



HAL
open science

Integrated Ant Colony Optimization and Mixed Integer Linear Programming for Multi-objective Railway Timetabling

Nicola Coviello, Giorgio Medeossi, Thomas Nygreen, Paola Pellegrini, Joaquin Rodriguez

► **To cite this version:**

Nicola Coviello, Giorgio Medeossi, Thomas Nygreen, Paola Pellegrini, Joaquin Rodriguez. Integrated Ant Colony Optimization and Mixed Integer Linear Programming for Multi-objective Railway Timetabling. 26th IEEE International Conference on Intelligent Transportation Systems ITSC 2023, Sep 2023, Bilbao, Spain. hal-04280466

HAL Id: hal-04280466

<https://univ-eiffel.hal.science/hal-04280466v1>

Submitted on 11 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Integrated Ant Colony Optimization and Mixed Integer Linear Programming for Multi-objective Railway Timetabling

Nicola Coviello¹, Giorgio Medeossi¹, Thomas Nygreen², Paola Pellegrini³ and Joaquin Rodriguez³

Abstract—This paper presents an algorithmic framework for automatic railway timetabling, developed within the project “Tools for mathematical optimization of strategic railway timetable models” funded by the Norwegian Railway Directorate (Jernbanedirektoratet). It describes the algorithmic core of the developed tool, called Automatic Timetabler with Multiple Objectives. The framework integrates a Multi-Objective Ant Colony Optimization (MOACO) algorithm and a Mixed Integer Linear Programming (MILP) formulation. MOACO performs a fast-but-coarse exploration of the solution space, populating and maintaining an approximated Pareto optimal set of timetables. The timetables generated by MOACO are refined by the MILP formulation, exploring a neighborhood of the input solution and returning feasible, high-quality timetables. The tool is assessed on case studies driven from real practice in Norway.

I. INTRODUCTION

To maximize the benefit of passengers and freight customers, railway agencies rely on the generation of timetable hypotheses to identify possible infrastructure and timetable improvements. Starting with conceptual railway service requirements, they face a trade-off between several conflicting objectives, such as travel time, capacity utilization, robustness and energy consumption. Many timetable hypotheses may have to be generated, with various service concepts and alternatives for each concept. Timetable generation is a time-consuming process, and its (partial) automation may largely improve planners’ productivity. The efficient production of optimized timetables is crucial to support such automation.

The Train Timetable Problem (TTP) is a classical NP-hard problem in Operations Research [11]. It consists in defining the arrival and departure times of trains in stations and in selecting their routing across the network. A number of exact methods have been proposed for the TTP [3], [4], [15], and they have shown their usefulness when their scope is appropriately defined. Besides exact methods, heuristic and meta-heuristic algorithms have been designed, with promising results. Despite the richness of the TTP literature,

the use of optimization algorithms for practical timetabling has not yet spread in the industry. One of the main reasons for this is that the problem variants tackled struggle to represent the multifaceted practical problem [1], [2], [12].

In this work, we model the TTP as the problem of transforming a service concept into a feasible timetable, and we specifically address all features considered important by Jernbanedirektoratet to design timetables. A service concept is a set of conceptual railway service requirements. It is basically a timetable draft. It provides the number and type of trains that have to figure in the output timetable, respecting given technical and operational constraints. Technical constraints make the output timetables feasible, i.e., ensure that in real operation trains can respect their schedule, at least in absence of traffic perturbations [10]. Operational constraints foster the compliance with commercial and organizational 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 optimize a set of objective functions.

This paper presents ATMO (Automatic Timetabler with Multiple Objectives), which is a novel algorithmic framework to tackle the TTP. ATMO was developed within the project “Tools for mathematical optimization of strategic railway timetable models” funded by the Norwegian Railway Directorate (Jernbanedirektoratet) and carried out by Trenolab SRLS and Université Gustave Eiffel. The term “strategic” in the project’s name, refers to the planning stage at which the tool is intended to be used. During strategic timetable planning, many variables are left open, regarding the service to be carried on as well as the infrastructure. Hence, a large number of scenarios are likely to be assessed, thus requiring to automatize as much as possible the whole process.

ATMO implements a multi-objective approach and returns a set of timetables representing a Pareto-Optimal Set approximation (POS). A preliminary version of ATMO was presented in [7]. In that paper, we focused mainly on the potential practical impact of the application of optimization for assessing infrastructure modification options. In the present paper, we formally describe the framework finally produced and we propose a more extensive experimental analysis.

