On the train timetabling problem and heuristics

Eggo Pinheiro
Computer Science Program, Federal University of Maranhão (UFMA)
Portugueses Av 1966, São Luís MA Brazil
pinheiroeggo@gmail.com

Enrico Miranda
Computer Science Program, Federal University of Maranhão (UFMA)
Portugueses Av 1966, São Luís MA Brazil
enrico.s.miranda@gmail.com

Alexandre C. M. Oliveira
Department of Informatics, Federal University of Maranhão (UFMA)
Portugueses Av 1966, São Luís MA Brazil
acmo@deinf.ufma.br

ABSTRACT

Freight railways are the major means of transportation of bulk material, such as iron ore from the origin to the destination. Usually for heavy haul railways, the detination is a port. To improve profit and ensure the quality of services of a railway, a good timetable design is crucial. However, all the work provided on the Train Timetable Problem, an NP-hard problem, is usually only applied locally to a single railway. This article deals with the train timetabling problem applied to mixed traffic railways with both cargo trains and passenger trains sharing the same resources with different priorities. Moreover, a public base benchmark of problems is proposed so that future work can be compared. This base is used to show that the problem cannot be solved to optimality within a reasonable time frame for production sized instances, and a genetic algorithm is proposed.

KEYWORDS. Genetic Algorithm. Operational Research. Heuristic. Train Timetabling Problem.

1. Introduction

There are several layers composing the Rail transportation planning: demand analysis, line planning, train scheduling, rolling stock planning, crew scheduling and crew rostering Ghoseiri et al. [2004]. In a nutshell, Train timetable or train scheduling is the layer of Rail transportation planning where it defines the departure and arrival times for all steps of train pathway at their scheduled stop locations following the constraints defined for each railway.

Railway operators commonly make use of running maps to plan the trains position and movements through time. These running maps support dispatchers on the definition of the routes the train must receive in order to arrive at its destination while also performing every planned activity for the train. The route dispatching must take into account several factors, such as stop locations, switches, junctions, priorities, necessity to do shunting operations, crew substitution, terminal status, crossing strategies and other dispatching rules.

These running maps are graphic tools that help the railway operator to plan the railway visually. It is a time-space diagram like the one shown in Figure 1. Each horizontal line is a crossing location, while the space between them is a running line that could be single or double. The names of the stop locations are presented on the left and right side.

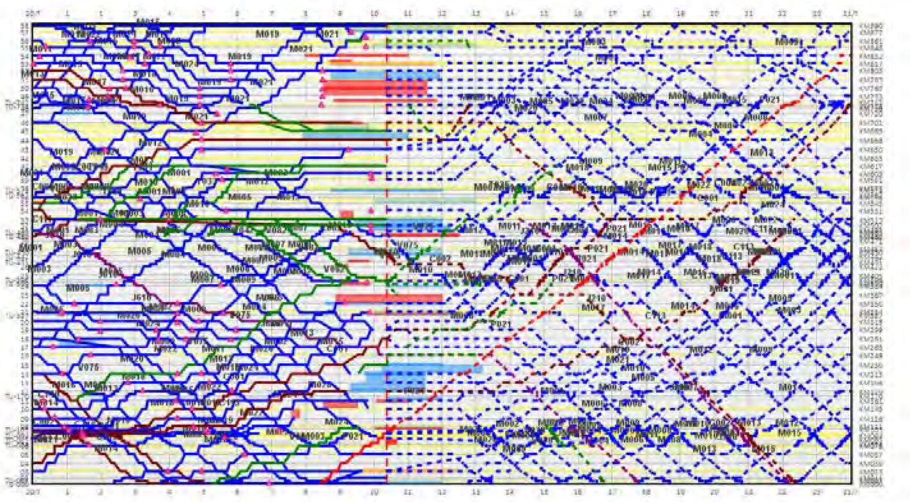

Figura 1: Running Map

Railways with single lines with heavy traffic routinely require conflict resolution. The single line needs to be shared among all the trains that must pass through it, and thus the decision of which train will have priority and which train will wait in order to achieve the optimal solution leads to a combinatorial search problem. This problem is a better description of train timetabling problem (TTP), or train scheduling problem.

In passenger railways, train scheduling is relatively static and cyclic. Usually, these trains have a target time in order to arrive at and depart from each station. However, freight railways, and more specifically heavy haul railways, work on a cyclic pattern, where each train is limited by the railroad segments availability, traffic saturation and terminal capacities for loading and unloading. Therefore, the trains do not have many fixed stations while also departing from the terminals as soon as possible. The real-world railway focused on this article use a mixed traffic composed of cyclic freight trains and passenger trains with a fixed timetable. Since the passenger train has a fixed timetable and is lighter, it has the highest priority over other trains. The cargo trains have different profiles of origin, destination, weight, length, cargo, etc.

