolling stock types \(rs \in RS\) can travel on a line stretch. A rolling stock can cross each location in two different modes \(y \in \{\text{PASS}, \text{STOP}\}\) (passing or stopping). Therefore, four types of events can characterise the traveling along a line stretch, depending on the so-called edge extremity events (EEE). These can be pass/pass, pass/stop, stop/pass, stop/stop: \(e \in EEE\) is a pair indicating what event occurs at the origin and at the destination node, respectively.

On an edge \(e \in E\) we have:

- A technical minimum run time \(mrt_e^{rs, dir, e}\) depending on the rolling stock of the train (\(rs \in RS\) and \(dir \in \{\text{PASS}, \text{STOP}\}\)) and the direction of travel;
Each group have smaller step than a freight one, although exceptions may exist. For example, a passenger train group

Moreover, a time discretisation step indicates the need of high precision associated to a train group. For example, a passenger train group will typically have smaller step than a freight one, although exceptions may exist.

In our model, trains can be scheduled with running times greater than the minimum technical ones. We adopt a linear relationship to model the relevant variation of energy consumptions and headways between trains running in the same direction. If a run time \( rt > mrt_{e,dir}^{rs} \) is chosen for a given rolling stock \( rs \in RS \), the energy consumption is reduced by \( kec_{e,dir}^{rs}(rt - mrt_{e,dir}^{rs}) \), i.e.,

\[
ec = mec_{e,dir}^{rs} - kec_{e,dir}^{rs}(rt - mrt_{e,dir}^{rs})
\] (1)

For a pair of trains \( t_1 \) and \( t_2 \) (using \( rs_1 \) and \( rs_2 \) respectively), if the running time of the first train is higher than the minimum one, the minimum headway is increased by \( kh_{e,dir}^{rs_1}(rt_1 - mrt_{e,dir}^{rs_1}) \), i.e.,

\[
h_{t_1,t_2} = mh_{e,dir}^{rs_1,rs_2} + kh_{e,dir}^{rs_1}(rt_1 - mrt_{e,dir}^{rs_1}) \] (2)

To model these relationships as linear ones is a strong assumption: it is well-known that, in reality, they feature non-linear trends. However, the linearisation allows for a much easier integration into the MILP formulation. Empirical experience gathered by the authors permits to assume that, at least in the considered case studies, the distortions introduced by the linearisation are acceptable as long as the run times upper bound is lower than or equal to than 1.15 times the minimum technical run time defined for each line section.

### 3.2 Service concept model

The service concept model describes the train services that shall be scheduled in the timetables. It contains all relevant data concerning, e.g., timing locations and line stretches to be traversed, possible timing, stop pattern. This information is provided for train groups, i.e., for sets of periodic trains. Spare courses are modelled as single-train groups. Three types of groups can be defined: fixed groups represent a mere constraint to the timetabling process as they are to be set exactly as described; movable groups can be adjusted in time and space (station routing) to optimise the timetables; optional groups are movable groups that can also be excluded from the timetable. Train groups can have different priorities, to be associated to their contribution to the objective functions. Moreover, a time discretisation step is defined for each group. The relevant times of a train group in a timetable are set to multiples of this discretisation step. A small discretisation step indicates the need of high precision associated to a train group. For example, a passenger train group will typically have smaller step than a freight one, although exceptions may exist.

Each group \( g \) is characterised by:

- The type \( type_g \in \{FIXED, MOVABLE, OPTIONAL\} \);
- The period \( p_g \);
- The number of trains belonging to the group \( nt_g \);
- The priority factor \( pr_g \geq 1 \);
- The sets \( V_g \) and \( E_g \) of nodes and edges respectively used by the group’s trains. We define with \( L_g \) and \( I_g \) the first and last nodes in \( V_g \) respectively. Arrival and departure events are
These objectives are computed by means of specific KPIs:

- **ATMO timetables**
- **The time discretisation** \( \varphi_g \).

For each node \( l \in V_g \), the following constraints are defined in the service concept:

- The minimum and maximum arrival times at \( l \), \( \overline{\text{arr}_{g,l}} \) and \( \underline{\text{arr}}_{g,l} \) respectively. They are defined, for the first train of the group, only if \( l \neq \underline{l} \);
- The minimum and maximum departure times from \( l \), \( \overline{\text{dep}}_{g,l} \) and \( \underline{\text{dep}}_{g,l} \) respectively. They are defined, for the first train of the group, only if \( l \neq \overline{l} \);
- The rolling stock used by the group’s trains on the edge leaving \( l \), only if \( l \neq \overline{l} \);
- The set \( T_{g,l} \) of usable station tracks;
- The set \( \Gamma_{g,l} \) of usable pass/stop modes;
- The minimum and maximum stop times \( \text{stop}_{g,l} \) and \( \overline{\text{stop}}_{g,l} \). Note that if there is no mandatory stop at \( l \), we will have \( \overline{\text{stop}}_{g,l} = 0 \) and \( \overline{\text{stop}}_{g,l} \) a large constant. If stops are forbidden, then \( \overline{\text{stop}}_{g,l} = 0 \);
- The minimum and maximum run times admitted on the edge \( e \) leaving \( l \) (only if \( l \neq \overline{l} \)), \( \overline{\text{rt}}_{g,e} \) and \( \underline{\text{rt}}_{g,e} \);
- The **periodicity tolerance** \( \text{tol}_{g,l} \geq 0 \), defined as the maximum deviation from a strictly periodic pattern, in absolute value. A tolerance equal to 0 imposes that arrivals and departures of pairs of trains of the group are separated by exactly a multiple of the group’s period \( p_g \) in location \( l \). Remark that the same periodicity tolerance is applied to all pairs of trains in the group in the same node but there may be different periodicity tolerances for each node in the group’s journey.

### 4 The ATMO solution framework

In this section, we detail the Multi-Objective Automatic Timetable Generator framework that we propose in this paper, as well as its algorithmic components. ATMO uses a multi-objective approach to find Pareto Optimal Set approximations for TTP instance. It exploits a novel MOACO algorithm and the solution of a MILP formulation.

The former quickly performs a wide-ranging exploration of the solution space and returns a set of timetables. The latter refines these timetables to improve their quality.

ATMO assesses the quality of timetables according to the following four objectives:

- **TTT**: Minimise total weighted travel time.
- **EC**: Minimise total weighted energy consumption.
- **NTR**: Maximise the weighted number of scheduled trains.
- **ST**: Maximise timetable stability.

These objectives are computed by means of specific KPIs:

1. A train’s travel time is the difference between the arrival time at the last station and the departure time from the first one. The total weighted travel time is the sum of the trains’ travel time multiplied by the respective group priority factor.
2. Energy consumption strictly depends on the running times of trains on each edge, as well as on whether trains perform passes or stops in stations where both options are allowed. All energy consumption values are input data. The total weighted energy consumption is the sum of the trains’ energy consumption multiplied by the respective group priority factors.
3. The total weighted number of scheduled trains is the product of the number of trains of each optional group included in the timetable and its priority factor.
4. Stability is measured as the minimum buffer time in the timetable. Buffer time is the time separation between two consecutive utilisation of the same infrastructure resource (line
stretch or station track) minus the minimum admissible headway between the corresponding train movements. For future extensions, different stability KPIs could be envisaged, provided that they can incrementally computed during ants’ exploration.

These KPIs can be computed incrementally by ants at each solution construction step, as it is described by the following sub-sections.

### 4.1 MOACO algorithm

The MOACO algorithm is in charge of finding a set of timetables optimising the multiple objectives of the TTP.

Ant Colony Optimisation algorithms, first proposed by Dorigo (1992), are based on the iterative construction of feasible solutions by a colony of artificial ants. The construction of a feasible solution consists in the sequential selection of a set of nodes on a pre-defined construction graph. Ants construct solutions progressively, stochastically selecting the node to add at each step through the so-called pseudo-random proportional rule. Here, the selection is biased by a greedy measure of the quality of the node (heuristic information \( \eta \)) and by a quantity representing the cumulated knowledge of the colony on the actual quality (pheromone trail \( \tau \)). In particular, being in node \( u \), an ant selects the next node to add from a candidate set \( \text{CAND} \): each node \( v \in \text{CAND} \) has probability:

\[
p_v = \frac{\tau_{(u,v)}^\alpha \eta_{(u,v)}^\beta}{\sum_{v \in \text{CAND}} \tau_{(u,v)}^\alpha \eta_{(u,v)}^\beta}
\]

with \( \alpha \) and \( \beta \) parameters of the algorithm. Pheromone is updated at each iteration to progressively increase the trail on the components of the best solutions. The process stops when a termination condition is met (depending on a maximum number of iterations or a maximum elapsed time).

Several ACO variants exist in the literature, mostly including variations to the application of pseudo-random proportional rule and pheromone update. MOACO is a multi-objective, two-layer extension of the Max-Min Ant System variant (Stützle and Hoos, 1997, 2000, López-Ibáñez and Stützle, 2012). In the Max-Min Ant System, the pseudo-random proportional rule of Eq. (3) is applied. At each iteration, pheromone evaporates from all solution components and is deposited on those of solution \( S \), which is either the best so far or the iteration best solution:

\[
\tau_{(u,v)} = \rho \tau_{(u,v)} + \delta(S)
\]

Here, \( \rho \) is a parameter of the algorithm included between zero and one, and \( \delta(S) \) is a measure of the quality of \( S \). To avoid stagnation, pheromone is always bounded to be in the interval \( [\tau_{\text{min}}, \tau_{\text{MAX}}] \).

#### 4.1.1 Multi-objective extension

In MOACO, to deal with the various objective functions of the TTP, we follow the multi-colony architecture proposed by López-Ibáñez and Stützle (2012). Specifically, several colonies of artificial ants build solutions concurrently, each considering its own pheromone trail and objective function. Throughout iterations, the POS is progressively filled and updated.

We have five colonies: each colony \( C_o \) optimises objective function \( o \in O \), with \( O = \{TTT, EC, NTR, ST, CFL\} \). These are the four TTP objective functions, to which we add the number of residual conflicts (\( CFL \)) to minimise. A conflict occurs when a pair of trains utilise the same infrastructure resource concurrently or anyway without respecting the minimum headway between the corresponding train movements. By adding this objective, we set technical constraints as soft ones. Hence, MOACO does not necessarily returns feasible solutions. However, preliminary analysis have shown that this choice is definitely benefitting in terms of final solution quality, especially as MOACO solutions are refined through the MILP formulation, which restores feasibility.