In ATMO, we model the TTP as a two-layer graph problem, merging the representation of a clique and a path problem. A clique on a Layer 1 graph defines the set of trains scheduled. Paths on Layer 2 graphs define the travel characteristics of these trains.

We exploit both the meta-heuristic and exact algorithmic perspectives for tackling the two-layer graph model: we com-

*This work was 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.

¹Nicola Coviello and Giorgio Medeossi are with Trenolab SRLS, Via Maniacco 7/A, 34170 Gorizia, Italy {n.coviello, g.medeossi}@trenolab.com

²Thomas Nygreen is with the Norwegian Railway Directorate (Jernbanedirektoratet), Postboks 16, Oslo, Norway, NO-0101 thomas.nygreen@jernbanedirektoratet.no

³Paola Pellegrini and Joaquin Rodriguez are with Univ Gustave Eiffel, COSYS-ESTAS, F-59650 Villeneuve d’Ascq, France {paola.pellegrini, joaquin.rodriguez}@univ-eiffel.fr

bine a Multi Objective Ant Colony Optimization (MOACO) algorithm and a Mixed-Integer Linear Programming (MILP) formulation. MOACO integrates the state of the art on Ant Colony Optimization (ACO) applications for solving minimum cost clique and path problems. It provides a fast-but-coarse exploration of the solution space, populating and maintaining a POS considering up to five objectives (four real objectives plus one fostering the minimization of soft-constraint violations). The solutions of this POS are then refined by MILP. It explores their neighborhood and finally provides feasible, high-quality timetables.

The rest of the paper is structured as follows. Section II describes the adopted data model and the relevant formulation for the TTP. Section III introduces the algorithmic principles of ATMO. Section IV presents and discusses numerical experiments and Section V reports our conclusions and proposes some possible future developments.

II. INFRASTRUCTURE, OPERATIONS AND SERVICE CONCEPT MODELS

We adopt a macroscopic model for infrastructure and operations, based on a multi-graph. Nodes represent timing locations (stations, junctions, halts) in the rail network. To ensure the microscopic feasibility of the produced timetable, for each node, we define: a set of tracks, one of which is identified as the main one; an additional running time if a track different from the main one is used by a rolling stock; 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, to ensure 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. Trains can pass locations in two different modes: passing or stopping. This influences running time, minimum headway and energy consumption of trains on line stretches. 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 consumption and headway between trains running in the same direction. Modeling these relationships as linear ones is a strong assumption, but it allows an easier integration into a MILP formulation. Empirical experience gathered by the authors permits to assume that, at least in the considered case studies, the distortions introduced by the linearization are acceptable as long as the run time upper bound is not larger than 1.15 times the minimum technical run time of each line section.

The service concept model describes the 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 modeled as single-train groups. Three types of groups can be defined: *fixed groups* are constraints to the timetabling process as they are to be set exactly as described; *movable groups* can be adjusted in time and

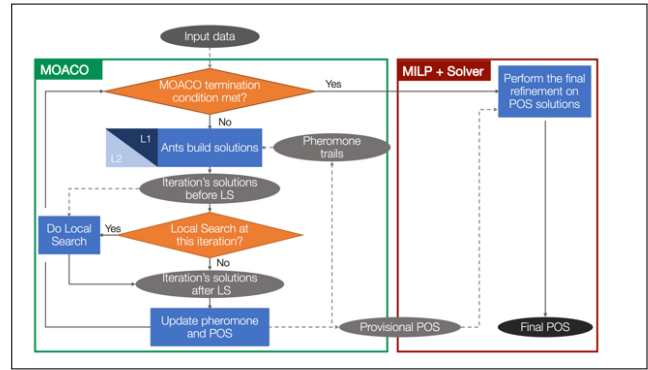


Fig. 1. Global architecture of the ATMO framework

space (station routing) to optimize the timetables; *optional groups* are movable groups that may be excluded from the timetable. Train groups can have different priorities, which define their contribution to the objective functions. Moreover, a time discretization step is defined for each group. The relevant times of a train group in a timetable must be set to multiples of this discretization step. A small discretization step indicates the need of high precision associated to a train group. For example, a passenger train group may have a smaller step than a freight one. A non-negative *periodicity tolerance* defines the maximum acceptable deviation from a strictly periodic pattern. Null tolerance imposes that arrivals and departures in a node of pairs of trains of the group are separated by exactly a multiple of the group's period.