Most of the papers found in literature has reported to solve small and medium-sized instances of the problem. Typically, they use real world railway in their tests, making it difficult to compare with other approaches. Thus, the purpose of this paper is provide a generated list of xml files at Pinheiro [2016] where each file representing a railway and proposes an approach to solve it.

The genetic algorithm proposed by Holland [1992] is suitable to solve this kind of problem. It can handle a wide search space and has been successfully applied to other combinatorial problems as it has been already shown. Based on the previous success of this metaheuristics it was chosen in order to tackle the problem.

This paper is organized as follows. In Section 2 prior works on Train Timetabling Problem are considered. Section 3 details the Train Timetabling Problem model used. Section 4 details the proposed Genetic Algorithm method, considering method particularities, solution encoding, genetic operators and individual reconstruction. Section 5 presents the results of the computer experiments, and Section 6 provides a conclusion of the work.

2. Related Works

There has been substantial prior work on TTP, considering issues related to modeling as a proposal Mixed Integer Programming Peña et al. and objective function that minimizes both the fuel consumption and travel delays. For solve it, a Branch-and-bound based approach assisted by a rounding heuristic are proposed Kraay e Harker [1995].

In Ping et al. [2001], a genetic algorithm are employed to adjust the departure order of the trains on a double track corridor. Lu et al. [2004] introduces train acceleration and deceleration rates into the scheduling model.

Epstein et al. [2005] proposes a branch and bound procedure to solve the dispatching problem for a complex rail network. Adjacent propagation and feasibility propagation is used to reduce the search space of the branch and bound procedure. The branch and bound procedure is guaranteed to find the optimal solution. However, the procedure assumes the path for each train (segment of tracks that a train follows) is given.

In Tormos et al. [2008], is proposed a genetic algorithm where the initial individuals are built based on Parameterized Regret-Based Biased Random Sampling (RBRS).

Niu e Zhang [2012] addressed train timetabling problem under the multi-period and uniform demand condition. The train capacity is formulated as a soft constraint and the objective function includes minimizing the waiting times of passengers and the penalty costs. A hybrid genetic algorithm is proposed for solving the non-linear integer programming model.

Sun et al. [2013] adopted Lagrangian duality theory to optimize train schedule for metro line taking into account the robustness and energy saving aspects. The proposed train scheduling model aims to minimize the waiting time of passengers and operational cost subject to the passenger demand, headway, and dwell time constraints.

Mao e Li [2013] presented three schedule adjustment strategies for the train dispatching problem. A novel algorithm is proposed to calculate the maximum traffic capacity in railway lines and a rescheduling method is employed to adjust the freight-train timetables.

Abril et al. [2008] presented distributed and asynchronous search algorithms with application to the train scheduling problem based on an meta-tree Constraint Satisfaction Problem (CSP) structure. Jamili et al. [2012] proposed a hybridized meta-heuristic algorithm based on integration of simulated annealing (SA) and particle swarm optimization (PSO) methods for a periodic single-track train timetabling problem, presented an adaptive large neighbourhood search (ALNS) method to minimize the average passenger waiting time at the stations.

In Barrena et al. [2014], an adaptive large neighbourhood search method to minimize the average passenger waiting time at the stations is presented. Cacchiani e Toth [2012] describes a survey dealing with nominal and robust versions of train timetabling problem. In Arenas et al. [2014], it is presented a constraint-based model for periodic and proposes a genetic algorithm approach to solve it. Cacchiani et al. [2012] proposes modification of the Lagrangian optimization scheme capable of dealing with robustness on train timetabling problem.

3. Train timetabling problem

The train timetabling problem consists in defining the route from the origin to the destination of all trains, considering scheduled activities at specific stations and priority of crossings on single lines. Each train must arrive and depart at all intermediate stations until the arrival at the final station. The problem formulation is similar to Caprara et al. [2002] and Tormos et al. [2008].

The TTP has been classified into two different sub-problems, according to the periodicity of the trains. It could be either a Periodic Train Timetabling or a Non-Periodic Train Timetabling:

Periodic Timetabling - Each trip of a train happens in a specific period and the stop locations have well defined arrival and stop times. Most passenger trains operate on a periodic time table, and some freight trains also operate this way.