Both heuristic information and pheromone trail are defined for each objective. The former are common to all colonies (\( \eta_{\varnothing}, \forall \varnothing \in O \)), the latter evolve independently in each of them (\( \tau_{\varnothing, C_o}, \forall \varnothing, o \in O \).
For the TTP, it describes arrival/departure times, pass/stop event and used track in each station. A set of Layer 2 sub-layers has been used in the literature for tackling the Multiple Depot Vehicle Routing Problem (Yao et al., 2014) and the Course Timetabling Problem (Nothéger et al., 2012). For the TTP, it actually mimics the real-world timetabling procedure performed mainly by hand by

\[ O \]. This means that each colony \( C_\delta \) (with \( \delta \in O \)), features its own pheromone trails \( \tau_{\delta,\epsilon} \) for each objective \( \epsilon \in O \).

The values associated to different objectives are normalised and they are all set to be included in the interval \([\tau_{\text{min}}, \tau_{\text{MAX}}]\), defined within the same bounds set to the pheromone trails. Normalisation permits to homogenise the different KPIs regardless of their actual dimension. In the normalisation, the computation is made so that the heuristic information decreases as the quality of the corresponding edge worsens according to the considered objective function. Depending on the need for minimisation or maximisation of the objective function, the worsening is either the increase of the decrease of the value used to quantify this quality. The weighted sum of heuristic information and pheromone trails associated to all objectives constitutes the \( \tau \) and \( \eta \) values to be used in the pseudo-random proportional rule. The weights used here vary across iterations and colonies. For colony \( C_\delta \), the weight of pheromone trail and heuristic information associated to objective \( \epsilon \) (\( \lambda_\epsilon^i \) at iteration \( i \)) is initially set to one and decreases by 0.1 at each iteration until it reaches the value 0.5. At this point, it is set back to one and the progressive decrease starts again. The weights of all other objectives (\( \lambda_\epsilon^i \forall \epsilon \in O \setminus \{\delta\} \) at iteration \( i \)) are randomly selected at each iteration with the only constraints that they are non-negative and the sum of all weights must sum to one. Each colony manages its own pheromone trains and builds solutions independently.

After each iteration, the POS is updated by adding all solutions that weakly dominate the already included ones and removing all strictly dominated solutions. Then, an update-by-region strategy drives pheromone updates: the POS is first split into equal size regions, i.e., equal cardinality subsets, one for each colony. Region \( R_\epsilon \) for colony \( C_\epsilon \) includes the \(|\text{POS}|/|O|\) best solutions according to objective \( \epsilon \). Then, the best \( N_\text{upd} \) (with \( N_\text{upd} \geq 1 \) parameter of the algorithm) solutions of each \( R_\epsilon \) are used to update pheromone trails of the corresponding \( C_\epsilon \) following Eq. (4). To promote solutions which minimise the number of residual conflicts disregarding the objective function considered by each colony, we consider the best solutions of region \( R_\epsilon \) as those minimizing CFL, first, and optimizing \( \epsilon \), second. Let \( S_\epsilon \) be the best solution of \( R_\epsilon \) according to such lexicographic ordering.

\[
\delta(S) = \frac{1}{1 + (\text{CFL}(S) - \text{CFL}(S_\epsilon))} \cdot \max \left( 1, \frac{1}{1 + k_\epsilon (\epsilon(S) - \epsilon(S_\epsilon))} \right)
\]

\( \delta(S) \) is the amount of pheromone deposited on the elements of solution \( S \), and it is given by the product of two terms, according to Eq. (5). The first one is the inverse of the difference of the number of residual conflicts in \( S \) and \( S_\epsilon \), augmented of one unit so as to avoid null denominators for \( S_\epsilon \) itself. The second one is the inverse of the worsening of \( S \) w.r.t. \( S_\epsilon \) in terms of objective function \( \epsilon \). If \( \epsilon \) is to be minimised, the difference is the subtraction of the value of the objective function of \( S_\epsilon \) from the one of \( S \) (\( k_\epsilon = 1 \)), and the opposite holds in case of maximisation (\( k_\epsilon = -1 \)). As for the first term, this value is augmented of one unit to avoid null denominators. Remark that, due to the lexicographic ordering of \( R_\epsilon \), it is possible that the subtraction is negative. To preserve the principle of depositing pheromone on the elements of good solutions, and hence having a strictly positive \( \delta(S) \), we consider the maximum between the second term and value one in the multiplication by the first term.

4.1.2 Two-layer extension

In MOACO, in any colony, a solution is the combination of the following sub-solutions (SS):

- A Layer 1 sub-solution \( S_1 \): defining:
  - The train groups actually scheduled;
  - The order in which these train groups are scheduled.
- A set of Layer 2 sub-solutions \( S_2 \), one for each scheduled train group \( g \). Each of them describes arrival/departure times, pass/stop event and used track in each station.

Similar two-layer structures have been used in the literature for tackling the Multi-Depot Vehicle Routing Problem (Yao et al., 2014) and the Course Timetabling Problem (Nothéger et al., 2012).
specialised planners.

4.1.2.1 Layer 1: train group selection

For finding a sub-solution in Layer 1, ants explore a directed graph $G_{L1} = (N_{L1}, E_{L1})$ whose set of nodes $N_{L1} = \{b_{L1}\} \cup \{C_{L1}\} \cup \{D_{L1}\}$ includes a source $b_{L1}$, a set of schedule nodes $c_{L1}^D \in C_{L1}$, and a set of discard nodes $d_{L1}^D \in D_{L1}$. Schedule node $c_{L1}^D$ is to be used if movable or optional train group $g$ is actually scheduled. Discard node $d_{L1}^D$ is to be used if optional group $g$ is not scheduled. Edges in $E_{L1}$ are defined between all pairs of nodes belonging to different train groups, and between the source and all other nodes.

A sub-solution $S1$ is a clique of cardinality $|C_{L1}|$ on $c_{L1}^D \in G_{L1}$, i.e., a subset of nodes including exactly one schedule or discard node per group of trains. Ants construct cliques progressively, starting from the source and by adding at each step the decision of whether to schedule a train group.

At each step, the candidate nodes to add are those corresponding to groups not considered yet and the choice is made according to the pseudo-random proportional rule. As explained in Section 4.1.1, heuristic information and pheromone trail are associated to specific objective functions. As for the heuristic information, for all edges terminating in a schedule node, for the NTR objective we set it equal to the number of trains belonging to the respective group multiplied by the group priority. For all other objectives, we use the overlap index. Specifically, we estimate how much the respective group would be in conflict with already scheduled ones (Appendix A). In our algorithm, the overlap index acts as a proxy for the TTT, EC, ST and CFL objectives. Indeed, a train with high overlap index will likely have to be scheduled considering extended running or dwell times to avoid conflicts, for example to allow for crossing trains at stations. Hence, travel time will increase. In parallel, additional braking which may come with conflict avoidance, are likely to increase also energy consumption. Moreover, trains avoiding conflicts will in general be scheduled as soon as feasible, penalizing stability. Finally, the link between overlap index and conflicts is straightforward. Whatever the objective among these four, the greedy assessment of the quality of a node selection decreases as a function of the overlap index. Thus, the normalisation described in Section 4.1.1 is done so that the highest overlap index corresponds to the worst quality, and hence to the minimum contribution to $\eta$. For edges terminating in discard nodes, we compute the heuristic information knowing that they have the highest quality in terms of overlap index and the lowest in terms of number of trains added, as they are both equal to zero.

When evaluating the probability of selecting a node $v$ to be added to a partial solution, the heuristic information values of all edges connecting nodes already belonging to the partial solution to $v$ are added up. The same holds for pheromone trails (Solnon and Bridge, 2006). As discussed in Section 4.1.1, all values are normalised and weighted depending on iteration and colony.

4.1.2.2 Layer 2: schedule definition

After every schedule node selection in Layer 1, the corresponding Layer 2 sub-solution construction is started. This sub-solution corresponds to a path on a Time Expanded (directed) Graph (TEG). Inspired by a classic timetabling model (Caprara et al., 2002), TEG’s nodes represent discrete timetable events for the first train of the group: the arrival (or departure) at (from) a certain station, at a certain time, at a certain station track, with or without a stop. The existing nodes depend on the group time discretisation step, the usable station tracks and pass/stop modes, and the bounds for minimum and maximum times as defined in Section 3.2. Moreover, two dummy nodes are added to represent the begin and end of the path. Edges represent train stops or passes at stations, or travels on line stretches. Their associated running time and energy consumption are constrained by the characteristics of the infrastructure (see Section 3.1), of those of the train group rolling stock and by the pass/stop mode used. At construction, we prune the TEG by performing an exhaustive backtracking from the dummy end node. All nodes that cannot be reached are immediately discarded.
Figure 1 provides a graphical example of the TEG relevant to a group travelling through three stations. The dummy *begin* and *end* nodes are highlighted, as well as the arrival and departure nodes in each station. While in Stations 1 and 3 just one track can be used, in Station 2 two tracks are available. In all three stations only the *STOP* station mode can be used, and in Station 2 the minimum stop time is greater than zero.

![Time Expanded Graph](image)

**Figure 1.** Graphical example of a Time Expanded Graph.

Each edge $e(u,v)$ of the TEG is characterised by the following set of heuristic information quantities, one for each objective $o \in O \setminus \{NTR\}$:

- $\eta_{(u,v),TTT}$ is the duration of the associated to the edge, multiplied by the number of trains $ntr_g$ of group $g$.
- $\eta_{(u,v),EC}$ is the energy consumed by the group’s trains using the edge, multiplied by the number of trains $ntr_g$ of $g$. If nodes $u, v$ refer to arrival and departure at the same station, $\eta_{(u,v),EC}$ is equal to zero;
- $\eta_{(u,v),CFL}$ is the number of conflicts with already scheduled train groups generated by using the edge.
- $\eta_{(u,v),ST}$ is the minimum buffer time with already scheduled train groups emerging by using the edge.

$\eta_{(u,v),TTT}$ and $\eta_{(u,v),EC}$ are computed at the TEG’s construction, as they do not depend on the current TTP partial solution. Instead, $\eta_{(u,v),CFL}$ and $\eta_{(u,v),ST}$ are dynamically computed each time a TEG is to be explored by an ant, as they must take into account all already scheduled train groups. These quantities are assessed considering all trains of the scheduled and to be scheduled groups. Indeed, we assume that trains of the same group are strictly periodic: the timetable of the trains following the first one is obtained by shifting the timetable of the latter of multiples of the group periodicity. Hence, a path on the TEG completely describes the timetable of the first train of each group.

