

Solving the Train Timetabling Problem, a mathematical model and a genetic algorithm solution approach

Diego Arenas, Rémy Chevrier, Said Hanafi, Joaquin Rodriguez

► **To cite this version:**

Diego Arenas, Rémy Chevrier, Said Hanafi, Joaquin Rodriguez. Solving the Train Timetabling Problem, a mathematical model and a genetic algorithm solution approach. 6th International Conference on Railway Operations Modelling and Analysis (RailTokyo2015), Mar 2015, Tokyo, Japan. <hal-01338609>

HAL Id: hal-01338609

<https://hal.archives-ouvertes.fr/hal-01338609>

Submitted on 28 Jun 2016

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.

Solving the Train Timetabling Problem, a mathematical model and a genetic algorithm solution approach

Diego Arenas ^{a,b,1}, Rémy Chevrier ^c, Saïd Hanafi ^b, Joaquin Rodriguez ^a

^a ESTAS, Univ. Lille Nord-de-France - IFSTTAR

¹ 20, rue Élisée Reclus, 59666 Villeneuve d'Ascq Cedex, France

E-mail: diego.arenas@ifsttar.fr

^b LAMIH, Université de Valenciennes

^c Innovation et Recherche, SNCF

Abstract

The construction of a timetable represents a critical part on the development of the yearly service plan for railway operations. The Train Timetabling Problem (TTP) aims to find a timetable that must respond both to commercial needs and certain capacity and security related constraints. The combination of the latter, makes the TTP a complex and time consuming process. While most approaches on the literature offers exact solving models, once they are applied on real-size instances, they fail to solve the problem within a reasonable amount of time. Reason for which, heuristic or relaxation techniques are extensively used. In this paper, we first propose an alternative mathematical model to tackle the TTP. Next, we present a Genetic Algorithm implementing our model in order to rapidly obtain near-optimal train timetables. Finally, we test the implementation of our model on a case study based on the German railway network.

Keywords

Timetabling, TTP, Train Scheduling, Genetic Algorithms

1 Introduction

A train timetable defines the set of departure and arrival times for the train lines at the stations or other relevant locations in the rail network.

The construction of the train timetable, or train timetabling problem (TTP), represents only the first step of an even larger and more complex process: the yearly rail service planning. As its name suggests it, it consists on the definition of the plan, and the allocation of resources to provide the annual train services, i.e., timetables, crew schedules, rolling stock usage, etc. It is an iterative process where the output of one step is used as input to the next one, and when a conflict appears and can not be solved on the current level, one must go back to the previous step to make the necessary adjustments. The process is described in Figure 1 and briefly summarized on the next paragraph.

Once a feasible timetable is obtained, a route plan has to be designed. This is known as the routing and platforming problems, and consists on the allocation of platforms, inbound and outbound routes for the scheduled trains into the stations. The next step is the rolling stock planning, which specifies among other things, which engines and carriages will be used for each scheduled train of the timetable. Finally, the crew schedule, which defines the

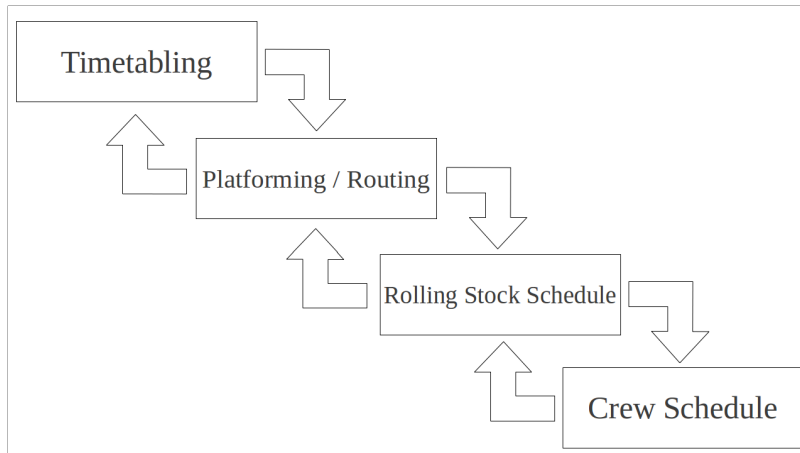


Figure 1: The Yearly Rail Service Planning

working plan for each person (role) involved in the train operations, e.g. drivers, controllers, etc.