Non-Periodic Timetabling - Usually employed in freight railways or heavy haul, each trains does not have a fixed time to depart from the origin or to arrive at different stations. As soon as the train is ready it is able to start the journey, in order to reduce the interval between two trips.

3.1. Constraints

In TTP a set of constraints shall be defined according to traffic rules, railway infrastructure topology and user requirements to obtain a feasible solution. Before we define the restrictions the following parameters are considered:

- T: Finite set of trains t considered in the problem. $T = \{t_1, t_2, ..., t_k\}$
- $T_C \subset T$: Subset of trains that are in circulation.
- $T_{new} \subseteq T$: Subset of non-scheduled trains that do not have yet a timetable and that must be added to the railway line after your planned departure time with a feasible timetable.
- $T = T_C \cup T_{new}$ and $T_C \cap T_{new} = \emptyset$
- l_i: The stop location. This is a place for trains to stop or pass through. Each stop location has capacity defined by a set of tracks (single or double line) and the length of each track must support the larger train.
- L = $\{l_0, l_1, \ldots, l_m\}$: railway set of stop locations that is composed by an ordered sequence that may be visited by trains $t \in T$. The contiguous stop locations l_i and l_{i+1} are linked by a single or double track section.
- J = $\{l_0^t, l_1^t, \ldots, l_{n_t}^t\}$: journey of train t. It is described by an ordered sequence of stop locations to be visited by a train t such that $\forall t \in T, \exists J_i : J_t \subseteq L$. The journey J_t shows the order that is used by train t to visit a given set of locations.
- T_D: set of trains traveling in the down direction. $t \in T_D \leftrightarrow (\forall l_i^t : 0 \leq i < n_t, \exists l_j \in \{L \backslash \{l_m\}\} : l_i^t = l_j \wedge l_{i+1}^t = l_{j+1})$.
- T_U: set of trains traveling in the up direction. $t \in T_U \leftrightarrow (\forall l_i^t : 0 \leq i < n_t, \exists l_j \in \{L \backslash \{l_0\}\} : l_i^t = l_j \wedge l_{i+1}^t = l_{j-1})$.
- $T = T_D \cup T_C$ and $T_D \cap T_C = \emptyset$
- C_i^t: Is an additional time for an schedule activity. Such activities often include crew replacement, fuel supply or any other commercial operation (such as boarding or leaving passengers, or wagon shunting operation) at station i.
- $\Delta^t_{i \to (i+1)}$: Journey time for train t from location l_i^t to l_{i+1}^t.

- H_l^1 initial time of interdiction of stop location l for maintenance purpose.
- H_l^2 end time of interdiction of stop location l for maintenance purpose.
- dep_i^t departure time of train $t \in T$ from the location i, where $i \in J_t \setminus \{l_{n_t}^t\}$
- arr_i^t arrival time of train $t \in T$ to the location i, where $i \in J_t \setminus \{l_0^t\}$.

Journey Time: for each train and each stop location. A journey time is given by $\Delta_{i\to(i+1)}^t$, which represents the time the train t spent to depart of one stop location and arrive on the next:

$$arr_{i+1}^t = dep_i^t + \Delta_{i\to(i+1)}^t \quad (1)$$

Crossing: any track or railroad segment resource between 2 stop locations can not be occupied by more than 1 train:

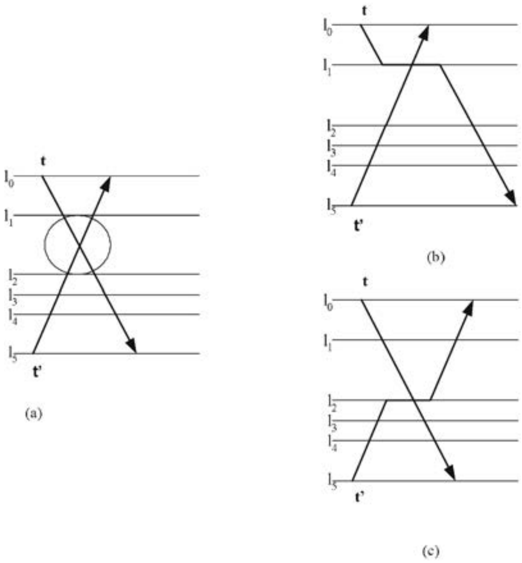

Figura 2: (a) Crossing conflict. (b) Train in Down direction waits. (c) Train in Up direction waits.