III. THE ATMO SOLUTION FRAMEWORK

Figure 1 displays the architecture of the ATMO framework. Grey ovals represent data contents, blue rectangles processes and orange diamonds conditional switches. Solid arrows mark the operation flows, while dashed ones are data feedings. After reading the input data, MOACO is started. It iteratively creates and updates a provisional POS (Section III-A). After the stopping criterion is satisfied, i.e., after a given number of iterations have been completed or the available computational time has elapsed, the MILP formulation refines all POS solution (Section III-B).

ATMO assesses the quality of timetables according to four objectives: *TTT*, minimize total weighted travel time, i.e., the sum over all trains of the difference between last arrival and first departure time, weighted by the group priority; *EC*, minimize total energy consumption, weighted by group priorities; *NTR*, maximize the number of scheduled trains weighted by group priorities; *ST*, maximize timetable stability. These objectives are computed as follows. A train travel time is the difference between its arrival time at the last station and its departure time from the first one. The total weighted travel time is the sum of the train travel times multiplied by the respective group priority factor. Energy consumption strictly depends on the running times of trains on each edge, as well as on whether trains perform passes or stops at stations where both options are possible. All energy consumption values are input data. The total weighted energy

consumption is the sum of the train energy consumption multiplied by the respective group priority factor. The total weighted number of scheduled trains is the product of the number of trains of each optional group included in the timetable and the respective group priority factor. Stability is measured as the minimum *buffer time* in the timetable (time separation between two consecutive utilizations of the same line stretch or station track minus the minimum admissible headway between the corresponding train movements). The equations used to compute the objective function values are detailed in [6].

A. MOACO algorithm

ACO algorithms [8] are based on the iterative construction of feasible solutions by a colony of artificial ants. Here, we focus on Max-Min Ant System [21], one of the most successful ACO algorithm, for which a very well performing multi-objective variant exists [14].

The construction of a feasible solution consists in the sequential selection of a set of edges on a construction graph. Ants construct solutions progressively, stochastically selecting the edge to add at each step through the pseudo-random proportional rule. Here, the selection is biased by a greedy measure of the quality of the node (heuristic information η) and by a quantity representing the accumulated knowledge of the colony on the actual quality (pheromone trail τ). In particular, being in node u , an ant selects the edge to add from a *candidate* set $CAND$: each edge $(u, v) \in CAND$ has a probability of being selected equal to $p_{(u,v)} = \frac{\tau_{(u,v)}^\alpha \eta_{(u,v)}^\beta}{\sum_{v' \in CAND} \tau_{(u,v')}^\alpha \eta_{(u,v')}^\beta}$, with α and β parameters of the algorithm. Pheromone is updated at each iteration to progressively increase the trail on the components of solution S , which is either the best so far or the iteration best solution: $\tau_{(u,v)} = (1-\rho)\tau_{(u,v)} + \delta S$, with $\rho \in [0, 1]$ parameter and δS a measure of the quality of S . To avoid stagnation, pheromone is always bounded to be in the interval $[\tau_{min}, \tau_{MAX}]$. The process stops when a termination condition is met (e.g., maximum number of iterations or elapsed time).

We adopt ACO for three main reasons. First, it suits well the solution of the TTP since it builds solutions incrementally, thus mimicking real-practice timetables planning. This will foster the understanding and acceptance of the way ATMO works by practitioners. Second, ACO 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 capture the implicit bonds between different choices eliminating the need for modeling them explicitly. On the other hand, (proxys of) our objective functions components can be exploited as powerful local information. Third, ants generate and maintain populations of solutions, which fits particularly well the production of a POS.

1) *Multi-objective extension*: To deal with the various objective functions of the TTP, we follow the multi-colony architecture proposed by [14]. Specifically, several colonies of artificial ants build solutions concurrently, each consider-

ing its own pheromone trail and objective function. Throughout iterations, the POS is progressively filled and updated. We have five colonies: each colony C_o optimizes 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 minimize. A conflict occurs when a pair of trains utilize the same infrastructure resource without respecting the minimum headway. By adding the CFL objective, we relax these technical constraints. Hence, MOACO does not necessarily return practically applicable solutions. However, preliminary analysis has shown that this choice is definitely improving final solution quality, especially as MOACO solutions are refined through the MILP formulation, which restores applicability.

Both heuristic information and pheromone trail are defined for each objective. The former is common to all colonies ($\eta_{\hat{o}}, \forall \hat{o} \in O$), the latter evolves independently in each of them ($\tau_{\hat{o}, C_o}, \forall \hat{o}, o \in O$). This means that each colony C_o , features its own pheromone trails $\tau_{\hat{o}, C_o}$ for each objective. The values associated to different objectives are normalized and they are all bounded in the interval $[\tau_{min}, \tau_{MAX}]$.