The timetable must respond to the commercial requirements of the customers, both passengers and freight traffic, while respecting also some security and capacity constraints. These constraints are determined by the characteristics of the rolling stock and the infrastructure of the network. Typically, these constraints are called hard constraints, which mean that they can not be violated, e.g., minimum running time, minimum headway time, maximum number of platforms at some station etc. While some kind of commercial requirements and quality aspects are called soft constraints, their observance is not mandatory, but their transgression often leads to economic costs or reduction of overall quality of the service, e.g., key time, connections, etc.

The combination of the requirements and constraints as well as the large number of trains and journeys to schedule, makes this problem very difficult to solve. The train timetabling problem (TTP) is proven to be NP-Hard by Caprara et al. [6].

Furthermore, some of the commercial requirements may come into conflict with capacity constraints or with the maintenance activities of the network. These conflicts should be rapidly identified, since, in most of the cases, a negotiation between the train operating companies and the railway infrastructure manager is required, leading to further time consumption. By Taking this into account, we observe that the preparation of a yearly timetable is a complex, costly and time consuming process that usually takes several months to be fully completed.

Traditionally, the timetabling process was undertaken by the experts planners of the railway service. However, since the nineties, the benefits of using optimization models and techniques to solve the TTP were recognized and applied. The majority of these approaches are based on a mathematical formulation of the problem that could give exact solutions to the problem. However, due to its high complexity and impractical computing time, meta-heuristics and relaxation techniques are applied to solve the model.

The contribution of this work is twofold. First, we propose a mathematical formulation of the TTP which can be implemented using different optimization techniques, such as lin-

ear programming or meta-heuristics. Second, we develop a genetic algorithm implementing our model in order to rapidly obtain optimal or near-optimal train timetables.

The rest of the paper is structured as follows. Section 2 reviews some related work. Section 3 defines the TTP and presents our proposed formulation. Section 4 describes the implementation of our genetic algorithm to solve the TTP. Section 5 presents the case study and discuss the results of experimentation. Finally, Section 6 concludes the paper and present some perspectives for future work.

2 Related Work

Several approaches to tackle the TTP can be found on the literature, for a comprehensive survey of these, review Assad [2], Cordeau et al. [7], Cacchiani and Toth [5].

On this section we will review some of the contributions for the non-periodic train timetabling problem, however, relevant references on the periodic timetabling field can be found on: [16], [13], [15] [1] and [19]. Most of them are based on the Periodic Event Scheduling Problem (PESP), initially proposed by Serafini and Ukovich [18].

One of the first contributions to solve the train scheduling or timetabling problems was made by Szpigel [20], he modelled the single track train scheduling problem as a job shop scheduling problem, and solved it using a branch and bound algorithm.

Higgins et al. [10] propose a mathematical programming model to schedule trains on a single track line, based on the Australian rail network. On this work, priorities are assigned to each train, which then are used to find optimal solutions using a branch and bound procedure.

In a subsequent work, Higgins et al. [11] tackle the single line train scheduling by implementing and comparing the results of different heuristics methods: local search heuristic with an improved neighbourhood structure, genetic algorithms, tabu search and two hybrid algorithms.

Oliveira et al. [17] model the single-track railway scheduling problem as a special case of the job shop scheduling problem by considering the train trips as jobs, which will be scheduled on tracks regarded as resources. Then they show how the problem can be modelled with constraint programming and use Ilog Scheduler to solve it.

Caprara et al. [6] focus on a particular case of the TTP: a single, one-way track linking two major stations, with a number of intermediate stations in between. They propose a graph theoretic formulation used to derive an integer linear programming model which is relaxed in a Lagrangian way.

Kwan and Mistry [14] use a co-evolutionary algorithm for the automatic generation of train timetables. The objectives are: first, to allocate as many capacity requests as possible, and second: to discover the conflicts that have to be solved by the train operating companies. They use three types of populations which are evolved one after another. The combination of three individuals, i.e., one of each population, result on a timetable. Within each step of the evolution, a timetable is generated and evaluated. Once the termination condition is achieved, the algorithm is stopped and the best timetable is given as result.

Borndörfer and Schlechte [3] present two types of integer programming formulations: a standard formulation that models block conflicts in terms of packing constraints, and a novel formulation of the extended type that is based on additional configuration variables. Both formulations are then relaxed and solved using MIP solvers.