We consider two alternative path construction options for building Layer 2 sub-solutions:

1. **Option 1:**
   - **Path Construction:** Direct path from *begin* to *end*.
   - **Heuristic Calculation:** $\eta_{(u,v),TTT}$ and $\eta_{(u,v),EC}$ are computed at the TEG’s construction. $\eta_{(u,v),CFL}$ and $\eta_{(u,v),ST}$ are computed dynamically each time a TEG is to be explored.

2. **Option 2:**
   - **Path Construction:** Multi-path from *begin* to *end*.
   - **Heuristic Calculation:** $\eta_{(u,v),TTT}$ and $\eta_{(u,v),EC}$ are computed at the TEG’s construction. $\eta_{(u,v),CFL}$ and $\eta_{(u,v),ST}$ are computed dynamically each time a TEG is to be explored.
1. **FAST exploration**: nodes are progressively added by applying the pseudo-random proportional rule considering only candidates that do not generate any conflict, if any. If no such candidate exists, then all neighbours of the current node are considered. In the pseudo-random proportional rule, the value of the heuristic information is simply the weighted sum of the quantities described above after normalisation, considering weights \( \lambda^2_S \) as described in Section 4.1.1.

2. **SMART exploration**: only nodes belonging to a low-conflict subset can be selected. To do so, we first apply a variation of Dijkstra algorithm to label the TEG’s nodes. Specifically, we define a conflict distance (cflDist) and an aggregated normalised distance (aggNormDist) labels. For each node, the former is the cost of the minimum cost path reaching it from the begin node. Here, the cost is the sum of \( \eta_{(u,v)},CFL \) for all edges of this path. If cflDist is equal to 0 for the end node, there exists at least one path in the TEG that does not introduce any additional conflict. Once computed this label for the whole graph, we identify all nodes with associated cflDist smaller than or equal to the one of the end node and we remove all the others. By doing so, we obtain the restricted TEG, which will be the graph explored by ants in this Layer 2 sub-solution construction step.

For all nodes of the restricted TEG, we compute the aggNormDist label as the cost of the minimum cost path, considering now the sum of the weighted sum of the normalised heuristic information quantities \( \eta_{(u,v)},TTE, \eta_{(u,v)},EC, \eta_{(u,v)},CFL \) and \( \eta_{(u,v)},ST \) for each used edge. The aggNormDist labels are then used to quantify the heuristic information on each edge, to be used in the pseudo-random proportional rule: \( \eta_{(u,v)} \) is equal to the ratio between aggNormDist of node \( v \) and the number of edges separating it from the begin node along the minimum cost path. We compute this ratio to keep the value of the heuristic information in the interval \([\tau_{\text{min}}, \tau_{\text{MAX}}]\), hence preserving its comparability with the pheromone trail.

In MOACO, we define two parameters \( n_{\text{FAST}} \) and \( n_{\text{SMART}} \) which define the number of consecutive iterations for which we use the FAST and the SMART exploration, respectively, starting with the latter. At each iteration, then, all ants of all colonies explore all TEG’s considering the same option. By setting these two parameters different from zero, we can balance the merits of the two options: the former is much quicker, while the latter is likely to find better solutions in terms of number of residual conflicts.

### 4.1.3 Local search

After building a TTP solution \( S \), composed of a Layer 1 sub-solution (\( S_1 \)) and several Layer 2 ones (\( S_2 \), one for each scheduled train group \( g \)), we apply a local search to try to improve by moving to the best solution of its neighbourhood. The main target aimed here is the reduction of the number of residual conflicts.

The neighbourhood we consider includes solutions which schedule exactly the same train groups as \( S_1 \) but with some different \( S_2 \) sub-solutions. In particular, we consider a subset of nodes of \( S_1 \) of cardinality \([|S_1 \cap C_{g}| : pcSol_{LS}]\), with \( pcSol_{LS} \in [0,1] \) parameter of the algorithm. The nodes in the subset are the first schedule nodes according to the selection order of the ant that built \( S_1 \). For each train group corresponding to these nodes, we modify the TEG solution by selecting the path with minimum cost in terms of total value of aggNormDist labels, as defined in Section 4.1.2 for the SMART exploration. Here, all scheduled train groups are considered for the conflict assessment, regardless of their position in the \( S_1 \) sub-solution.

The choice of the TEG’s to be re-explored is motivated by the observation that the first selected train groups in \( S_1 \) are scheduled without considering possible conflicts with groups selected later. With the local search we try to see if the schedules chosen for the former can be modified to better fit the ones chosen for the latter.

We apply this local search for \( n_{LS} \) consecutive iterations, followed by \( n_{noLS} \) iterations in which we do not explore any ant solutions’ neighbourhood. Recall that an iteration includes the activity of all colonies, with several ants building solutions within each of them.
4.2 MILP formulation

Solutions $S$ generated by MOACO are then refined through the solution of a MILP formulation. It takes as input the train groups to be scheduled, and passing and stopping times at all locations visited in their journey. By exploiting the periodicity tolerance of trains belonging to the same group and dropping temporal discretisation, the formulation slightly modifies these times and seeks for a conflict-free solution which optimises the weighted sum of the normalised values in $[\tau_{\text{min}}, \tau_{\text{MAX}}]$ of three objectives: TTT, EC, ST. Indeed, the number of trains is constant, as all train groups scheduled in $S$ are to be scheduled here. Moreover, conflicts are modelled by hard constraints, making their minimisation meaningless. The weights considered in the formulation objective function are those used by the ant that built solution $S$.

The MILP formulation is modelled after those proposed by Mannino et al. (2015) and Pellegrini et al. (2015). It is reported in Appendix B. It considers the infrastructure and service concept models described in Section 3. Binary variables control the use of specific tracks at stations and of specific EEs, as well as the precedence between pairs of trains using the same resource. Continuous variables control arrival and departure times, as well as running times and energy consumption. Arrival and departure time variables are bounded to be chosen within thin intervals around the ones fixed in $S$. The maximum allowed time modification is a parameter that we call MILP degree of freedom ($DOF_{\text{MILP}}$). By varying the value of continuous variables, the formulation can:

- Shift a whole periodic train group by a maximum value $DOF_{\text{MILP}}$;
- Modify the schedule of individual trains provided that its group’s periodic pattern is relaxed within the tolerance permitted in the service concept.

The restricted variable domains allow (in principle) for a fast solution. This process returns either the optimal solution or an infeasibility due to the modelling of conflicts as hard constraints.

Figure 2. Graphical representation of the MILP formulation operation in the objectives space.

Figure 2 provides a graphical representation of the operation of the MILP formulation stage in the objective function space. Two objectives are considered here. MOACO solutions are blue points. Some are conflict-free, other not (the orange-circled ones). Dark blue points represent the POS provided by MOACO. Green points are MILP solutions obtained starting from MOACO ones. Not all MOACO solutions with residual conflict can be converted into conflict-free timetables, which is shown as arrows terminating in red crosses. A new POS is originated by the MILP solutions. Light green points represent MILP solutions that belong to this new POS, while dark green points are removed as they are dominated.
4.3 ATMO framework

Figure 3 displays the architecture of the ATMO framework. Grey ovals represent data contents, blue rectangles processes and orange diamonds conditional switches. Solid arrows mark the operations flows, while dashed ones are data feedings. After reading the input data, MOACO is started. It iteratively creates and updates a provisional POS as explained in Section 4.1. After the stopping criterion is satisfied by MOACO, i.e., after a given number of iterations have been completed or the available computational time has elapsed, the MILP formulation refines all POS solutions.

This framework has several parameters. Table 1 provides a summary of the MOACO parameters. In addition, ATMO requires setting the MILP degree of freedom (DOF\textsubscript{MILP}) and the stopping criteria for MOACO and MILP solution.

Table 1. Summary of the MOACO algorithm’s parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nant</td>
<td>Number of ants per colony</td>
</tr>
<tr>
<td>maxIter</td>
<td>Maximum number of iterations</td>
</tr>
<tr>
<td>maxTime</td>
<td>Maximum computation time</td>
</tr>
<tr>
<td>$\alpha_{L1}$</td>
<td>Pheromone information weight in L1</td>
</tr>
<tr>
<td>$\beta_{L1}$</td>
<td>Heuristic information weight in L1</td>
</tr>
<tr>
<td>$\rho_{L1}$</td>
<td>Pheromone evaporation factor in L1</td>
</tr>
<tr>
<td>$\tau_{MIN,L1}$</td>
<td>Lower pheromone bound in L1</td>
</tr>
<tr>
<td>$\tau_{MAX,L1}$</td>
<td>Upper pheromone bound in L1</td>
</tr>
<tr>
<td>N\textsuperscript{weights}</td>
<td>N\textsuperscript{o} of main objective aggregation weights per colony</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Min. value of the aggregation weight of the colony’s main objective</td>
</tr>
<tr>
<td>$n_{SMART}$</td>
<td>Smart mode usage period</td>
</tr>
<tr>
<td>$n_{FAST}$</td>
<td>Fast mode usage period</td>
</tr>
<tr>
<td>$n_{upd}$</td>
<td>Number of updating solutions per region</td>
</tr>
<tr>
<td>pcSol\textsubscript{LS}</td>
<td>Percentage of solution components refined by Local Search</td>
</tr>
<tr>
<td>$n_{LS}$</td>
<td>Local search usage period</td>
</tr>
<tr>
<td>$n_{noLS}$</td>
<td>Local search not-usage period</td>
</tr>
</tbody>
</table>
5 Applications and results

In this section, we report the details of the experimental analysis carried out to assess the ATMO framework. We first present the test instances we consider. Then we describe the ATMO assessment. Finally, we showcase a practical example of ATMO deployment.

5.1 Test instances

Tuning and tests are performed on 54 instances drawn from real timetabling practice in Norway. Instances are built on two parts of the Norwegian national network, namely the Bergen line (Bergensbanen, BB) and the region around Trondheim (Trønderbanen, TRB). These areas form a good test-bench for our framework under working conditions. Both are realistic instances of regional strategic timetable projects, as well as realistic medium-sized assignments to individual timetable planners when partitioning the national network as part of a network-wide project. While this partitioning helps overcome the complexity of the TTP, it easily leads to a sub-optimisation of the network-wide timetable, for human planners and ATMO alike.

Table 2. Main characteristics of the test infrastructure.

