A two-stage framework for strategic railway timetabling based on multi-objective ant colony optimization

Nicola Coviello [a,1], Giorgio Medeossi [a], Thomas Nygreen [b], Paola Pellegrini [c], Joaquin Rodriguez [d]
[a] TrenoLab SRlS, Via Maniacco 7/A, 34170 Gorizia, Italy
[1] E-mail: n.coviello@trenolab.com, Phone: +39 0481 30031
[b] The Norwegian Railway Directorate, Postboks 16, NO-0101 Oslo, Norway
[c] COSYS-LEOST, Univ Gustave Eiffel, IFSTTAR, Univ Lille, F-59650 Villeneuve d'Ascq, France
[d] COSYS-ESTAS, Univ Gustave Eiffel, IFSTTAR, Univ Lille, F-59650 Villeneuve d'Ascq, France

Type B: Professional paper.

Abstract
This professional paper presents the algorithmic framework for automatic timetable generation which is actually under development within the research project "*Tools for mathematical optimization of strategic railway timetable models*" funded by the Norwegian Railway Directorate. This framework is composed by two main components, which tackle the Train Timetabling Problem in a macroscopic model of infrastructure and operations. The first stage is a Multi-Objective Ant Colony Optimization (MOACO) algorithm which produces a Pareto Optimal Set of solutions. These are timetables optimized with respect to four main objectives, namely: the total travel time, the total energy consumption, the timetable robustness and the total number of scheduled trains. The MOACO treats conflict constraints as soft ones and produces strictly periodic traffic patterns. The second stage is a MILP formulation tackled with a commercial solver. It refines timetables produced by MOACO, looking for improvements in the neighbourhood of the solutions provided by the latter. MILP solutions are conflict-free timetables and can profit of given tolerances on trains periodicity. The MILP stage is used as a final refinement of the results as well as an intermediate local search during MOACO iterations.

Keywords

Multi Objective, Ant Colony Optimization, Train Timetabling Problem, MILP

1 Introduction

This paper presents an algorithmic framework to perform strategic timetabling, 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 and Gustave Eiffel University. The aim of the project is to develop a prototype tool to automatically generate timetable drafts, to help planners perform tasks such as capacity studies and strategic timetable planning.

The use of strategic timetables is a key factor in customer-oriented railway network development. Based on the predicted demand for, or politically desired, transport capacity, Jernbanedirektoratet develops service concepts characterising the train services provided to the passengers and freight customers through stopping patterns, travel times, frequencies, distribution of departures throughout each day or week, and generic type of rolling stock. Strategic timetabling is the task of transforming these service concepts into feasible timetables on a matching infrastructure model.

This task requires looking at infrastructure capacity as a limited resource that can be exploited in different ways. Relevant literature (see for instance Goverde et al., 2016, and Yan et al., 2019) points out that capacity utilisation is effectively represented by sets of timetables representing an approximation of a Pareto-Optimal Set (POS) in the objective functions (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. From a practical point of view, it represents the best trade-offs between different Key Performance Indexes (KPIs).

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 as those based on an integer linear programming formulation tackled by commercial solvers cannot manage Paretian multi-objective optimization 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 optimization problem. They have proven to be effective in tackling combinatorial NP-hard problems as the TTP. On one hand, they can provide rather good solutions within a reasonable computation time. Furthermore, they can be easily extended to perform multi-objective optimization, and in particular to search for a POS of solutions. On the other hand, they often rely on sets of parameters that shall be carefully tuned and they cannot ensure the solution's optimality.

In our approach, we exploit the two perspectives, by combining an Ant Colony Optimization (ACO) algorithm and a Mixed Integer Linear Programming (MILP) formulation tackled by a commercial solver.

The paper is structured as follows: Section 2 introduces the input data model exploited by the algorithmic framework, as well as the outputs returned to the user. Section 3 presents the overall architecture of the algorithmic framework, while Section 4 focuses on the multi-objective ACO algorithm. Section 5 describes the role played by the MILP formulation. Finally, conclusions are drawn in Section 6.