Fischer et al. [9] use a lagrangian relaxation of the conflict constraints combined with a

cutting plane approach to deal with very large instances of the train timetabling problem.

Tormos et al. [21] use a genetic algorithm to improve timetable robustness, by assigning time supplements in running and headway times over an optimized timetable obtained with the methods described on Ingolotti et al. [12].

Wong et al. [22] present a mixed integer programming model for generating synchronized timetables, which enable smooth connections while minimizing delays for passengers, the model is then solved using a heuristic technique.

Cacchiani et al. [4] propose a different formulation of the TTP, where each variable corresponds to a full timetable for a train, whereas, in classic formulations, where each variable is associated with a departure and/or arrival time of a train at some specific station. This approach leads to an easier and faster relaxation, thus, better and faster resulting timetables.

3 Train Timetabling Problem: definition and formulation

3.1 Problem definition

We define the TTP as: given a railway infrastructure, a list of rolling stock available and a list of train journeys to schedule, produce a timetable in which, all scheduled train journeys respect both capacity and security constraints while optimizing a given criteria.

Three important inputs are thus identified: the rolling stock, the railway infrastructure and the journey requirements. They are detailed below.

The rolling stock: On a real scenario, different types of trains circulate on the railway infrastructure. Each one of them has specific characteristics such as: speed, length, braking distance, etc.

The Infrastructure: The description of the infrastructure can be extremely detailed, however, for train timetabling purposes, only a macro perspective is used, this because higher details implies a more complex model and thus, larger processing time. For instance: several approaches sees stations as black boxes, which means that they do not deal with the stations' capacity, i.e. number of tracks for manoeuvres and platforms, this could lead to infeasible timetables which will have to be fixed later. In our model, we define three main elements to represent the infrastructure:

- **Tracks:** Tracks connecting two stations, each track definition contains important information as: if the track is single or both ways, the minimum running time for each train type that can circulate on it, the minimum headway time between two train types, among others.
- **Knot:** A knot defines the locations where trains are allowed to stop, dwell, overtake and perform other manoeuvres. They are typically stations and sidings. Information about the geographical location of the tracks connecting one knot to another is also defined, this allows to determine if a train must perform a turnaround maneuver before pursuing with its itinerary. All knot definitions are composed by a set of Inner Tracks.
- **Inner tracks:** These are the tracks or platforms inside a knot, they define the capacity of the knot.

The Journey Requirements: A set of train journeys to be scheduled on the timetable. Each journey J consists of a set of trips K_j . A trip is defined as the circulation of the train from one knot to another, and it has associated minimal, maximal and ideal departure

and arrival times. Some journeys are defined as mandatory, which means that they must be present on the generated timetable, while others will be scheduled only if it is possible. Each journey has a profit value associated, which is used to prioritize the schedule of some train journey over the others.

3.2 Definition of data and variables

Data:

J : Set of required train journeys.

$prof_j$: Profit value of a journey j .

K_j : Set of trips for a journey j .

$idealDep/idealArr_{j,k}$: Ideal departure and arrival times, trip k of journey j .

$lbDep/ubDep_{j,k}$: Lower and Upper bounds for the departure time, trip k of journey j .

$lbArr/ubArr_{j,k}$: Lower and Upper bounds for arrival time, trip k of journey j .

$minRT/maxRT_{j,k,k+1}$: Min/Max running time on trip k of journey j .

$minDT/maxDT_{j,k,k+1}$: Min/Max dwelling time between trips k and $k+1$ of journey j .

$minPT/maxPT_{j,k,k+1}$: Min/Max passing time between trips k and $k+1$ of journey j .

$minTT/maxTT_{j,k,k+1}$: Min/Max turnaround time between trips k and $k+1$ of journey j .

$minHT/maxHT_{j,k,j',k'}$: Min/Max headway time between train journeys j and j' on trips k and k' respectively.