<table>
<thead>
<tr>
<th>Infrastructure</th>
<th>BB</th>
<th>TRB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>370</td>
<td>200</td>
</tr>
<tr>
<td>Signalling system</td>
<td>Conventional Norwegian signalling, based on axel-counters</td>
<td></td>
</tr>
<tr>
<td>Number of timing locations</td>
<td>53</td>
<td>52</td>
</tr>
<tr>
<td>Of which stations</td>
<td>32</td>
<td>30</td>
</tr>
<tr>
<td>Average n° of available sidings in stations</td>
<td>2.3 (excluding the 8-tracks Bergen station)</td>
<td>2.4 (excluding the 13-tracks Trondheim station)</td>
</tr>
</tbody>
</table>

Table 2 provides a concise overview of the main infrastructure characteristics of these lines, while Figure 4 frames them within the whole Norwegian railway network.

![Figure 4. Geographical framing of the considered lines. Major stations only are displayed.](image)

On each infrastructure, different instances are built using a reference daily traffic pattern, whose main characteristics are described in Figure 5 and Table 3. For each considered train group we report:
the number of trains in the two directions; the periodicity, for trains in the same group; the number of stations in which trains can stop and use a passing loop; the number of such stations in which a stop is mandatory, the remaining ones being optional stops, where a passing is normally preferred. Groups are named after their category: FR for freight; LH for long-haul passenger; R for regional. All train groups are movable and none is optional.

From the base traffic patterns, we derive 24 and 30 test instances for the TRB and the BB line, respectively. These are produced by combining:

- Four different sizes of the admissible time windows of train group timings in stations (the difference between \(\text{arr}\) and \(\text{arr}\) (as well as \(\text{dep}\) and \(\text{dep}\)) in each location of a train’s journey): 15, 30, 60 and 90 minutes (for the TRB line only, also 120 minutes);
- Six different active train configurations, in which 0 to 5 train groups (randomly chosen in the traffic patterns) are deactivated and removed from the scheduling process.

The input datasets of the instances were prepared using the Treno suite of railway timetable planning software (see Medeossi and Nash, 2020) currently used by Jernbanedirektoratet. A prototype dedicated interface provides seamless data transfer between Treno and the ATMO framework and vice-versa.

All the 54 resulting instances are tested activating the objectives \(TTT\) and \(ST\), in addition to the default \(CFL\). As no train group is optional, the maximisation of the number of scheduled trains is not meaningful here. As for energy consumption, we excluded it because input data are not currently available in the necessary level of detail.

### 5.2 ATMO configuration

In this section, we assess different configurations of ATMO, aiming to understand the contribution of different algorithmic components to the framework performance. As a first step, we focus on MOACO alone, i.e., we do not activate any MILP refinement. Then, we assess the impact of activating the MILP refinement.
5.2.1 MOACO configuration

The proposed MOACO features 18 parameters, which can hardly be tuned manually. We apply the IRACE tool (López-Ibáñez et al., 2011) to do so. IRACE automatically selects the best configuration of continuous, discrete and cardinal parameters among those defined as an input, considering a set of reference problem instances. This is an open-access state-of-the-art tuning procedure, which can be applied both to single and multi-objective problems. It is based on iterated racing procedures, namely the iterated F-race algorithm and several improvements over it based on advanced machine learning techniques, exploiting statistics observations supported by the Friedman test.

The parameter settings for the tuning procedure is reported in Table 4. Discrete sets of possible parameter configurations are described as comma-separated lists, while continuous ranges are reported in square brackets. A bold font highlights the values selected by IRACE. A fixed time limit of 500 s and a tuning budget (the maximum number of algorithm’s invocation that IRACE can exploit to perform the tuning) of 5000 iterations are set. The hypervolume HV (to be maximised, Fonseca et al., 2006) is chosen as the comparison KPI. With these settings, the tuning procedure takes almost 12 days to be completed, on an Intel(R) Xeon(R) CPU E5-2637 v3 @ 3.50GHz with 16 CPUs and 128 GB ram.

Table 4. Settings and results of the IRACE tuning procedure. “int” annotation highlights intervals defined in \( \mathbb{N} \), otherwise intervals are defined in \( \mathbb{R} \).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Best Parameter Configuration</th>
<th>Fixed Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{\text{ant}} )</td>
<td>20, 35, 50</td>
<td>( n_{\text{SMART}} : n_{\text{FAST}} )</td>
</tr>
<tr>
<td>( \alpha_{11} )</td>
<td>[0, 10] int 3.0</td>
<td>( n_{\text{LS}} : n_{\text{noLS}} )</td>
</tr>
<tr>
<td>( \beta_{11} )</td>
<td>[0, 10] int 3.0</td>
<td>( pc_{\text{Sol}_{LS}} )</td>
</tr>
<tr>
<td>( \rho_{11} )</td>
<td>[0.01, 0.1] 0.092</td>
<td>( N_{\text{upd}} )</td>
</tr>
<tr>
<td>( \tau_{\min,1} )</td>
<td>[0.05, 1.0] 0.467</td>
<td>1, 2, 3</td>
</tr>
<tr>
<td>( \tau_{\max,1} )</td>
<td>[6.0, 12.0] 8.0</td>
<td>( \text{maxTime} ) 500 s</td>
</tr>
<tr>
<td>( \alpha_{12} )</td>
<td>[0, 10] int 2.0</td>
<td>( \lambda_{\text{weights}} ) 5</td>
</tr>
<tr>
<td>( \beta_{12} )</td>
<td>[0, 10] int 8.0</td>
<td>( \lambda ) 0.5</td>
</tr>
<tr>
<td>( \rho_{12} )</td>
<td>[0.01, 0.1] 0.070</td>
<td></td>
</tr>
<tr>
<td>( \tau_{\min,2} )</td>
<td>[0.05, 1.0] 0.618</td>
<td></td>
</tr>
<tr>
<td>( \tau_{\max,2} )</td>
<td>[6.0, 12.0] 12.0</td>
<td></td>
</tr>
</tbody>
</table>

The tuning highlighted that the alternative use of the smart and fast TEG exploration modes is the best option, with an optimal ratio of 1:9 respectively. The optimal value for \( N_{\text{upd}} \) is equal to 3. This means that at the end of each iteration, three different solutions from the POS contribute to update the pheromone trails of a colony. This implies that a differentiation during the exploration of the solutions’ space is fostered. On the other side, it emerged that the proposed local search does not significantly improve the performance of the algorithm, and the tuning decides not to exploit it.

To strengthen the tuning conclusions with respect to four crucial parameters, namely \( \alpha, \beta \) and \( n_{\text{LS}} \), we perform further comparisons between four MOACO configurations:

- **BEST**: The parameter configuration returned by the tuning;
- **GRASP**: This configuration implements a Greedy Randomised Adaptive Search Procedure (GRASP) obtained from BEST by setting the \( \alpha \) parameters to 0 (in both L1 and L2), while all the others are left unchanged. In this way, we exclude the pheromone contribution to exploration. Exploration will therefore be guided (in a stochastic way) by the heuristic information only.
- **noHeurL1**: The dynamic computation of the heuristic information in Layer 1 is a computationally onerous operation, as it requires assessing the overlap index. However, the tuning highlights that it produces a performance improvement. In order to further investigate the trade-off between computational burden and performance boost, here we set \( \beta_{11} = 0 \) and turn off the computation.
- **BEST+LS**: The tuning chooses not to exploit at all the proposed local search. However, the 5th best configuration identified by IRACE in its internal ranking takes advantage of the local search with \( n_{\text{LS}} : n_{\text{noLS}} \) equal to 1:9 and \( pc_{\text{Sol}_{LS}} \) equal to 25. This configuration is
therefore tested.

For each configuration, we run all the 54 test instances, each of them with a random seed. A time limit of 500 s is set, consistently with the IRACE tuning. For having a different perspective with respect to IRACE, we then compare pairs of configurations by applying the Wilcoxon test to the differences between the HVs returned, for the same instance, with different configurations. The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test used to compare two populations using a set of matched samples. It returns whether the two sample sets belong to different populations (i.e., they are statistically different) and if the considered KPI of one set is statistically greater (or less) than the other, with a given confidence level. This test can therefore return if one configuration will in general perform better than the other for all the items of a hypothetic set from which the test instances are drawn. The confidence level used in our tests is 5%.

Also according to the Wilcoxon test, the performance of \textit{BEST} is significantly better than the one of the other configurations. The stronger evidence is in favour of the need of using the dynamic heuristic information in Layer 1, as the p-value returned from the test is the smallest in this comparison (9.918270374692144e-06). As for \textit{GRASP} and \textit{BEST+LS}, they lead to rather similar observations, with p-values of the order of $10^{-3}$.

Figure 6 reports the number of iterations performed by the four configurations within the 500 s time limit. Solutions are ordered on the x-axis depending on the number of iterations performed by \textit{BEST}. With respect to \textit{BEST}, an average difference of $+1.89\%$, $+15.82\%$ and $-48.97\%$ in the number of performed iterations is obtained for \textit{GRASP}, \textit{noHeurL1} and \textit{BEST+LS} respectively. For \textit{BEST}, the number of performed iterations spans between 190 and 4530. This shows that the larger impact on computational burden is definitely the one imposed by the application of the local search. This burden is not balanced by an improvement of performance in the fewer iterations performed. The computationally intensive heuristic information in Layer 1 has a minor impact on iterations, although indeed deactivating it in \textit{noHeurL1} allows increasing the count. However, the merits of the overlap index are such that the fewer iterations performed when using it allow ants to return better quality solutions. Finally, as expected, suppressing the use of pheromone in \textit{GRASP} does not translate in a remarkably different number of iterations performed.

![Figure 6. Number of performed iterations within the 500 s time limit.](image)

5.2.2 \textit{MILP refinement}

In this section, we report the results obtained when refining with the MILP formulation the timetables of the 54 POSs obtained with the \textit{BEST} configuration. To this purpose, $\text{DOF}_{\text{MILP}}$ is set to 20 minutes and the periodicity tolerance of all groups in all relevant locations is set to 10 minutes. The MOACO is run until a 500 s time limit is met, and then the MILP formulation is solved considering each timetable of the produced POS as input. The MILP formulation is solved by the GUROBI 9.1 commercial solver, which stops when either a 2% MIP gap or a 600 s time limit are
27 % of refined timetables are dominated and, as such, discarded. Not all the timetables limit (marked as “?” remaining 5, the MILP solver is not capable to verify no feasible timetable exists in the neighbourhood of the solutions produced by MOACO. For the reaching the 2 % MIP gap. The MILP formulation does not provide any feasible timetable for 13 for the remaining 24 the MILP solver find a feasible solution but stops after the 600 s without 17 of these 41 instances, the 2% MIP gap condition is met for all the MILP solution is found. required to solve the MILP formulation, calculated only f MILP refinement, calculated only for those timetables for which a feasible solution is found. For 41 of the 54 instances, the MILP formulation computes at least one conflict-free timetable. For 17 of these 41 instances, the 2% MIP gap condition is met for all the MILP-refined timetables, while for the remaining 24 the MILP solver find a feasible solution but stops after the 600 s without reaching the 2 % MIP gap. The MILP formulation does not provide any feasible timetable for 13 instances: for 8 of them, unfeasibility is attested (marked as “N/F” in column t MILP), meaning that no feasible timetable exists in the neighbourhood of the solutions produced by MOACO. For the remaining 5, the MILP solver is not capable to verify if the model is feasible within the 600 s time limit (marked as “?” in column t MILP).