2 Input and output

We adopt a macroscopic model for infrastructure and operations, based on a multigraph, where nodes are stations and edges line stretches connecting consecutive stations. Edges can be mono- or bi-directional, and more than one edge can connect the same pair of consecutive stations.

Infrastructure input data describe the rolling stock / infrastructure interaction: for each edge and rolling stock type, the minimum technical run times, the minimum energy consumption and the minimum headways are provided. A timetable is said to be conflict-free (in this model) when minimum headways are respected by all pairs of consecutive trains. Maximum energy consumptions and headways are computed for trains (pairs of trains, in case of headways) travelling at their minimum technical run time. The timetabling algorithm is let free to schedule train-paths with a run time greater than the minimum technical one. This would affect the relevant energy consumptions and headways. In reality they would vary in a non-linear way, mainly depending on the speed profile followed by trains. In our model, the relationship linking run times with energy consumptions and headways is modelled as a linear one, assuming that run times are let free to vary only within a small neighbourhood (+10 % max) of the minimum technical ones.

Timetable input data describe the train services that shall be scheduled in the resulting timetables. This information is provided for each train group, being a train group set of periodic trains. Three types of groups can be defined: fixed groups represent a mere constraint to the timetabling process; movable groups can be adjusted in time and space to optimize the resulting timetables; the same applies to optional groups, for which in addition the choice to schedule them or not can be performed by the algorithm.

Each group is qualified by a period, a number of courses, a priority factor and an admissibility region in time and space. The latter defines the path (in time-space), the available station tracks in each node, the minimum and maximum stop, arrival and departure times in each station. If time-admissibility allows some flexibility, we say that the group is *movable*. In each station on the group's journey, a periodicity tolerance is defined. It will allow the algorithm to schedule the group's courses with some deviation from the strictly periodic pattern. Spare courses are modelled as single-train groups.

To perform multi-objective optimization, we consider the following objectives:

1. Minimization of the total weighted travel time of trains;
2. Minimization of the total weighted energy consumption;
3. Maximization of the total weighted number of optional train groups actually scheduled in the resulting timetable;
4. Maximization of the timetable robustness.

All these objectives are computed by means of specific KPIs, calculated on the resulting timetables:

1. Travel times are trivially calculated as the difference between the arrival time at a train's last station and the departure time from the train's first station. The total travel time is the weighted sum of the trains' travel times, weights being proportional to train priorities.
2. Energy consumption strictly depends on the running times of trains in each infrastructure edges, 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 energy consumption is the weighted sum of the trains' energy consumptions, weights being proportional to train priorities.
3. The total weighted number of optional trains scheduled is trivially computed taking into account train priorities.
4. Robustness can be evaluated by means of several KPIs (see Goverde and Hansen, 2013). For a strong integration within the ACO architecture, we consider a KPI that can be incrementally computed during the solution construction. Specifically, we maximise the minimum buffer time in the timetable. A buffer time is the time separation between two feasible consecutive utilisations of the same infrastructure part (line stretch or station track) minus the minimum separation imposed between them (minimum technical headway).

Commentato [NC1]: 4. The multi-objective of the train timetabling problem should be formally formulized, especially because this paper is supported by the project related to the mathematical optimization. Most descriptions of the objectives are clear except the calculation of buffer time (might be easier to understand if it is expressed by formulations).

Commentato [NC2R1]: I'm sorry but I think that to add the formal equations would result in a useless prolongation of the paper, since it is not just a matter of adding 5 more formulae, but also to introduce and explain all the adopted formulation.

As an output, the framework returns the whole POS of timetables found, each of them characterized by the relevant values of the considered KPIs. Timetables can be exported in common formats (CSV, RailML) in order to be used in third-party editing and simulation tools.