The weighted sums of heuristic information and pheromone trails associated to all objectives are the τ and η values to be used in the pseudo-random proportional rule. The weights used here vary across iterations and colonies. For colony C_o , the weight of pheromone trail and heuristic information associated to objective o (λ_o^i at iteration i) is initially set to one and decreases by 0.1 at each iteration until it reaches 0.5. Then, it is set back to one and the progressive decrease starts again. The weights of all other objectives ($\lambda_{\hat{o}}^i, \forall \hat{o}, o \in O$ 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 be 1. Each colony manages its own pheromone trails 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 [14]. The best $N_{upd} \geq 1$ solutions of each region are used for the pheromone update. Solution quality is measured in terms of number of residual conflicts, first, and on the o objective function value, second. Let S be a solution of R_o , and \hat{S}_o be the best in R_o according to the defined lexicographic order. The pheromone deposited on the edges composing S is the ratio of two terms. The denominator is the difference of the number of residual conflicts in S and \hat{S}_o , augmented of one unit to avoid zeros. The numerator is the inverse of the worsening of S w.r.t. \hat{S}_o for o .

To the best of our knowledge, the mix of actual objective functions and soft constraint penalization in multi-objective ACO has never been considered before. Preliminary experiments led us to the lexicographic ordering of R_o , which strongly improves the ability of MOACO to find solutions with few residual conflicts.

2) *Two-layer extension*: In MOACO, a solution is the combination of the following sub-solutions: a Layer 1 sub-solution defining the train groups actually scheduled and the

order in which these train groups are scheduled; a set of Layer 2 sub-solutions, one for each scheduled train group g . Each Layer 2 sub-solution describes arrival/departure times, pass/stop events and used track in each station. Similar two-layer structures have been used in the literature for other problems [18], [22]. For the TTP, it mimics the real-world timetabling procedure by specialized planners.

Layer 1: train group selection

For finding a sub-solution in Layer 1, ants explore a directed graph. The set of nodes includes a source, a set of *schedule nodes*, and a set of *discard nodes*. A schedule node is used if the corresponding movable or optional train group is scheduled. The respective discard node is used otherwise. Edges are defined between all pairs of nodes belonging to different train groups, and between the source and all other nodes. A Layer 1 sub-solution is a clique of cardinality equal to the number of groups of trains.

Ants construct cliques inserting nodes progressively [20], starting from the source and adding at each step the decision of whether to schedule a train group.

As explained in Section III-A.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 an indicator called *overlap index*, whose detailed description is provided in [6]. It estimates how much the respective group would be in conflict with already scheduled ones, acting as a proxy for the TTT, EC, ST and CFL objectives.

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 [5], TEG nodes represent *discrete timetable events* for the first train of the group: the arrival (or departure) at (from) a station, at a time, at a track, with or without a stop. The existing nodes depend on the group time discretization step, the usable station tracks and pass/stop modes, and the bounds for minimum and maximum times (Section II). Dummy nodes stand for the begin and end of the path. Edges represent stops or passes at stations, or travels on line stretches. Running time and energy consumption are constrained by the characteristics of the infrastructure and of the rolling stock, and by the pass/stop modes used. The schedule of trains of the same group is strictly periodic: a path on the TEG that describes the timetable of the first train of a group defines the one of the whole group.

Each edge $e(u, v)$ of the TEG is characterized by the following set of heuristic information quantities, one for each objective $o \in O \setminus NTR$: $\eta_{(u,v),TTT}$ travel time associated to the edge, multiplied by the number of trains of the group; $\eta_{(u,v),EC}$ energy consumed by the trains of the group when using the edge, multiplied by the number of trains of the group (if nodes u and v refer to arrival and departure at the

same station, $\eta_{(u,v),EC} = 0$); $\eta_{(u,v),CFL}$ number of conflicts with already scheduled train groups generated by using the edge; $\eta_{(u,v),ST}$ minimum buffer time with already scheduled train groups due to the use of the edge.

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

In the *FAST exploration*, edges are progressively added by applying the pseudo-random proportional rule considering only candidates that do not generate conflicts, if any. If no such candidate exists, all neighbors of the current node are considered. In the pseudo-random proportional rule, the value of the heuristic information is the weighted sum of the quantities described above after normalization, considering weights λ_o^i as described in Section III-A.1.