Not all the timetables produced by the MILP refinement belong to a not-dominated POS: on average, 27 % of refined timetables are dominated and, as such, discarded.

Table 5 reports the results obtained for the 54 instances. For each instance, columns min CFL and max CFL report the minimum and maximum numbers of residual conflicts in the timetables produced by the MOACO. Columns ns MOACO and ns MILP report the cardinality of the POS before and after the MILP refinement, respectively. The difference between these two values indicates the number of timetables for which a conflict-free solution cannot be found by solving the MILP formulation. Column t MILP indicates the average computation time in minutes per timetable required to solve the MILP formulation, calculated only for those timetables for which a feasible solution is found.

![Figure 7. Improvement in the values of objectives TTT (blue bars) and ST (green bars) produced by the MILP](image_url)

**Table 5. Overview of main results from the application of the MILP formulation to the 54 test instances.**

<table>
<thead>
<tr>
<th>#</th>
<th>min CFL</th>
<th>max CFL</th>
<th>ns MOACO</th>
<th>ns MILP</th>
<th>t MILP (min)</th>
<th>#</th>
<th>min CFL</th>
<th>max CFL</th>
<th>ns MOACO</th>
<th>ns MILP</th>
<th>t MILP (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0,8</td>
<td>28</td>
<td>0</td>
<td>23</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>9</td>
<td>7</td>
<td>1</td>
<td>0,9</td>
<td>29</td>
<td>0</td>
<td>179</td>
<td>13</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>84</td>
<td>9</td>
<td>9</td>
<td>1,1</td>
<td>30</td>
<td>0</td>
<td>11</td>
<td>6</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>42</td>
<td>9</td>
<td>9</td>
<td>1,2</td>
<td>31</td>
<td>0</td>
<td>47</td>
<td>9</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>39</td>
<td>5</td>
<td>2</td>
<td>1,2</td>
<td>32</td>
<td>0</td>
<td>136</td>
<td>14</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>90</td>
<td>130</td>
<td>5</td>
<td>5</td>
<td>1,2</td>
<td>33</td>
<td>0</td>
<td>179</td>
<td>15</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>15</td>
<td>5</td>
<td>2</td>
<td>1,4</td>
<td>34</td>
<td>0</td>
<td>18</td>
<td>6</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>66</td>
<td>130</td>
<td>8</td>
<td>8</td>
<td>1,4</td>
<td>35</td>
<td>0</td>
<td>126</td>
<td>11</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>24</td>
<td>9</td>
<td>9</td>
<td>1,7</td>
<td>36</td>
<td>0</td>
<td>34</td>
<td>8</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>24</td>
<td>7</td>
<td>7</td>
<td>2,3</td>
<td>37</td>
<td>2</td>
<td>205</td>
<td>23</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>114</td>
<td>26</td>
<td>15</td>
<td>2,4</td>
<td>38</td>
<td>0</td>
<td>40</td>
<td>11</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>32</td>
<td>137</td>
<td>13</td>
<td>13</td>
<td>2,4</td>
<td>39</td>
<td>70</td>
<td>407</td>
<td>19</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>95</td>
<td>19</td>
<td>15</td>
<td>2,7</td>
<td>40</td>
<td>33</td>
<td>116</td>
<td>13</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>172</td>
<td>20</td>
<td>13</td>
<td>3,6</td>
<td>41</td>
<td>121</td>
<td>404</td>
<td>23</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>104</td>
<td>19</td>
<td>5</td>
<td>3,9</td>
<td>42</td>
<td>102</td>
<td>455</td>
<td>28</td>
<td>0</td>
<td>N/F</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>213</td>
<td>23</td>
<td>12</td>
<td>4,3</td>
<td>43</td>
<td>1</td>
<td>9</td>
<td>6</td>
<td>0</td>
<td>N/F</td>
</tr>
<tr>
<td>17</td>
<td>10</td>
<td>11</td>
<td>2</td>
<td>2</td>
<td>7,2</td>
<td>44</td>
<td>3</td>
<td>9</td>
<td>6</td>
<td>0</td>
<td>N/F</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>23</td>
<td>6</td>
<td>6</td>
<td>8,9</td>
<td>45</td>
<td>54</td>
<td>106</td>
<td>11</td>
<td>0</td>
<td>N/F</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>9,0</td>
<td>46</td>
<td>155</td>
<td>339</td>
<td>24</td>
<td>0</td>
<td>N/F</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>9,5</td>
<td>47</td>
<td>35</td>
<td>84</td>
<td>11</td>
<td>0</td>
<td>N/F</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>48</td>
<td>61</td>
<td>479</td>
<td>23</td>
<td>0</td>
<td>N/F</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>10</td>
<td>49</td>
<td>150</td>
<td>463</td>
<td>20</td>
<td>0</td>
<td>N/F</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>10</td>
<td>50</td>
<td>1</td>
<td>8</td>
<td>7</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>42</td>
<td>6</td>
<td>6</td>
<td>10</td>
<td>51</td>
<td>92</td>
<td>508</td>
<td>24</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>25</td>
<td>36</td>
<td>159</td>
<td>11</td>
<td>9</td>
<td>10</td>
<td>52</td>
<td>35</td>
<td>103</td>
<td>21</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>26</td>
<td>4</td>
<td>106</td>
<td>23</td>
<td>10</td>
<td>10</td>
<td>53</td>
<td>1</td>
<td>10</td>
<td>9</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>153</td>
<td>19</td>
<td>10</td>
<td>10</td>
<td>54</td>
<td>242</td>
<td>406</td>
<td>18</td>
<td>0</td>
<td>?</td>
</tr>
</tbody>
</table>
refinement.

Figure 7 describes the average percentage improvement in the objective functions’ values produced by the MILP refinement, for those instances for which at least one feasible timetable is found. Blue bars refer to improvement of the TTT objective, while green bars to the ST one. The figure highlights how dramatic improvements are produced by the MILP formulation, thanks to the exploitation of periodicity tolerances. Indeed, it is well-known that the enforcement of rigid periodic patterns is a severe capacity constraint mainly for operations on single track (Emery, 2010).

5.3 An application case study

In this section, we report the details of an application in which we actually deploy the ATMO framework in collaboration with Jernbanedirektoratet. Specifically,

We apply the ATMO framework to an infrastructure planning case study based on the BB infrastructure. As introduced in Section 5.1, this infrastructure consists in 370 km of single-track line linking Bergen and Hønefoss. The existing infrastructure is designated as scenario zero (S0) and two infrastructure improvement scenarios are defined each of which added 50 km of double track along the line. In scenario 1 (S1) three double track sections are placed equally spaced along the line. In scenario 2 (S2) a single long double track stretch is placed approximately in the middle of the line.

The case study objective is to obtain and analyse conflict-free timetables starting from a specified service concept, one for each scenario. The service concept covers a 24-hours day (starting at midnight) and includes passenger trains and a large set of optional freight trains. The aim is to maximise the number of freight trains scheduled and minimise additional total travel time due to crossings. Therefore, two ATMO objectives are activated, namely minimisation of additional travel time and maximisation of number of scheduled optional trains.

Three traffic patterns are defined in the case study:

- FR: 37 freight trains can be scheduled.
- LH+FR: 10 quasi-periodic long haul passenger trains (5 in each direction) must be scheduled, with up to 37 freight trains.
- R+LH+FR: In addition to the long-haul passenger trains, 30 periodic regional trains (15 in each direction) must be scheduled following a clock-face headway pattern between Bergen and Myrdal in the time window 7:00 AM to 10:00 PM, with up to 37 freight trains.

In addition to the traffic patterns, we also test the influence of periodicity tolerances in the periodic pattern of passenger trains. To this purpose the following two sets of traffic patterns are constructed to be overlaid on the FR, LH+FR and R+LH+FR ones:

- “X” pattern where strict periodicity is required (no periodicity tolerance).
- “T” pattern where a tolerance of ±15 minutes for long-haul trains and ±5 minutes tolerance for regional trains is allowed at all stations.

The infrastructure scenarios as well as the base service concepts are illustrated schematically in Figure 8.
The combination of three infrastructure alternatives and six service concepts leads to 18 scenarios. The ATMO framework is applied to all 18 scenarios. The following termination criteria are set to the algorithmic framework. The MOACO algorithm is stopped after a 30-minute time limit. Then the MILP formulation is applied to all the timetables in the provisional POS. The MILP formulation is solved using the GUROBI commercial solver, which stops when a 2% MIP gap or a 5-minute time limit is met. In 85% of the experiments for which the MILP formulation returns a feasible solution, the MIP gap termination condition is met. Differently from the test described in Section 5.2, experiments are carried out on a MacBook Pro with 2.6 GHz Intel Core i7 6 core processor and 16 GB RAM. The choice of this machine intends to provide a snapshot of the computational performance in a practical context.

Figure 9 presents the results as approximations of the Pareto fronts relevant to the “X” scenarios (no periodicity tolerance allowed). The points composing the fronts (lines) represent actual conflict-free timetables produced by ATMO. Each point on the line quantifies the KPIs selected for this case study, namely the number of scheduled freight trains and the total additional travel time due to crossings.

The lines display the best trade-off between the objectives (KPIs) for each infrastructure/service concept scenario. In particular, the figure shows that an upper bound exists for the number of freight trains that it is possible to schedule in each scenario, no matter how much additional travel time is scheduled.
The analysis of the timetables obtained by ATMO and shown in Figure 9 allows the assessment the various options considered in the case study and leads to the following conclusions:

- The difference between infrastructure scenarios S1 and S2 only arises when passenger trains are scheduled. In this case, S1 performs better than S2. When only freight trains are considered, there is no difference between S1 and S2 (dashed and dotted red line).

- The presence of passenger trains strongly affects the maximum number of freight trains that can be scheduled. In LH+FR scenarios, the maximum number of scheduled freight trains is almost 20% lower than in the FR scenarios, in average. In R+LH+FR scenarios, this percentage increases to 45%.