3 Algorithm architecture

In principle, the ACO algorithm performs a broad exploration of the problem space, selecting a POS of "promising" solutions. They are further refined by the MILP formulation tackled by a commercial solver, which looks for improvements in the neighbourhood of the solutions provided by ACO.

Figure 1 displays the architecture of the algorithm. Grey ovals represent data contents, blue rectangles processes and orange diamonds conditional switches. Solid arrows mark the operations flows, while dashed ones data feedings. Starting from input data, the Multi-Objective ACO (MOACO) through multiple iterations maintains and updates a provisional POS. The double layer architecture exploited to construct MOACO solutions is described in Section 4. A Local Search (LS) procedure is then applied, which looks for improvements in a neighbourhood of the solutions provided by the pure ants' search. Improvements are obtained by removing the "noise" in the solution's quality produced by the stochastic behaviour of the ants' search, as recommended by literature sources (see for instance Samà et al., 2016). Finally, the best-so-far solutions in the provisional POS guide the ACO search during subsequent iterations via the pheromone trails.

The MILP formulation further improves these solutions. It plays a double role: first, it acts as a Local Search (LS) which refines some of the POS solutions during the MOACO search process. Alternatively, local search is performed by a simpler heuristic. The criterion for selecting the local search procedure is defined by the user. For example, MILP local search can be performed once every ten iterations. Second, after the MOACO algorithm stops according to a termination criterion, the MILP formulation refines all the solutions in the POS before their presentation to the user.

The MILP formulation deals with a model that is slightly different from the one considered by MOACO.

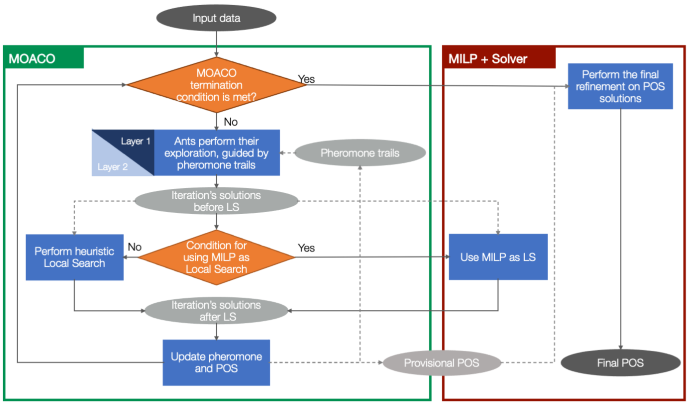

Figure 1. Global architecture of the algorithm.

On the one hand, conflict constraints (both in line stretches and in station tracks) are relaxed within the MOACO algorithm to improve its searching capability. In particular, MOACO uses an additional objective, i.e., the minimization of conflicts. As it is explained in Section 4, priority is accorded to conflict minimization with respect to the other objectives. Yet, some residual conflicts may remain in the MOACO timetables. They are solved (if possible) in the MILP stage, which considers conflict constraints as hard ones.

On the other hand, MOACO enforces rigid regular intervals between trains of the same train group. This permits to dramatically speed-up the exploration. However, it makes it impossible to profit of periodicity tolerances (defined as input data) to solve conflicts and improve objective function values. In the MILP formulation, train paths are free to vary exploiting the abovementioned tolerances.

4 MOACO for the TTP

In this section, we provide an overview of our original Ant Colony Optimization algorithm for the multi-objective automatic railway timetable generation. The proposed ACO is a multi-objective extension of the Max-Min Ant System algorithm. The designed ACO extension follows the guidelines proposed by López-Ibáñez and Stützle (2012) for multi-objective optimization.