In the *SMART exploration*, only nodes belonging to a low-conflict subset can be selected. To do so, we build a *restricted TEG* to be explored by ants, by applying a variation of Dijkstra algorithm to label the TEG nodes. Specifically, we consider two labels for each node, as the sum of η values over the edges of the minimum cost path reaching the labeled node. The first label is the *conflict distance* ($cflDist$): the sum of $\eta_{(u,v),CFL}$. The second is the *aggregated normalized distance* ($aggNormDist$): the weighted sum of $\eta_{(u,v),TTT}$, $\eta_{(u,v),EC}$, $\eta_{(u,v),CFL}$ and $\eta_{(u,v),ST}$, considering λ_o^i . All nodes with $cflDist$ larger than the end node's one are discarded. The remaining ones constitute the restricted TEG. The heuristic information used in the pseudo-random proportional rule is equal to the ratio between $aggNormDist$ of the final node of the edge and the number of edges separating it from the begin node along the minimum cost path.

In MOACO, we define two parameters n_{FAST} and n_{SMART} which state the number of consecutive iterations for which we use the FAST and the SMART exploration, respectively, starting with the latter.

3) *Local search*: After building a TTP solution, composed of one Layer 1 sub-solution $S1$ and several Layer 2 sub-solutions, one for each scheduled train group, we apply a local search to explore its neighborhood. It mainly aims at the reduction of the number of residual conflicts.

The neighborhood we consider includes solutions which schedule exactly the same train groups as in the Layer 1 sub-solution and in the same order, but with some different Layer 2 sub-solutions. In particular, we consider a portion $pcSol_{LS} \in [0, 1]$ of the scheduled nodes in $S1$. For each of them, we recompute the Layer 2 sub-solution equal to the path with minimum cost in terms of total value of $aggNormDist$ labels (Section III-A.2). Here, the conflict assessment for $\eta_{(u,v),CFL}$ is performed considering all trains of the timetable. We apply the local search for n_{LS} consecutive iterations, followed by n_{noLS} iterations in which we do not.

B. MILP formulation

All the solutions generated by MOACO are refined through a MILP formulation. It takes as input the list of scheduled train groups and their passing and stopping times. Here, conflicts are not allowed, hence the solutions produced are practically applicable. To eliminate residual conflicts and

optimize the multiple objective functions, the formulation slightly modifies train passing and stopping times, as well as station tracks. The used edge sequence cannot be modified.

With respect to MOACO, it drops temporal discretization and exploits the periodicity tolerance of trains of the same group. The objective function is the weighted sum of the normalized values in $[\tau_{min}, \tau_{MAX}]$ of three objectives: TTT , EC , ST . Indeed, the number of trains is constant, as the scheduled train groups are an input. The weights considered are those used by the ant that built the input solution.

The MILP formulation is modeled after those proposed in [16] and [19]. A detailed description is provided in [6] and is omitted here for lack of space. Binary variables control the use of tracks at stations and of specific passing/stopping events, 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 the input solution. The maximum allowed time modification is a parameter that we call *MILP degree of freedom*. By varying the value of continuous variables, the formulation can: shift a whole periodic train group by a maximum value equal to the degree of freedom; modify the schedule of individual trains provided that its group periodic pattern is relaxed within the tolerance permitted in the service concept.

IV. EXPERIMENTAL ANALYSIS

This section reports the details of the experimental analysis carried out to assess the ATMO framework. We aim to understand the contribution of different algorithmic components to the performance. We first present the test instances we consider. Then we discuss the ATMO assessment.

A. Test instances

Parameter 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 are realistic medium-sized assignments to individual timetable planners when partitioning the national network as part of a network-wide analysis. They cover 370 and 200 km of tracks, respectively, and are equipped with the conventional Norwegian signaling system, based on axel-counters. The number of timing location is 53 and 52, of which 32 and 30 stations. In average, stations have 2.3 available sidings in BB (excluding the 8-track Bergen station) and 2.4 in TRB (excluding the 13-track Trondheim station).