- Adding double track portions increases the number of freight trains that can be operated under all service concepts, without increasing the additional travel time. This achievable thanks to the reduction of the number of stops for crossings, which can now be performed in double track stretches. In particular, considering the upper-right extremities of the Pareto fronts, in scenarios S1 an average travel time reduction of 36% can be achieved compared to scenarios S0 (average value over the three service concepts). This is the horizontal distance between the right extremity of solid and dashed lines of the same colour. For scenarios S2, the average reduction is equal to 27% (solid and dotted lines). The maximum number of scheduled freight trains remains substantially constant in S0, S1 and S2.

![Figure 10](image-url)  
**Figure 10.** POS representing two objective function values in the “T” scenarios.

Figure 10 has the same structure of Figure 9, but it considers “T” scenarios in which periodicity can be relaxed. As expected, periodicity tolerances have no effects on FR scenarios, in which only single-train groups are considered. For the other scenarios, we can observe that results are qualitatively equivalent as far as the impact of the alternative infrastructure scenarios is considered. The main observable difference is a slight improvement of solution quality in terms of both KPIs optimised here, as it can be expected when constraints are relaxed as we do here. The highest impact is, not surprisingly, on total additional travel time.

This case study highlights the benefits of the new ATMO framework. Using a multi-objective optimisation and POS, it provides planners with much more information than a classic timetabling
approach which produces just one timetable at a time.

Furthermore, the tool also can produce these timetables efficiently. Table 6 illustrates, for each scenario, the total number of timetables returned as an output (# TT tot), the number of timetables in the resulting POSs (# TT POS) as well as the relevant computation time in minutes (c.t.). The table shows how the use of a multi-objective metaheuristic combined with a MILP-based refinement produces a significant number of feasible timetables in a relatively short computation time. However, not all the produced timetables belong to the resulting POSs, as already highlighted in Section 5.2.2. In total, this case study requires approximately 13.6 hours of computation time (it can be run overnight) and produces 822 timetables.

Table 6. Number of returned timetables and computation times for each scenario.

<table>
<thead>
<tr>
<th>scenario</th>
<th>S0</th>
<th></th>
<th></th>
<th>S1</th>
<th></th>
<th></th>
<th>S2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># TT tot</td>
<td># TT POS</td>
<td>c.t. (min)</td>
<td># TT tot</td>
<td># TT POS</td>
<td>c.t. (min)</td>
<td># TT tot</td>
<td># TT POS</td>
</tr>
<tr>
<td>FR</td>
<td>37</td>
<td>28</td>
<td>32,2</td>
<td>76</td>
<td>29</td>
<td>56</td>
<td>70</td>
<td>28</td>
</tr>
<tr>
<td>R+LH+FR X</td>
<td>53</td>
<td>10</td>
<td>107,2</td>
<td>34</td>
<td>8</td>
<td>68,5</td>
<td>37</td>
<td>7</td>
</tr>
<tr>
<td>LH+FR X</td>
<td>41</td>
<td>13</td>
<td>39,1</td>
<td>126</td>
<td>18</td>
<td>52,7</td>
<td>55</td>
<td>15</td>
</tr>
<tr>
<td>R+LH+FR T</td>
<td>30</td>
<td>10</td>
<td>62,3</td>
<td>59</td>
<td>8</td>
<td>66,1</td>
<td>38</td>
<td>8</td>
</tr>
<tr>
<td>LH+FR T</td>
<td>27</td>
<td>13</td>
<td>37,9</td>
<td>68</td>
<td>18</td>
<td>35,2</td>
<td>71</td>
<td>14</td>
</tr>
</tbody>
</table>

In summary, the case study shows that the ATMO framework can be used to quickly find feasible timetables in practical conditions. By doing so, it can be a fundamental tool both for defining yearly timetables and for supporting strategic planning decisions, by efficiently creating many feasible timetables for comparing infrastructure – service concept scenarios.

The ATMO results can also help planners to identify problematic areas to be studied in detail. More specifically, each point on a POS line represents an actual timetable that can be further analysed using, e.g., an iterative loop of adjustments and traffic simulations. Although omitted from this paper, the ATMO framework also includes a set of effective human-machine interfaces to ease as much as possible timetable analyses.

6 Conclusions

The paper proposed a novel algorithmic framework for automatic railway timetable generation, named ATMO. Thanks to its multi-objective approach, it provides the user with sets of Pareto-optimal timetables according to four different KPIs. The paper described how the ATMO is composed by the integration of a novel MOACO algorithm and a MILP formulation, and discussed a series of application cases which highlighted how its performance permits its practical deployment.

In future works, we will work to further improve MOACO, and hence ATMO, performance pursuing various directions. The first one concerns the local search improvement of ant-generated solutions. Local search is known to strongly improve algorithm performance, in general, but it fails to do so in our experiments. This is possibly due to its significant computational overhead. On the one hand, we will further investigate this overhead and possibly develop alternative approaches. On the other hand, we may exploit the MILP formulation refining ant solutions as a further local search. Particular attention must be paid when doing so, as the search spaces of MOACO and MILP are different: the former consider traffic conflicts as soft constraints and strict periodicity of train groups, while latter forbids conflicts and allows for periodicity tolerance.

With improved performance, the size of instances may be increased, reducing the necessary partitioning of the railway network and the resulting sub-optimisation of the overall timetable. Here, we have used real-world medium-sized instances. Future developments should be able to overcome larger instances than those manageable to human planners alone.

In this study, the neighbourhood of the MOACO solutions to be explored by the MILP formulation
is defined for time variables only. Other variables (station tracks, EEEs) are let free to vary regardless of the choices performed in the MOACO solution. A second direction for future works consists in focusing on an effective definition of a neighbourhood for these variables too, in order to reduce MILP computation times possibly without major quality detriment of solution quality.

From a railway-related perspective, the model of one of the objectives optimised, namely timetable stability, lends itself to further investigations. in this study, stability is modelled by means of the a-priori (according to Goverde and Hansen, 2013) KPI “minimum buffer time in the timetable”, to be maximised. However, many other KPIs could be designed to measure stability, provided that they can be integrated into the sequential solution construction implied by the MOACO. Further research should focus on a comparison between possible KPIs. Microsimulation of the generated timetables can be used to get a-posteriori stability indicators and quantify the comparison.

Finally, the ATMO provides the user with a potentially high number of timetable variants. This number would further increase if the user is interested not only in the Pareto Optimal Set of solutions, but in more of or all the feasible solutions produced during the algorithm’s iterations. Such a number of timetables can be exploited to perform analysis based on timetable patterns recognition which could provide more insight on the ways capacity can be used. Methods to massively analyse whole sets of timetables should therefore be developed, opening a novel (to the best of our knowledge) direction in the field of Railway Operation Research.

**Declaration of interest**

Thomas Nygreen is employed at the Norwegian Railway Directorate (Jernbanedirektoratet).

**Acknowledgments**

This paper has been developed as an integrating part of the Research Project n° 202000783 *Verktøy for utvikling av matematisk optimering av strategiske rutemodeller* (Tools for mathematical optimisation of strategic railway timetable models) promoted by the Norwegian Railway Directorate (Jernbanedirektoratet) and carried out by Trenolab SRLS and Gustave Eiffel University.

**Bibliography**


Appendix A. Computation of the Overlap Index

The overlap index estimates how much a yet to be scheduled train group would conflict with already scheduled ones.

Given a set $\mathcal{S}$ of already scheduled groups and a group $g$ to be scheduled, we denote as $\mathcal{W}$ the set of already scheduled trains (belonging to groups in $\mathcal{S}$) and $\mathcal{C}$ the set of trains belonging to group $g$. Given two trains $w \in \mathcal{W}$ and $c \in \mathcal{C}$, we consider the infrastructure resources shared by the two, each resource being a line section or a station track. On each of them, we calculate the conflict probability $p_{o,c,w}$ as the probability of having an overlap between the two “utilisation blocks” of $w$ and $c$. An “utilisation block” represents the time a resource is utilised by a single train, and cannot be utilised by other ones for safety reasons. On the one hand, the utilisation block of $w$ is fixed, as $w$ is already scheduled. It is defined by a start time $s_{w}$ and an end time $e_{w}$. On the other hand, the start time of $c$’s utilisation block can vary within a set of feasible times $[u_{c,\text{start}}, u_{c,\text{end}}]$ depending on the train journey, stops and running time capabilities. Given this start time, the separation between utilisations of $w$ and $c$ can be computed. If it is smaller than the minimum headway time, computed accounting for travel directions, then a conflict occurs. If $w$ uses $r$ before $c$, the minimum headway time is a constant, as it is a function of the first train travel and everything is already fixed for $w$. Otherwise, it depends on the speed and the EEE chosen for $c$ and it can vary in $[h_{c}, h_{c}]$. Let $\Delta_{h}=\bar{h}_{c}−h_{c}$ and $\Delta_{\text{start}}=u_{c,\text{end}}−u_{c,\text{start}}$ be the duration of the two relevant time intervals. We assume that $c$ utilisation start time $u_{c,\text{start}}$ and minimum headway if $c$ precedes $w$ are two independent random variables with uniform distributions in the intervals $[u_{c,\text{start}}, u_{c,\text{end}}]$ and $[h_{c}, h_{c}]$ respectively.