In MOACO, we define a two-layer architecture. It applies to the TTP an approach already used for ACO applied to the Multi-Depot Vehicle Routing Problem (Yao et al. (2014)) and to the Course Timetabling Problem (Nothegger et al. (2012)). This two-layer architecture actually mimics the real-world timetabling procedure, performed mainly by hand by specialized timetable planners. Two basic actions can be identified in this procedure:

A1. The selection of the next train group to be scheduled. Here we also include the decision whether to schedule or not an optional train. Two main criteria may guide this choice:

a. The priority of each train group, with respect to the other ones. In principle a higher priority train should follow its ideal timetable path more than a lower priority one. A train scheduled before another one is likely to be designed closer to its optimal path since it is subject to fewer constraints.
b. In case of optional train groups, the estimation of existing conflicts with already scheduled ones. This may lead to the decision of not-scheduling a train group in case it is believed that it would not fit into an already populated timetable.

A2. The scheduling of a certain train group within a timetable already populated by previously set train-paths. The scheduling will actually define the arrival and departure times, as well as the utilised station track, of the train group in each station of its journey.

In principle, an action A1 (selection of the next train group to be scheduled) is always followed by an action A2 (scheduling of the selected train group). A2 is skipped when A1 chooses not to schedule a certain train group. In reality, timetable planners iteratively repeat pairs of A1-A2 actions.

Exploiting this architecture, we define two types of graph to be explored by artificial ants in MOACO.

A Layer 1 graph is associated to actions of type A1. They concern the decision of which group of trains actually figure in the timetable and in which order they are scheduled.

A set of Layer 2 graphs is associated to actions of type A2. These are *time-expanded directed graphs* (TEG). They refer to path schedule decisions, considering one group of trains in each graph, as proposed by De Fabris et al. (2014). TEG's nodes represent discrete timetable events of the first train of the group, i.e. the arrival (or departure) at (from) a certain station, at a certain time, at a certain station track, with or without a stop. The definition of discrete timetable events requires to discretize time. A 30 seconds or 1 minute time discretization is suitable to the TTP. TEG's edges represent transitions, in stations or line stretches. A station transition represents a train stopping or passing at a certain station, while a line transition represents a train travelling from a station to the following one. Two fictious nodes "START" and "END" are included into a TEG, and a path linking them completely describes the timetable of the first train of each group.

Figure 2 shows an example of TEG for a trains group stopping at three stations. In station 2 it can use two different tracks, while in stations 1 and 3 only the main one can be used. Arrival and departure events at station 2-track 2 are represented with a different degree of transparency, as well as edges linking them. Yet, they respect the same time discretization. The fictious START and END nodes are shown as well. More than one edge can exit from station 2 arrival nodes: this means that different stop times can be chosen by the algorithm. On the contrary, only one edge exit from each departure node, i.e. run times are fixed in this example. In case of a pass, besides of a stop, was admissible in station 2, other two sets of nodes (for both tracks 1 and 2) should figure in the TEG.

These alternatives are presented with pairs of arrival and departure nodes with different degrees of transparency. For each degree, different nodes correspond to different times. The edges connecting most departure nodes from Station 1 and arrival nodes of Station 2 are omitted for readability.

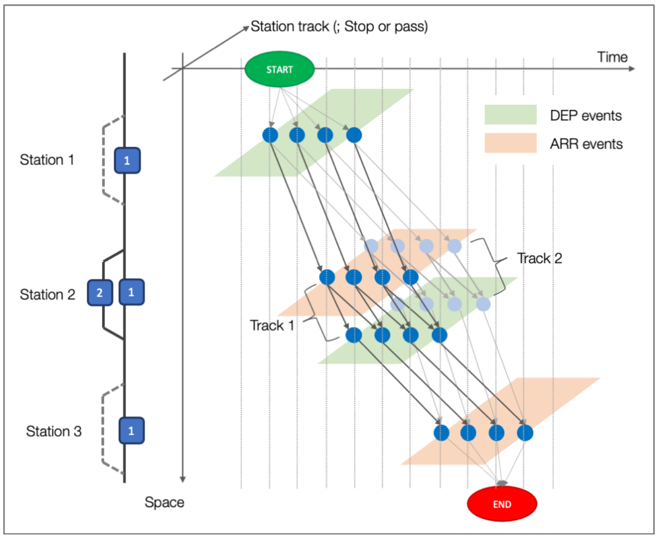

Figure 2. Example of a Layer 2 graph for a train group travelling through 3 stations.