On each infrastructure, different instances are built using a reference daily traffic pattern, whose main characteristics are described in Figure 2 and Table I. For each considered train group we report: the number of trains in the two directions (separated by a comma); the periodicity of 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. Groups are named after their category: FR

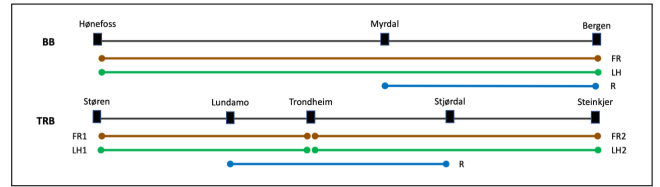


Fig. 2. Reference daily traffic patterns on the test infrastructures

TABLE I
MAIN CHARACTERISTICS OF THE BASE TRAFFIC PATTERNS

TRB					
Category	FR1	FR2	LH1	LH2	R
# trains in each direction	5, 5	5, 5	8, 7	6, 7	15, 16
Period (min)	-	-	-	60	60
# usable crossing stations	5	20	6	21	16
Of which mandatory stops	0	0	6	16	16
Max stop time (min)	60	60	8	8	5

BB			
Category	FR	LH	R
# trains in each direction	8, 7	5, 5	15, 15
Period (min)	-	60	60
# usable crossing stations	26	21	21
Of which mandatory stops	0	21	21
Max stop time (min)	60	8	5

for freight; LH for long-haul passenger; R for regional. In TRB, two types of freight and long-haul passenger trains are to be scheduled. They differ for their route, stopping pattern and type of rolling stock used. 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 sizes of the admissible time windows of train group timings in stations: 15, 30, 60 and 90 minutes (for the TRB line only, also 120 minutes);
- Six train configurations, in which 0 to 5 train groups (randomly chosen in the traffic patterns) are deleted.

The input data sets were prepared using the Treno suite, a railway timetable planning software [17] currently used by Jernbanedirektoratet.

All the 54 resulting instances are tackled considering the TTT and ST objectives. As no train group is optional, the maximization of the number of scheduled trains is omitted. As for energy consumption, we exclude it because input data are not currently available in the necessary level of detail. We run experiments on an Intel(R) Xeon(R) CPU E5-2637 v3 @ 3.50GHz with 16 CPUs and 128 GB ram.

B. Assessing the ATMO configuration

In this section, we assess different configurations of ATMO. 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.

1) *MOACO performance*: We apply the IRACE tool [13] to tune the 18 parameters of ATMO. A time limit of 500 s and a tuning budget of 5000 iterations are set. The hypervolume

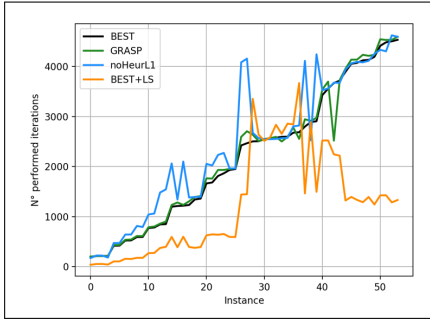


Fig. 3. Number of performed iterations within the 500 s time limit.

HV (to be maximized [9]) is chosen as the comparison indicator. With these settings, the tuning procedure takes almost 12 days. The details on this process are omitted for lack of space. They are available in [6]. The chosen settings are: 35 ants per colony; $\alpha_{L1} = 3$; $\beta_{L1} = 3$; $\rho_{L1} = 0.092$; $\tau_{min,L1} = 0.467$; $\tau_{MAX,L1} = 8$; $\alpha_{L2} = 2$; $\beta_{L2} = 8$; $\rho_{L2} = 0.07$; $\tau_{min,L2} = 0.618$; $\tau_{MAX,L2} = 12$; $n_{SMART} : n_{FAST} = 1:9$; $n_{LS} : n_{noLS} = 0:1$; $pcSol_{LS} = 25$; $N_{upd} = 3$. The tuning highlights that the alternative use of the smart and fast TEG exploration modes is the best option. 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 solution space is fostered. Finally, the local search does not significantly improve the performance of the algorithm, and the tuning decides to not exploit it.

To strengthen the tuning conclusions with respect to three crucial parameters, namely α , β and n_{LS} , we perform further comparisons between four MOACO configurations:

- *BEST*: The parameter configuration returned by the tuning;
- *GRASP*: Greedy Randomized Adaptive Search Procedure obtained from *BEST* by setting $\alpha = 0$, to exclude pheromone contribution to exploration;
- *noHeurL1*: *BEST* with $\beta = 0$, to avoid the costly computation of the heuristic information in Layer 1;
- *BEST+LS*: *BEST* with $n_{LS} : n_{noLS} = 1:9$ and $pcSol_{LS} = 25$, as in the best configuration returned by IRACE including local search.