The probability that the two blocks overlap is given by the probability that the combination of the choices of $c$ utilisation start time and minimum headway if $c$ uses $r$ first overlaps the utilisation by $w$ and the minimum headway if the latter goes first. $p_{o,c,w}$ is equal to 0 if $u_{c,\text{start}} \leq u_{w,\text{end}}$ or $u_{c,\text{end}} \leq u_{w,\text{start}}$, otherwise we formalise it as

$$
\int_{A}^{B} \int_{A}^{C} 1 \Delta_{\text{start}} \left( \int_{u_{c,\text{start}}}^{u_{c,\text{end}}} \frac{1}{\Delta_{h}} dh_{c} \right) du_{c,\text{end}} + \int_{B}^{C} \frac{1}{\Delta_{\text{start}}} du_{c,\text{start}} 
$$

(A.1)

With

$$
A = \max \left( u_{c,\text{start}}, u_{w,\text{end}} − \bar{h}_{c} \right) \quad \text{(A.2)}
$$

$$
B = \min \left( u_{c,\text{end}}, u_{w,\text{start}} − h_{c} \right) \quad \text{(A.3)}
$$

$$
C = \min \left( u_{c,\text{end}}, u_{w,\text{start}} \right) \quad \text{(A.4)}
$$

Eq. (A.1) can be reduced to:

$$
p_{o,c,w} = \frac{u_{c,\text{end}} − u_{w,\text{start}}}{\Delta_{c,\text{end}}\Delta_{\text{start}}} (B − A) + \frac{B^{2} − A^{2}}{2\Delta_{c,\text{end}}\Delta_{\text{start}}} + \frac{C − B}{\Delta_{\text{start}}} \quad \text{(A.5)}
$$
The overlap index \( o_{c,w} \) of a train to be scheduled \( c \) with regards to an already scheduled one \( w \) is defined as

\[
o_{c,w} = \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{G}_c} \sum_{s \in \mathcal{S}} p_{t,c,w} + \sum_{i \in \mathcal{S}} \left( \sum_{t \in \mathcal{G}_c} p_{t,c,w} \right) \frac{1}{|\mathcal{T}_{g,t}|}
\]  \hspace{1cm} (A.6)

where \( \mathcal{L} \) and \( \mathcal{S} \) are the set of the line section and stations used by both \( c \) and \( w \). \( \mathcal{S} \) is the set of stations used by both \( c \) and \( w \), respectively. \( \mathcal{G}_c \), for a station \( s \in \mathcal{S} \), is the set of tracks used by both \( c \) and \( w \). This has always cardinality equal to 1, since the track for \( w \) is already defined. Finally, \( |\mathcal{T}_{g,t}| \) is the cardinality of the set of available tracks in station \( s \) for course \( c \).

The overlap index \( o_{i,g,s} \) of a group \( g \) with respect to a set of already scheduled groups \( \mathcal{S} \) is computed as the sum of the overlap index for each pair of trains.

**Appendix B. A MILP formulation for the TTP**

The MILP formulation used to refine solutions provided by the MOACO algorithm is here introduced. We mainly exploit the same data model and notation presented in Section 3. Two main differences exist with the approach implemented for the MOACO.

On the one hand, conflict constraints are implemented as hard ones. Output timetables are therefore conflict-free. On the other hand, the model considers individually each train of each group, with its own set of variable and constraints, which are no more defined at a train group level. Trains belonging to the same group are linked by a dedicated set of constraints, called train-linking, which allow to properly handle periodicity tolerances.

To this purpose, we define a set of trains \( \mathcal{T} \), and for each train \( t \in \mathcal{T} \) let \( g \) be the train group to which \( t \) belongs, \( i_t \) being the index of \( t \) within \( g \) (having period \( p_g \)), starting at 0 for the first train of the group. We define the sets of nodes \( \mathcal{V}_t \) and edges \( \mathcal{E}_t \) used by \( t \) along its journey, and a priority factor \( \rho_t = \rho_g \). We denote with \( l_t \) and \( l_t \) the first and last nodes respectively used by \( t \).

For each location (node) \( l \in \mathcal{V}_t \) we define:

- The minimum and maximum arrival times at \( l \), \( arr_{t,l} = arr_{g,l} + i_t \cdot p_g \) and \( \bar{arr}_{t,l} = arr_{g,l} + i_t \cdot p_g \) respectively. They are defined, only if \( l \neq l_t \).
- The minimum and maximum departure times from \( l \), \( dep_{t,l} = dep_{g,l} + i_t \cdot p_g \) and \( \bar{dep}_{t,l} = dep_{g,l} + i_t \cdot p_g \) respectively. They are defined, only if \( l \neq l_t \).
- The rolling stock used by the train on the edge leaving \( l \), only if \( l \neq l_t \);
- The set \( T_{t,l} = T_{g,l} \) of usable station tracks;
- The set \( \Gamma_{t,l} = \Gamma_{g,l} \) of usable pass/stop modes;
- The minimum and maximum stop times \( \bar{stop}_{t,l} = stop_{g,l} \) and \( \bar{stop}_{t,l} = stop_{g,l} \);
- The minimum and maximum run times admitted for the train in the edge \( e \in \mathcal{E}_t \) leaving \( l \) (only if \( l \neq l_t \)), \( \bar{rt}_{t,e} = rt_{g,e} \) and \( \bar{rt}_{t,e} = rt_{g,e} \);
- The linked train at a location \( l \) \( l_{t+1} \). This is the train w.r.t. which we need to impose a periodicity at \( l \): it is the one following \( t \) in its group \( i_{l_{t+1}} = i_t + 1 \). If no linked train exists, \( l_{t+1} = \).
- The minimum and maximum time separation w.r.t. the arrival and departure times of \( l_{t+1} \) at a location \( \bar{tsa}_{l_{t+1}} \), \( \bar{tsd}_{l_{t+1}} \), \( \bar{tsa}_{l_{t+1}} \), \( \bar{tsd}_{l_{t+1}} \). With these values we manage group periodicity and periodicity tolerance as follows:
  - \( tsd_{l_{t+1}} = p_g - tol_{g,l} \);
  - \( tsd_{l_{t+1}} = p_g + tol_{g,l} \);
  - \( tsd_{l_{t+1}} = p_g - tol_{g,l} \);
treated as hard constraints and the number of scheduled trains is an input. 

In the formulation, we only deal with three objectives

\[ \text{Multi} \]

large constant, respectively.

Finally, to measure robustness, we de

node. The precedence relation on edges only imposes the impossibility of having two trains travel in the same direction, the relation considers the moment at which they leave the same node. The precedence relation on edges only imposes the impossibility of having two trains simultaneously present if they travel in the opposite directions. Conversely, when the precedence concerns tracks in nodes, in \( s \) variables, the precedence relation concerns the whole duration of the presence of two trains, as a simultaneous occupation of a track is forbidden.

For \( p \) and \( s \) variables, the precedence symbol \(<\) indicates two slightly different relations. When it refers to edges, in \( p \) variables, the relation concerns edge entrance times. Indeed, if two trains travel in opposite directions, then their entrance in the edge occurs at the opposite extremes. Instead, if two trains travel in the same direction, the relation considers the moment at which they leave the same node. The precedence relation on edges only imposes the impossibility of having two trains simultaneously present if they travel in the opposite directions. Conversely, when the precedence concerns tracks in nodes, in \( s \) variables, the precedence relation concerns the whole duration of the presence of two trains, as a simultaneous occupation of a track is forbidden.

Finally, to measure robustness, we define two non-negative continuous variables:

\[ \text{Minimum buffer time on edges } mbe; \]
\[ \text{Minimum buffer time on tracks } mbr. \]

In addition to the input data described above, we will use \( m \) and \( M \) to indicate vary little and very large constant, respectively.

Multi-objective optimisation is handled by the MILP formulation by aggregating a set of normalised KPIs into a single objective function to be minimised (Eq. (B.7)). With respect to the MILP formulation, we only deal with three objectives-KPIs, namely TTT, EC and ST, since conflicts are treated as hard constraints and the number of scheduled trains is an input. In Eq. (B.7) the notation
At each node, the separation between the arrival and the departure times of each train must be coherent:

\[
\min \sum_{t \in T} \left( wTT \cdot (pr_t \cdot (arr_{t,k} - dep_{t,k})) + wEC \cdot (pr_t \cdot \sum_{e \in E_t} ec_{t,e}) \right) + wST \tag{B.7}
\]

On the one hand, the objective considers, for each train, the weighted travel time and energy consumption. On the other hand, it accounts for weighted minimum buffer time on edges and tracks. Remark that while the train components are to be minimised, the buffer time ones are to be maximised, but the normalisation operator takes care of properly setting the sign of each addendum.

This objective must be optimised while respecting several sets of constraints:

Each train must run according to exactly one Edge Extremity Event for each edge:

\[
\sum_{e \in E_{EE_t,e}} y_{t,e} = 1 \quad \forall t \in T, e \in E_t \tag{B.8}
\]

Each train must use exactly one of its available tracks in each node:

\[
\sum_{e \in E_{T_e,v}} z_{t,e} = 1 \quad \forall t \in T, v \in E_t \tag{B.9}
\]

Each train cannot stay in a node less than the minimum or more than the maximum stop time:

\[
dep_{t,v} - arr_{t,v} \geq stop_{t,v} \quad \forall t \in T, v \in V_t \setminus \{I_t, \bar{I}_t\}, stop_{t,v} > 0 \tag{B.10}
\]

\[
dep_{t,v} - arr_{t,v} \leq \bar{stop}_{t,v} \quad \forall t \in T, v \in V_t \setminus \{I_t, \bar{I}_t\}, \bar{stop}_{t,v} > 0 \tag{B.11}
\]

Arrival and departure times of a train at a node are separated by little constant \(m\) if the train does not stop. This always holds for nodes where stops are forbidden \((\bar{stop}_{t,v} = \bar{stop}_{t,v} = 0)\). If the stop is optional \((\bar{stop}_{t,v} = 0, \bar{stop}_{t,v} > 0)\) this must hold if the chosen EEE imposes a stop: they will link the EEE with the first node of the edge:

\[
arr_{t,v} = dep_{t,v} - m \quad \forall t \in T, v \in V_t \setminus \{I_t, \bar{I}_t\}, stop_{t,v} = \bar{stop}_{t,v} = 0 \tag{B.12}
\]

\[
arr_{t,v} \leq dep_{t,v} - m \quad \forall t \in T, v \in V_t \setminus \{I_t, \bar{I}_t\}, stop_{t,v} = 0, \bar{stop}_{t,v} > 0 \tag{B.13}
\]

\[
dep_{t,v} - arr_{t,v} - m \leq \left( \bar{stop}_{t,v} - m \right) \sum_{e \in E_{EE_t,e}(\{s,s\},\{s,p\})} y_{t,e} \quad \forall t \in T, v \in V_t \setminus \{I_t, \bar{I}_t\}, e \in E_t, e = v, \bar{stop}_{t,v} = 0, \bar{stop}_{t,v} > 0 \tag{B.14}
\]

By using constant \(m\), we impose that the presence of the train is recorded for a strictly positive time in all situations. This has no actual impact for practical purposes (we can set \(m\) to half a second for example) but allows ensuring the disjunction of track utilisation by different trains.