Our ACO algorithm exploits this structure. We consider that a TTP solution (i.e. a railway timetable) is obtained by the combination of the following partial solutions (PS):

- A Layer 1 solution, which models the sequence of actions A1, defining:
  - Which trains actually figure in the resulting timetable;
  - The order in which these trains are scheduled.
- A set of Layer 2 solutions, one for each scheduled train. Each of them describes how a train is actually scheduled, in terms of arrival/departure times and used track in each station.

Nodes on the Layer 1 graph represent the decision to schedule or not a train. Directed edges indicate the reciprocal ordering of a pair of these decisions. A clique on the Layer 1 graph is a Layer 1 solution, and represents an ordered sequence of scheduling or not-scheduling actions.

Figure 3 shows an example of this graph. Here, four train groups are given as input. Group 2 is optional, so it presents both the “schedule” and “not-schedule” nodes. The bold clique on the graph shown that Group 4 is scheduled first, before Group 1, Group 2, which is not actually scheduled, and Group 3.

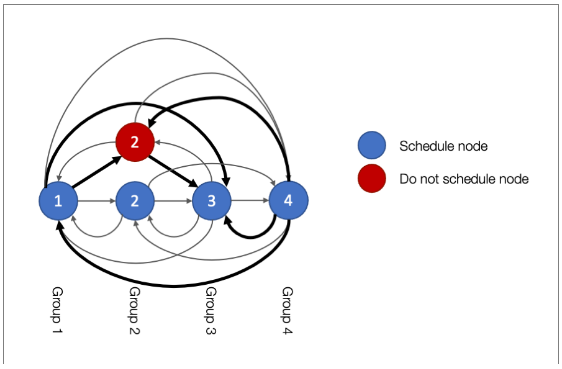

Figure 3. Example of a Layer 1 graph with 3 movable and 1 optional train groups. Bold edges represent a solution given by the sequence {S4, S1, NS2, S3}.

A path on the Layer 2 graph, i.e., a TEG path, is a Layer 2 solution. The TEG considered is the one corresponding to the first train of the group. Timetables of the other trains are directly defined considering the strict periodicity constraints (enforced during the ACO solution) between trains of the same group.

Each PS is constructed by the virtual ants by exploring a dedicated *construction graph*, with its own pheromone trail and heuristic information. Specifically, each ant starts choosing a Level 1 node, then, if this implies the schedule of a group, it builds a path on the corresponding Level 2 graph, before choosing the next Level 1 node. During the construction of the TEG path, interactions with already-built L2 solutions are taken into account. In particular, for a given group, the path is chosen among those that minimize (and possibly avoid) conflicts with already scheduled ones. Within this set, the path is constructed according to pheromone trail and heuristic information. This permits to prioritize the minimization of conflicts with respect to other objectives. This procedure continues until decisions are made for all groups at Level 1.

We model our MOACO after Max-Min Ant Systems (MMAS, Stützle and Hoos, 1997, 2000), an ACO variant which proved effective for a large number of different combinatorial problem. We implement a clique pheromonal strategy for Layer 1 (Solnon and Bridge, 2006), and a more classical path strategy for Layer 2. To the best of our knowledge, this hybridisation is a novel contribution to the field of ACO.