Variables:

Dep : Departure Time for trip k of train journey j .

$$lbDep_{j,k} \leq Dep_{j,k} \leq ubDep_{j,k} \quad (1)$$

Arr : Arrival Time for trip k of train journey j .

$$lbArr_{j,k} \leq Arr_{j,k} \leq ubArr_{j,k} \quad (2)$$

$sched$: Binary variable that is used to determine if a train journey j is scheduled (is part of the timetable) or not.

$$0 \leq sched_j \leq 1 \quad (3)$$

3.3 Objective function

We define our objective function as the maximisation of the the global profit of the generated timetable, i.e., the sum of the profit value for each train scheduled ($sched_j = 1$). The profit value of a train is proportionally decreased when the train is scheduled to depart/arrive before or after its desired departure/arrival times. However, such variations are often necessary in order to comply with the constraints defined on the next sub section. In case of any constraint transgression, the associated train is not scheduled ($sched_j = 0$). Equation 4 presents a simplified version of the objective function

$$Maximize \sum_{j=1}^J sched_j \cdot prof_j - \sum_{k=1}^{K_j} penaltyArr_{j,k} + penaltyDep_{j,k} \quad (4)$$

3.4 Constraints

As stated before, constraints can be classified into two groups, soft and hard constraints. In our formulation, soft constraints are included into the objective function, i.e. the penalization over the profit generated when a train is not scheduled to arrive or depart at the ideal desired times. Hard constraints must be always respected by all scheduled trains, they are defined as follows.

- **Running Time:** Minimal and maximal running time for a train journey j on trip k . These values take into account the speed, the acceleration and braking times of the train. Therefore, the running time vary depending if the train has to stop at the next knot or if it will only pass through. Hence, four types of minimal running times are predefined: stop-stop, stop-pass, pass-stop, pass-pass. The choice of which minimal running time is used is determined by the particular itinerary of the train journey. Finally, these values also include a predefined buffer time to improve the robustness of the timetable.

$$\min RT_{j,k} \leq Arr_{j,k} - Dep_{j,k} \leq \max RT_{j,k} \quad (5)$$

- **Dwelling Time:** Minimal and maximal dwelling time for a train j that stops at a knot between trips k and $k + 1$. The dwelling time must be sufficient to allow passengers to disembark and board the trains. Additional time might be needed in case of connections between train journeys.

$$\min DT_{j,k,k+1} \leq Dep_{j,k+1} - Arr_{j,k} \leq \max DT_{j,k,k+1} \quad (6)$$

- **Passing Time:** The minimal and maximal time that takes a train to pass trough a knot between trips k and $k + 1$. Normally, the speed in which a train passes through a station is reduced, and some stations are very large, therefore, this time should not be disregarded.

$$\min PT_{j,k,k+1} \leq Dep_{j,k+1} - Arr_{j,k} \leq \max PT_{j,k,k+1} \quad (7)$$

- **Turnaround Time:** The minimal and maximal time that takes a train to perform a turnaround maneuver at knot between trips k and $k + 1$. A turnaround occurs when the concerned train arrives in a track that is located on some side of the station and it departs in other track that is located on the same side of the station.

$$\min TT_{j,k,k+1} \leq Dep_{j,k+1} - Arr_{j,k} \leq \max TT_{j,k,k+1} \quad (8)$$

- **Headway Time:** The minimal and maximal time that must exist between the departure time of two trains j and j' leaving from the same knot.

$$\min HT_{j,k,j',k'} \leq Dep_{j',k'} - Dep_{j,k} \leq \max HT_{j,k,j',k'} \quad (9)$$

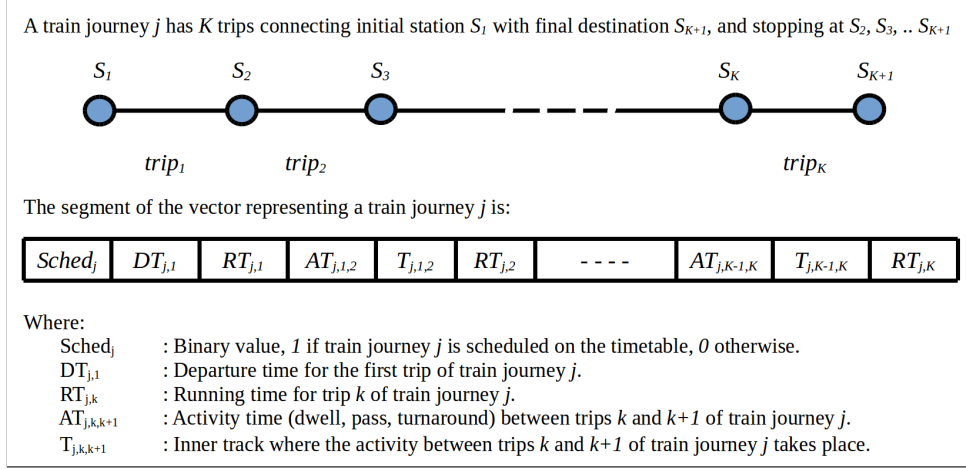


Figure 2: Data representation: Example of a segment

4 Solving the TTP using Genetic Algorithms

Due to the NP-Hardness nature of the problem, heuristic methods are extensively used to deal with real world instances of the TTP.