If an optional stop can be performed at a node, then the EEE chosen for reaching and leaving the node must be coherent:

\[
\sum_{e \in E_t, e \neq v} \sum_{e \in E_{EE_t,e}(\{s,s\},\{s,p\})} y_{t,e} = \sum_{e \in E_t, e \neq v} \sum_{e \in E_{EE_t,e}(\{s,s\},\{s,p\})} y_{t,e} \quad \forall t \in T, v \tag{B.15}
\]

At each node, the separation between the arrival and the departure times of each train \(t\) and its linked one \(lt_{t,v}\) at a node are in the acceptable intervals:
\[ \text{arr}_{l_{t,v}} - \text{arr}_{t,v} \geq \text{tsa}_{l_{t,v} v}, t \in T, v \in V \setminus \{l_v\} : \text{lt}_{t,v} \neq t \quad \text{(B.16)} \]
\[ \text{arr}_{l_{t,v}} - \text{arr}_{t,v} \leq \text{tsa}_{l_{t,v} v}, t \in T, v \in V \setminus \{l_v\} : \text{lt}_{t,v} \neq t \quad \text{(B.17)} \]
\[ \text{dep}_{l_{t,v}} - \text{dep}_{t,v} \geq \text{tsd}_{l_{t,v} v}, t \in T, v \in V \setminus \{l_v\} : \text{lt}_{t,v} \neq t \quad \text{(B.18)} \]
\[ \text{dep}_{l_{t,v}} - \text{dep}_{t,v} \leq \text{tsd}_{l_{t,v} v}, t \in T, v \in V \setminus \{l_v\} : \text{lt}_{t,v} \neq t \quad \text{(B.19)} \]

Remark that we only impose this constraint for a train \( t \) if \( \text{lt}_{t,v} \neq t \), i.e. if there is a train following \( t \) in the group.

For each train, the running time for an edge must comply with the minimum technical one, also considering the track used at each extreme node:

\[
\text{rt}_{t,e} \geq \sum_{e \in \text{EE}_t e} \left( m_{e_{\text{dir},t,e}} r_{e} + \sum_{e \in \text{EE}_t e} \Delta r_{\epsilon_{e}} (z_{t,e} + y_{t,e} - 1) \right) \forall t, e \in E
\]

(B.20)

Here, we do the sum over all EEEs that can be used on the edge, knowing than at most one of them will be used. If an EEE is not used, variable \( y_{t,e} \) is set to 0 and all the elements of the sum disappear. In particular, additional run time is considered only if both the EEE and the track are used: in this case \( z_{t,e} + y_{t,e} - 1 = 1 \), otherwise it equals 0.

For each train, the energy consumption on an edge must be coherent with the EEE is runs according to, plus the additional consumption due to the difference between the chosen and the minimum running time:

\[
\text{ect}_{t,e} \geq \text{me}_{e_{\text{dir},t,e}} r_{e} \cdot \text{rt}_{t,e} - \text{me}_{e_{\text{dir},t,e}} r_{e} \cdot \left( \text{rt}_{t,e} - \text{mrt}_{e_{\text{dir},t,e}} r_{e} - M(1 - y_{t,e}) \right) \forall t, e \in E
\]

(B.21)

If two trains \( t, t' \) may claim the same edge \( e \) concurrently traveling in the same direction, their departure times from the first node of the edge must be separated at least of the minimum headway time increased of a factor proportional to the increase of running time of the first train passing with respect to the minimum:

\[
\text{h}_{t,t'} = \text{mh}_{e_{\text{dir},t,e}} + \text{kh}_{e_{\text{dir},t,e}} \cdot \left( \text{rt}_{t,e} - \text{mrt}_{e_{\text{dir},t,e}} \right)
\]

(B.22)

\[
\text{dep}_{t,u} - \text{dep}_{u,t} \geq \text{h}_{t,t'} = \text{M}(3 - \text{p}_{t,t',e} - y_{t,e} - y_{t',e}) \forall e \in E, (t, t') \in S_{e}, u \in V_{e}, u = \text{u} \text{ for dir}_{t,e} \in \text{EE}_{t,e}, e' \in \text{EE}_{t',e}, \text{dir}_{t,e} = \text{dir}_{t,e} e
\]

(B.23)

\[
\text{h}_{t,t'} = \text{mh}_{e_{\text{dir},t,e}} + \text{kh}_{e_{\text{dir},t,e}} \cdot \left( \text{rt}_{t,e} - \text{mrt}_{e_{\text{dir},t,e}} \right)
\]

(B.24)

\[
\text{dep}_{t,u} - \text{dep}_{u,t} \geq \text{h}_{t,t'} = \text{M}(2 + \text{p}_{t,t',e} - y_{t,e} - y_{t',e}) \forall e \in E, (t, t') \in S_{e}, u \in V_{e}, u = \text{u} \text{ for dir}_{t,e} \in \text{EE}_{t,e}, e' \in \text{EE}_{t',e}, \text{dir}_{t,e} = \text{dir}_{t,e} e
\]

(B.25)

In Eq. (B.22), if \( t < t' (p_{t,t',e} = 1) \) and both trains use the edge them the constraint imposes the respect of the suitable minimum headway time, otherwise it is trivially satisfied. In parallel, Eq. (B.24) imposes the headway if \( t' < t (p_{t,t',e} = 0) \).

If two trains \( t, t' \) may claim the same edge \( e \) concurrently traveling in opposite directions, the departure time of the second train from the first node of \( e \) it crosses must be greater than the arrival time of the first train at this same node:

\[
\text{dep}_{t',v} \geq \text{arr}_{t,u} + \text{mh}_{e_{\text{dir},v,e}} - M(1 - \text{p}_{t,t',e}) \forall e \in E, (t, t') \in S_{e}, v \in V_{e}, v = \text{v} \text{ for dir}_{t,e} \neq \text{dir}_{t,e}
\]

(B.26)
\[
\text{dep}_{t,u} \geq \text{arr}_{t,u} + mh_{e, \text{dir}_{t,e}, u} - M \cdot p_{t,t',e} \forall e \in E, (t, t') \in S_{\tau}, u \in V_t, e \tag{B.27}
\]

\[
= u \text{ for } \text{dir}_{t,e}, \quad \text{dir}_{t,e} \neq \text{dir}_{t',e}
\]

As for the previous couple of constraints, the first one is relevant if \(p_{t,t',e} = 1\) and the second one if \(p_{t,t',e} = 0\).

For each track \(\tau\), for each pair of train \(t, t'\) that may claim it concurrently, we impose that the track occupation is not overlapping, also considering the minimum separation time:

\[
\text{arr}_{t',v} \geq \text{dep}_{t,v} + t g_v - M(3 - s_{t,t',e} - z_{t,e} - z_{t',e}) \forall v \in V_t, \tau \in T_{t,v} \cap T_{t',v}, (t, t') \tag{B.28}
\]

\[
\text{arr}_{t,v} \geq \text{dep}_{t',v} + t g_v - M(2 + s_{t,t',e} - z_{t,e} - z_{t',e}) \forall v \in V_t, \tau \in T_{t,v} \cap T_{t',v}, (t, t') \tag{B.29}
\]

The minimum buffer time on edges \(mbe\) is the shortest time separating the entrance of two trains on the same edge if they travel in the same direction, minus the minimum headway time.

\[
mbe \leq \text{dep}_{t',u} - \text{dep}_{t,u} + h t_{t'} + M(3 - p_{t,t',e} - y_{t,e} - y_{t',e}) \forall e \in E, (t, t') \in S_{\tau}, u \in V_t, e \tag{B.30}
\]

\[
= u \text{ for } \text{dir}_{t,e}, \quad \text{dir}_{t,e} \neq \text{dir}_{t',e}
\]

\[
mbe \leq \text{dep}_{t',u} - \text{dep}_{t,u} + h t_{t'} + M(2 - y_{t,e} - y_{t',e}) \forall e \in E, (t, t') \in S_{\tau}, u \in V_t, e \tag{B.31}
\]

\[
= u \text{ for } \text{dir}_{t,e}, \quad \text{dir}_{t,e} \neq \text{dir}_{t',e}
\]

\[
mbe \leq \text{dep}_{t,u} - \text{dep}_{t',u} + mh_{e, \text{dir}_{t',e}, u} + M \cdot p_{t,t',e} \forall e \in E, (t, t') \in S_{\tau}, u \in V_t, e \tag{B.32}
\]

\[
= u \text{ for } \text{dir}_{t,e}, \quad \text{dir}_{t,e} \neq \text{dir}_{t',e}
\]

Remark that Eq. (B.31) and Eq. (B.34) concern the pairs of trains that use the same edge but for which the utilisation order is defined in the input data, due to the minimum and maximum arrival and departure times at the extreme nodes.

The minimum buffer time on station tracks \(mbr\) is the shortest time separating the departure and the arrival of a train on the same platform of a node, minus the minimum separation time:

\[
mbr \leq \text{arr}_{t',v} - \text{dep}_{t,v} - t g_v + M(3 - s_{t,t',e} - z_{t,e} - z_{t',e}) \forall v \in V_t, \tau \in T_{t,v} \tag{B.33}
\]

\[
\text{arr}_{t,v} \geq \text{dep}_{t',v} - t g_v + M(2 + s_{t,t',e} - z_{t,e} - z_{t',e}) \forall v \in V_t, \tau \in T_{t,v} \tag{B.34}
\]

\[
\text{arr}_{t,v} \geq \text{dep}_{t',v} - t g_v + M(2 + s_{t,t',e} - z_{t,e} - z_{t',e}) \forall v \in V_t, \tau \in T_{t,v} \tag{B.35}
\]

Through \(p\) and \(s\) variables, this formulation uses disjunctive constraints to guarantee the respect of capacity constraints on tracks, following the modelling principles of RECIFE-MILP (Pellegrini et al., 2015). However, the two models consider different representations of the infrastructure, which
here is macroscopic while it is microscopic in RECIFE-MILP. Here, we associate arrival and departure times to nodes rather than specific tracks. This explains higher complexity of the disjunctive constraints presented from Eq. (B.22) to Eq. (B.29), in which we need to ensure no relation is imposed between times if trains use alternative tracks.

A set of valid inequalities, also used in by Mannino et al. (2015) for railway scheduling, may be implemented for strengthening the model:

\[ p_{tt',e} \leq p_{tt',e'} \quad \forall e, e' \in E, (t, t') \in S_e \cap S_{e'}, \text{dir}_{t,e} \neq \text{dir}_{t',e}, e' < e \text{ for } \text{dir}_{t,e} \quad (B.39) \]

They exploit the observation that the precedence relation between trains traveling in opposite direction propagates along their route. In particular, if two trains \( t \) and \( t' \) use a sequence of edges including \( e \), and \( t \) precedes \( t' \) on one of \( e \), the \( t \) is the first to pass also on all edges \( e' \) it crosses before arriving to \( e \): \( p_{tt',e} = 1 \Rightarrow p_{tt',e'} = 1 \). From the opposite perspective, if \( t' \) passes first in \( e' \), then it is also first in \( e \), as for it \( e \) precedes \( e' \): \( p_{tt',e'} = 0 \Rightarrow p_{tt',e} = 0 \).