We extend the MMAS approach to the multi-objective variant following the multi-colony architecture proposed by López-Ibáñez and Stützle (2012). Each colony mostly focuses on one objective, and it uses its own pheromone matrices and set of aggregation weights to blend together pheromone and heuristic information (relevant to different objectives) during the ants' search. Colonies perform each iteration independently from each other. At the end of each iteration, the approximation of the Pareto Optimal Set is updated with the new solutions obtained. Previously obtained solutions of lower quality, are discarded. Finally, POS solutions update colonies' pheromone matrices. To this purpose, the POS is split into equal-cardinality non-disjoint subsets, in such a way that solutions in each subset are the best according to an objective. Each subset of solutions updates the pheromone of the colony which focuses on the objective in point.

The proposed MOACO features a remarkable number of parameters, which can hardly be tuned manually. The tuning will be performed by means of the IRACE tool (López-Ibáñez et al., 2011). It 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 advanced machine learning techniques.

5 The MILP formulation

As anticipated, the MILP formulation takes care of further refining the MOACO solutions. By exploiting the periodicity tolerance of trains belonging to the same group, and by thoroughly exploring the neighbourhood of MOACO solutions, it produces conflict-free solutions (according to the definition given in Section 2) and further improves their quality. To this purpose, time variable domains are tight intervals centred at the values belonging to the MOACO solutions. Hence, within these intervals:

- The whole periodic train groups can shift in time;
- Each single train of a group can change its station timing and track running time with respect to the group's strictly periodic pattern.

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

The MILP formulation is modelled after those proposed by Mannino et al. (2015) and Pellegrini et al. (2015). It is omitted here for lack of space. It minimizes the weighted sum of the objectives described in the introduction. The same set of aggregation weights which produced the original MOACO solution is used to this purpose. It is worthwhile to point out that, with this approach, not all the timetables obtained by refining the MOACO POS will belong to the new POS.

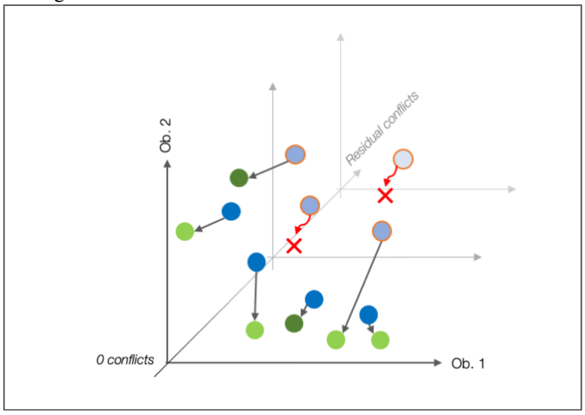

Figure 4. Graphical representation of the MILP operation in the objectives space.

Figure 4Figure 4 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 red-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 a conflict free timetable (red crosses). A new POS is originated by the MILP solutions. Light green points represent MILP solutions which belong to this new POS, while dark green points do not.

6 Conclusions

This paper presents an overview of an original framework for the automatic generation of railway timetables, performed on a macroscopic infrastructure and operation model. This framework is to be integrated into a tool prototype to be used by Norwegian railways for strategic timetabling.

The tool prototype will be a standalone executable, which will read the input dataset, invoke the timetabling algorithm, monitor the solution process and finally navigate, analyse and export solutions. A specifically-designed module of the tool will allow the user to edit, load and record custom configurations of the algorithm parameters, or to call an automated procedure based on the IRACE tool to perform a new algorithm tuning.

To overcome the complexity of creating timetables on a complete network, strategic timetabling projects normally partitions the network into regions, either studying one specific region, or developing regional strategic timetables which are then merged carefully. The algorithmic framework is currently being tested on a set of such regional instances, designed on the Trondheim and Bergen lines as well as to the eastern Oslo node, belonging to the Norwegian Railways. These three areas permit to define instances which cover a variety of operative conditions and with a scalable complexity in the input timetable data. This allows testing the algorithm performance in different application cases, representative of the real timetabling practice in Norway.