Genetic Algorithms (GA) have been successfully used for solving different types of planning, scheduling and optimization problems. They are able to handle very large and complex search spaces in a reduced amount of time. They perform a multi-directional stochastic search on the complete search space, which is intensified in the most promising areas.

This population based technique consist of a set of individuals, each individual represents a possible solution to the problem, or a part of it. In order to determine how good (or bad) a solution is, each individual is evaluated with a fitness function, which assigns fitness values to them. The core of the algorithm is the evolution of the population stage: an iterative process in which, best individuals are selected and combined between them (using variation operators) to form new individuals that will conform a new generation of the population. This process is repeated until certain condition is reached, e.g. number of generations, time limit, etc.

4.1 Data Representation

In our GA implementation, each individual represents a complete timetable. To do so, we use one vector of integer values. The length of the vector is equal to $\sum_{j=1}^J 3 \cdot |K_j|$.

The vector is divided into small segments, each segment corresponding to a train journey, inside one segment, the first element represents the $sched_j$ variable, which determines if the train journey is scheduled or not on the timetable. The next element is the departure time of the train journey at its first station ($Dep_{j,1}$). The next element represents the running time of the first trip. The next two elements designs the activity time and the inner track used by the train at the knot between the first and the second trip. The rest of the elements

follow the same pattern, Figure 2 gives an example of a segment. Each element of the vector, only allows values between those defined by its respective constraint. This ensures that every timetable generated will automatically respect all running, dwelling, passing and turnaround time constraints.

4.2 Algorithm and Operators

Algorithm 1 Genetic Algorithm for solving the TTP

```

1: Generate initial population  $P$  using a greedy algorithm
2: while Termination conditions not met do
3:   for each individual  $i$  of  $P$  do
4:     Generate a timetable using individual  $i$ 
5:     Evaluate the timetable using the fitness function.
6:   end for
7:   Apply variation operators to generate new offspring population  $P'$ 
8:   Combine  $P$  and  $P'$  to generate new population  $P$ 
9: end while

```

The main algorithm is detailed in Algorithm 1. The essential components are briefly described next:

- **Generate initial population:** The initial population of a GA, in the vast majority of cases, have a great impact on the overall performance of the algorithm, and this case is not the exception. Since randomly generated initial populations led to relatively poor performance, we implemented a greedy algorithm that generates an initial population of solutions that maximizes the sum of profit for all scheduled train journeys. Train journeys are randomly ordered and are scheduled one by one if they do not cause any hard constraint violation.
- **Generate a timetable using an individual:** Having the initial departure time, and the subsequent running and activity times of each trip and knot of the train journey, is possible to calculate the departure and arrival time for each trip, and thus, the complete timetable for all scheduled journeys.
- **Evaluate the timetable using the fitness function:** The fitness function is an implementation of the objective function described on Equation 4 with additional penalization when hard constraints are violated. It is important to note that even if a train journey appears as not scheduled on the timetable, because it violates at least one hard constraint, it affects negatively the fitness value of the individual.
- **Variation Operators:** Typical crossover and mutation operators were implemented. Single and multiple point crossover techniques were taken into account.
- **Combination of the current and offspring populations:** We have opted for an elitist schema for the populations combination, this means that the best individuals form the parents and offspring populations are selected to conform the new population.

5 Experimentation

5.1 Case Study description

For experimentation we used a case study proposed by Erol et al. [8]. The study case consists of 3 train types, and an artificial network with 15 stations, 90 station capacities (tracks inside stations) and 42 tracks, leading to 126 running and 378 headway times. Several requests sets are defined, consisting of a different number of train journeys and time horizons.

5.2 Experiment Details

The specifications of the computer used to solve the Case study are: Processor Intel Core i7 @2.9 GHz x4, 3.5 GB of RAM memory and running Ubuntu-64 v12.10 as operating system.