For each configuration, we tackle all the 54 test instances, each of them with the same random seed. A time limit of 500 s is set. We then compare pairs of configurations by applying the Wilcoxon signed-rank test to the differences between the HVs returned by the different configurations.

Confirming the results of IRACE, the performance of *BEST* appears significantly better than the other configurations. The stronger evidence is in favor of the need of using the dynamic heuristic information in Layer 1, as the p-value returned from the test is the smallest in the comparison between *BEST* and *noHeurL1* ($9.918e - 06$). As for *GRASP* and *BEST+LS*, they lead to rather similar observations, with p-values of the order of 10^{-3} .

Figure 3 reports the number of iterations performed by the four configurations within the time limit. Solutions

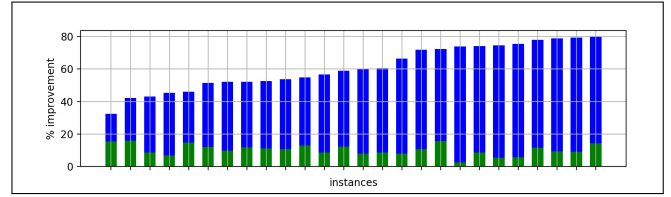


Fig. 4. Improvement in the values of objectives *TTT* (blue bars) and *ST* (green bars) produced by the MILP refinement

are ordered on the x-axis depending on the number of iterations performed by *BEST*. With respect to *BEST*, an average difference of +1.89%, +15.82% and -48.97% in the number of performed iterations is obtained for *GRASP*, *noHeurL1* and *BEST+LS* respectively. 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 the number of iterations, although indeed deactivating it in *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 *GRASP* does not translate in a remarkably different number of iterations performed.

2) *MILP refinement*: In the following, we discuss the results obtained when refining with the MILP formulation the timetables of the 54 POSs obtained with the *BEST* configuration. To this purpose, the MILP degree of freedom is set to 1200 s and the periodicity tolerance of all groups in all relevant locations is set to 600 s. MOACO is run until a 500 s time limit is met, and then the MILP formulation is solved with each timetable of the produced POS as an input. The MILP formulation is solved by the GUROBI 9.1 commercial solver, which stops when either a 2% MIP gap is met or when the MILP computation time exceeds 600 s.

The solutions of the POSs for the 54 instances include between 0 and 508 conflicts, that need to be solved by the MILP. In average, MOACO produces POSs of cardinality 12 (minimum 2, maximum 28).

For 41 of the 54 instances, the MILP formulation computes at least one conflict-free timetable (average 7, minimum 1 and maximum 15). 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 finds a feasible solution but stops after the 600 s without reaching the 2% MIP gap. 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.

Figure 4 describes the average percentage improvement in the objective function values produced by the MILP refinement, for the 26 instances for which at least one feasible (i.e., conflict-free) timetable is found. For each instance, the blue bar displays the improvement of the *TTT* objective, while

the green bar refers to the *ST* one. Instances are sorted along the x-axis according to the ascending order of the *TTT* objective improvement. The figure highlights how dramatic improvements are produced by the MILP formulation, thanks to the exploitation of periodicity tolerances. On the one hand, this is particularly evident for the *TTT* objective, for which the improvements range between 32% and 79%. On the other hand, improvements of the *ST* objective, while always present, are comprised within 2% and 16%. Since this dramatic difference is caused by a sharp decrease of the minimum buffer time in the resulting timetable, we conjecture that this is a consequence of the solution of residual conflicts. A conflict can be seen as a “negative buffer time” occurrence which, as such, is not considered for the computation of *ST* objective. As soon as the MILP formulation is capable to solve the conflict, the corresponding buffer time will be considered in the objective computation.

V. CONCLUSIONS

This paper proposed a novel algorithmic framework for automatic railway timetable generation, named ATMO. Thanks to its multi-objective approach, it provides the user with an approximated Pareto Optimal Set of timetables according to four different KPIs. The paper described how ATMO is composed by the integration of a novel MOACO algorithm and a MILP formulation.

In future research, we will work to further improve MOACO performance, and hence ATMO, pursuing various directions. The first one concerns the local search: we may exploit the MILP formulation refining ant solutions as a further local search, keeping in mind that the search spaces of MOACO and MILP are different. Further investigation will concern the balance between the computation times of MOACO and MILP, in order to pursue the best trade-off. Finally, alternative algorithmic approaches will be considered for both MOACO and MILP, to benchmark their performance.