An extensive testing campaign is currently being carried out on these instances, benefitting of the feedback of experienced timetable planners from both Jernbanedirektoratet and TrenoLab, who analyse the timetables produced by the tool. Our preliminary results are promising, since they highlight how the proposed framework is capable to properly handle real-size instances and to provide in reasonable time timetables that are deemed meaningful and realistic. They show that a proper tuning of the algorithm parameters is crucial to ensure a good performance, making it necessary to resort to automatic tuning procedures.

As a mere example, a test has been performed on the Bergen railway, a 370 km long single-track line with more than 35 intermediate stations. Three infrastructure alternatives and 6 different service concepts (with up to 76 trains, of which 22 were optional ones) produced 18 different test scenarios. The process of calculating a conflict POS for all the scenarios (with a total of more than 250 produced timetables) was carried out in about 8.5 h.

Acknowledgements

The contents presented in this paper have been developed within the Research Project n° 202000783 *Verktøy for utvikling av matematisk optimering av strategiske rutemodeller* (Tools for mathematical optimization of strategic railway timetable models) promoted by the Norwegian Railway Directorate.

References

1. T. Stützle, H. Hoos, 1997. Improvements on the Ant System: Introducing the MAX-MIN Ant System. In G. Smith, N. Steele and R. Albrecht (eds.), Procs. of Artificial Neural Nets and Genetic Algorithms, pp.245–249.
2. T. Stützle, H. Hoos, 2000. MAX − MIN Ant System. In Journal of Future Generation Computer Systems, vol. 10, pp.889–914.
3. C. Solnon, D. Bridge, 2006. An Ant Colony Optimization Meta-Heuristic for Subset Selection Problems. In System Engineering using Particle Swarm Optimization, Nova Science Publisher, 7-29.
4. M. López-Ibáñez, J. Dubois-Lacoste, T. Stützle, M. Birattari, 2011. The irace package, Iterated Race for Automatic Algorithm Configuration. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université libre de Bruxelles, Belgium.
5. M. López-Ibáñez, T. Stützle, 2012. The automatic design of multiobjective ant colony optimization algorithms. IEEE transactions on evolutionary computation, 16(10).
6. C. Nothegger, A. Mayer, A. Chwatal, G. R. Raidl, 2012. Solving the post enrolment course timetabling problem by ant colony optimization. Annals of Operations Research volume 194: 325–339.
7. R.M.P. Goverde and I.A. Hansen, "Performance indicators for railway timetables," 2013 IEEE International Conference on Intelligent Rail Transportation Proceedings, Beijing, 2013, pp. 301-306, doi: 10.1109/ICIRT.2013.6696312.
8. S. De Fabris, G. Longo, G. Medeossi, R. Pesenti, 2014. Automatic generation of timetables based on a mesoscopic infrastructure model. Journal of Rail Transport Planning & Management, 4(2014), 2-13.
9. B. Yao, P. Hu, M. Zhang, X. Tian, 2014. Improved Ant Colony Optimization for Seafood Product Delivery Routing Problem. PROMET - Traffic&Transportation. 26. 10.7307/ptt.v26i1.1478.
10. P. Pellegrini, G. Marlière, R. Pesenti, J. Rodriguez, 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.
11. C. Mannino, L. Lamorgese, M. Piacentini, 2015. Optimal train dispatching by benders'-like reformulation. Transportation Science, 50(3): 763-1138.
12. M. Samà M., P. Pellegrini, A. D'Ariano, J. Rodriguez, 2016. Ant colony optimization for the real-time train routing selection problem. Transportation Research Part B, 85(3): 89-108.
13. R.M.P. Goverde, N. Bešinović, A. Binder, V. Cacchiani, E. Quaglietta, R. Roberti, P. Toth, 2016. A three-level framework for performance-based railway timetabling. Transportation Research Part C, 67: 62-83.
14. F. Yan, N. Bešinović, R.M.P. Goverde, 2019. Multi-objective periodic railway timetabling on dense heterogeneous railway corridors. Transportation Research Part B, 125: 52-75.