Additionally to our genetic algorithm, for benchmarking the results, we implemented our model using a MIP solver: CPLEX v 12.6. We set the execution time limit for the solver to 1 hour.

5.3 Experimental Results

Different requests sets based on the infrastructure described on subsection 5.1 were solved using both our genetic algorithm implementation and the MIP solver. The results are summarized on Table 1.

As many others metaheuristic techniques, GA have a stochastic nature, thus, different executions of the algorithm having the same input may result on different outputs. In order to assess the average performance of the GA, a total of 40 program executions were performed for each request set.

Table 1: Results of experimentation

Req. Journeys	Time Horizon	Const. Count	CPLEX Profit	CPLEX Time	Max. Eval.	GA Profit	GA Time
8	45	171	3500	0.05 s	1.5 K	3500	0.08 s
11	77	247	4226.13	0.73 s	11 K	4193.66	2.43 s
14	72	340	5789.36	0.15 s	10 K	5783.67	3.69 s
15	22	409	5467.72	1.03 s	16 K	5456.06	5.91 s
46	86	3524	16949.39	1 m 6 s	10 K	15294.79	28.23 s
55	79	4721	18009.22	22 m 19 s	55 K	17460.15	3 m 44 s
106	55	17536	25644	1 h	70 K	23508.35	17 m 13 s

We observe that for a small number of requests, our GA was outperformed by the CPLEX solver. Considering 8 train requests, even though the optimal value is attained by the GA, it was slightly slower than the solver. As for 11, 14 and 15 train requests, on average, our GA obtained a near optimal value was but considerably slower than CPLEX. However, as the instance grows in complexity, the advantages of the GA become more evident. On the last 3 request sets, the GA found near optimal solutions on a considerable reduced amount of processing time compared to the MIP solver.

These results show the potential of genetic algorithms to deal with large, real size instances of the TTP. However, we believe that these results could be further improved. In many cases, the genetic algorithm converged prematurely, which means that only local optimal solutions were found. To deal with this, some adjustments to the mutation rate were made, however this action is not enough to effectively deal with this issue. We shall consider to implement a different selection method that would maintain a diverse population of solutions.

6 Conclusions

The Train Timetabling Problem (TTP) represents a complex and interesting research subject, research on this area should optimize the usage of railway infrastructure while respecting security issues, boost the profit generated and increase timetable robustness and connectivity.

On this paper, we present a mathematical formulation to tackle the TTP, our model takes into account macro-level representations of both infrastructure and rolling stock elements to optimize the train journey scheduling based on the global profit generated. Then, we present a genetic algorithm implementation of our model that we use to solve the TTP.

The experimental results of our genetic algorithm are very encouraging. Despite the high complexity and NP-hardness nature of the problem, the performance of the GA is very satisfactory. Near-optimal results were achieved within short computation times. Moreover, our model proved to be very flexible, it could be tuned to focus on other particular optimization aspect such as the reduction of total waiting time.

Further extension and improvements to our work include : First, to experiment with different optimization criteria, such as improving connectivity or robustness of the final timetable, or reducing the rolling stock needed or energy consumption. The purpose of this is to develop a compound multi criteria objective function that reflects better the optimization needs of the timetabling process.

Even if our mathematical model can find the optimal timetable for a given instance of the problem, on real size scenarios, the required computing time to achieve optimality would be impractical. Thus, heuristic techniques must be always adopted. In this sense, regardless the good performance of the GA, other approaches should be also explored, therefore, as a second extension of this work, we would like to implement and experiment with other heuristic techniques in order to compare their performance.

Finally, we are also considering to combine our genetic algorithm with other optimization technique: either exact, such as mathematical programming, or another heuristic, e.g. local search, to create a hybrid method. The objective of this is, not only to overcome the limitations of the genetic algorithms, e.g., a tendency to converge towards local optimal solutions, but also to take advantage of the strengths of both methods.

Acknowledgements

The research leading to this paper and Phd.Thesis has received funding from the Institute de Recherche Technologique RAILENIUM. Espace Technopôle, Le Mont Houy, Route départementale 958, 59300 FAMARS, France.