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

REFERENCES

- [1] Bešinović N., Goverde R.M.P., 2018. Capacity Assessment in Railway Networks. In Borndörfer, Klug, Lamorgese, Mannino, Reuther, Schlechte (Eds.), Handbook of Optimisation in the Railway Industry. International Series in Operations Research & Management Science, 268, Springer, Cham (2018).
- [2] Borndörfer R., Klug T., Lamorgese L., Mannino C., Reuther M., Schlechte T., 2017. Recent success stories on integrated optimisation of railway systems, Transportation Research Part C: Emerging Technologies, Volume 74(2), 196-211.
- [3] Cacchiani V., Furini F., Kidd M., 2016. Approaches to a real-world train timetabling problem in a railway node. Omega 58, 97-110.
- [4] Caimi G., Kroon L., Liebchen C., 2017. Models for railway timetable optimisation: Applicability and applications in practice. Journal of Rail Transport Planning & Management, 6(4), 285-312.
- [5] Caprara A., Fischetti M., Toth P., 2002. Modeling and Solving the Train Timetabling Problem. Operations Research. 50. 851-861.
- [6] Coviello N., Medeossi G., Nash A., Nygreen T., Pellegrini P., Rodriguez J., 2021. Automatic generation of timetable with the ATMO tool. Technical report available at <https://www.trenolab.com/research/>.
- [7] Coviello N., Medeossi G., Nash A., Nygreen T., Pellegrini P., Rodriguez J., 2023. Multiobjective Timetable Development Tool for Railway Strategic Planning in Norway, Transportation Research Record, 2677(1), 720-729.
- [8] Dorigo M., 1992. Optimisation, learning and natural algorithms (in Italian). Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.
- [9] Fonseca C. M., Paquete L., López-Ibáñez M., 2006. An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. 2006 IEEE International Conference on Evolutionary Computation, 1157-1163.
- [10] Goverde R.M.P., Hansen I. A., 2013. Performance indicators for railway timetables. 2013 IEEE International Conference on Intelligent Rail Transportation Proceedings, Beijing, 2013, 301-306, doi: 10.1109/ICIRT.2013.6696312.
- [11] Hansen I.A., Pachl J., 2014. Railway timetabling & operations. Hamburg: Eurailpress.
- [12] Jordi J., Toletti A., Caimi G., Schüpbach, 2019. Applied Timetabling for Railways: Experiences with Several Solution Approaches. In RailNorrköping 2019. 8th International Conference on Railway Operations Modelling and Analysis (ICROMA), Norrköping, Sweden, June 17th–20th, 2019 (No. 069, pp. 462-470). Linköping University Electronic Press.
- [13] López-Ibáñez M., Dubois-Lacoste J., Stützle T., Birattari M., 2011. The IRACE package, Iterated Race for Automatic Algorithm Configuration. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université libre de Bruxelles, Belgium.
- [14] López-Ibáñez M., Stützle T., 2012. The automatic design of multi-objective ant colony optimisation algorithms. IEEE transactions on evolutionary computation, 16(10).
- [15] Lusby R.M., Larsen J., Bull S., 2018. A survey on robustness in railway planning. European Journal of Operational Research 266 (1), 1–15.
- [16] Mannino C., Lamorgese L., Piacentini M., 2015. Optimal train dispatching by Benders'-like reformulation. Transportation Science, 50(3), 763-1138.
- [17] Medeossi G., Nash A., 2020. Reducing Delays on High-Density Railway Lines: London–Shenfield Case Study, Transportation Research Record: Journal of the Transportation Research Board, 2674 (7), 193-205.
- [18] Nothegger C., Mayer A., Chwatal A., Raidl G. R., 2012. Solving the post enrolment course timetabling problem by ant colony optimisation. Annals of Operations Research volume 194, 325–339.
- [19] Pellegrini P., Marlière G., Pesenti R., Rodriguez J., 2015. RECIFE-MILP: An effective MILP-based heuristic for the real-time railway traffic management problem. IEEE Transactions on Intelligent Transportation Systems 16(5), 1-11.
- [20] Solnon C., Bridge D., 2006. An Ant Colony Optimisation Meta-Heuristic for Subset Selection Problems. In System Engineering using Particle Swarm Optimisation. Nova Science Publisher, 7-29.
- [21] Stützle T., Hoos H., 2000. MAX - MIN Ant System. In Journal of Future Generation Computer Systems, vol. 10, 889–914.
- [22] Yao B., Hu P., Zhang M., Tian X., 2014. Improved Ant Colony Optimisation for Seafood Product Delivery Routing Problem. PROMET - Traffic & Transportation. 26.