References

- [1] D. Arenas, R. Chevrier, J. Rodriguez, C. D’haenens, and S. Hanafi. Application of a co-evolutionary genetic algorithm to solve the periodic railway timetabling problem. In *Industrial Engineering and Systems Management (IESM), Proceedings of 2013 International Conference on*, pages 1–7, Oct 2013.
- [2] A. A. Assad. Models for rail transportation. *Transportation Research Part A: General*, 14(3):205 – 220, 1980.
- [3] R. Borndörfer and T. Schlechte. Models for railway track allocation. In Christian Liebchen, Ravindra K. Ahuja, and Juan A. Mesa, editors, *7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS’07)*, volume 7 of *OpenAccess Series in Informatics (OASICs)*, Dagstuhl, Germany, 2007. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [4] V. Cacchiani, A. Caprara, and P. Toth. A column generation approach to train timetabling on a corridor. *4OR*, 6(2):125–142, 2008.
- [5] V. Cacchiani and P. Toth. Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727 – 737, 2012. Feature Clusters.
- [6] A. Caprara, M. Fischetti, and P. Toth. Modeling and solving the train timetabling problem. *Operations Research*, 50(5):851–861, 2002.
- [7] J.-F. Cordeau, P. Toth, and D. Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32(4):380–404, 1998.
- [8] B. Erol, M. Klemenz, T. Schlechte, S. Schultz, and A. Tanner. Ttplib 2008 - a library for train timetabling problems. Technical Report 08-19, ZIB, Takustr.7, 14195 Berlin, 2008.
- [9] F. Fischer, C. Helmberg, J. Janßen, and B. Krostitz. Towards solving very large scale train timetabling problems by lagrangian relaxation. In Matteo Fischetti and Peter Widmayer, editors, *8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS’08)*, volume 9 of *OpenAccess Series in Informatics (OASICs)*, Dagstuhl, Germany, 2008. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [10] A. Higgins, E. Kozan, and L. Ferreira. Optimal scheduling of trains on a single line track. *Transportation Research Part B: Methodological*, 30(2):147 – 161, 1996.
- [11] A. Higgins, E. Kozan, and L. Ferreira. Heuristic techniques for single line train scheduling. *Journal of Heuristics*, 3(1):43–62, 1997.
- [12] L. Ingolotti, A. Lova, F. Barber, P. Tormos, M.A. Salido, and M. Abril. New heuristics to solve the csop railway timetabling problem. In Moonis Ali and Richard Dapoigny, editors, *Advances in Applied Artificial Intelligence*, volume 4031 of *Lecture Notes in Computer Science*, pages 400–409. Springer Berlin Heidelberg, 2006.

- [13] L. G. Kroon, R. Dekker, and M. J. C. M. Vromans. Cyclic railway timetabling: A stochastic optimization approach. In Frank Geraets, Leo Kroon, Anita Schoebel, Dorothea Wagner, and Christos D. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes in Computer Science*, pages 41–66. Springer Berlin Heidelberg, 2007.
- [14] R.S.K. Kwan and P. Mistry. A co-evolutionary algorithm for train timetabling. Technical report, University of Leeds, July 2003.
- [15] C. Liebchen, M. Proksch, and F. H. Wagner. Performance of algorithms for periodic timetable optimization. In Mark Hickman, Pitu Mirchandani, and Stefan Vo, editors, *Computer-aided Systems in Public Transport*, volume 600 of *Lecture Notes in Economics and Mathematical Systems*, pages 151–180. Springer Berlin Heidelberg, 2008.
- [16] M. A. Odijk. A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research Part B: Methodological*, 30(6):455 – 464, 1996.
- [17] E. Oliveira and B. M. Smith. A job-shop scheduling model for the single-track railway scheduling problem. Technical Report 2000.21, University of Leeds, [HTTP://WWW.OPENTRACK.CH](http://www.opentrack.ch), 2000.
- [18] P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM J. Discret. Math.*, 2(4):550–581, November 1989.
- [19] M. Siebert and M. Goerigk. An experimental comparison of periodic timetabling models. *Computers & Operations Research*, 40(10):2251 – 2259, 2013.
- [20] B. Szpigel. Optimal train scheduling on a single track railway. *Operational Research '72*, pages 343–351, 1972.
- [21] P. Tormos, A. Lova, L. Ingolotti, and F. Barber. A genetic approach to robust train timetabling. Technical report, ARRIVAL Project, 2008.
- [22] R. C. W. Wong, T. W. Y. Yuen, K. W. Fung, and J. M. Y. Leung. Optimizing timetable synchronization for rail mass transit. *Transportation Science*, 42(1):57–69, February 2008.