Comercial Stop: each train $t \in T$ is required to remain in stop location l_i^t at least C_i^t time units:

$$dep_i^t = arr_i^t + C_i^t \quad (2)$$

Closing Time: Let $[H_l^1, H_l^2]$ be the closing time for interdiction maintenance operations of stop location l, thus trains can neither pass nor stop on this stop location:

$$dep_l^t < H_l^1 \vee arr_l^t > H_l^2 \quad (3)$$

Headway Time: is time interval that resources (track or segment) between stop locations are occupied by a train.

Departure Time: is determined by max time between:

- Last HeadWay Time from other train arrived in stop location i if the other train is in a different direction;
- Last HeadWay Time from other train arrived in next stop location (i + 1) if the other train has the same direction;

- First departure time on next stop location (i + 1) after arrival time of train t_i on current stop location.

Stop Location: it's not allowed for a train to stop out of one of the predefined stop locations in order (for example, on a track of single line) to avoid deadlock constraints.

Overtaking: it is allowed overtaking another train on any stop location of multiple-tracks since deadlocks must be avoided. Pachl [2011] proposed rules to handle this. These rules were implemented, however due to the fact that they can lead to a substantial increase in evaluation time, some of the rules had a limit on how many track reservations on the stack.

For this approach the arr_i^t and dep_l^t depends on the performance of each class of train in the last few hours. The journey from one stop location to another $\Delta^t_{i\to(i+1)}$ thus is not constant and speed restrictions or traffic saturation can change this values.

3.2. Objective function

The measurement of the quality of each solution is based on Tormos et al. [2008], but since the freight trains have no restriction on timetabling on the different stations, the whole set T will be used instead of only T_{new}. In order to estimate the loss of production for a single train dispatch, the minimum time of travel must be calculated, assuming that the train being considered was the only one on the network. This reference value for train t is defined like:

$$T^t_{opt} = \sum_{i=0}^{n_t-1} \Delta^t_{i\to(i+1)} + \sum_{i=0}^{n_t-1} C^t_i \tag{4}$$

Once the best reference value has been obtained the solution will be the average delay of all trains with respect to their best reference value:

$$\eta_t = (arr^t_{n_t} - dep^t_0) - T^t_{opt} \tag{5}$$

$$\eta = \frac{\sum_{t\in T}\eta_t}{\mid T \mid} \tag{6}$$

The objective function for this problem is to minimize η

4. The Genetic Algorithm

The TTP is an optimization combinatorial problem whose search space grows exponentially when the number of trains or track sections increases. In this approach we consider that every train must be optimized since the trains already set on track do not have a previously fixed timetable, and thus have joined the sets T_C and T_{new}. This also evokes that over takings are now possible.

Each train pathway can be defined as a list of sorted train-stop location pair from its origin or current location to its destination since they are built verifying constraints of time, resource allocation for other train and stop location precedence.

4.1. Individual Encoding

The individual is composed of a set of genes. Each gene is a tuple in the form (train, stop location) represented by (t_i, j^l_k) where l is a binary indicator of whether the train is arriving or departing from section j.

An example can be seen on Table 1, where we have a representation of a train t_0 departing from l_0 to l_2 passing through l_1.

(t_0, l^1_0)	(t_0, l^0_1)	(t_0, l^1_1)	(t_0, l^0_2)

Tabela 1: $(t_0, l^1_0) \to (t_0, l^0_2)$

The movement of a train from a stop location to another can not be split between the departure and arrival, because you can't park a train outside the defined stop locations. As a simple example, you can't stop a train on a single line. This constraint must be taken into account in individual generation and on the operations of crossover and mutation. The individual is built as a list of all trains considered in the problem since the current position for $t \in T$ or origin for $t \in T_{new}$ up to their destinations. An individual is then be represented like on table 2. The individuals then can be seen simple as a permutation of all the tuples (t_i, j_k^l) that belong to the problem.

(t_i, l_0^1)	(t_i, l_1^0)	(t_j, l_4^1)	(t_j, l_5^0)	(t_k, l_4^1)	(t_k, l_3^0)	$\cdots$

Tabela 2: Example individual

4.2. Initial Population

The initial population is built by creating a set of individuals without constraints. To create a individual a roulette wheel is used where a heuristic based algorithm define the proportional value of the roulette choose a train. The individual is build by add a gene to them verifying the Pachl [2011] rules or other constraint. If any rule or constraint is broken the train postponing his turn in roulette wheel. Five initial strategies are used:

1. The most delayed train has more proportional value at roulette.
2. all trains has the same proportional value at roulette, the choice is completely random.
3. Unloaded train or mine direction has more proportional value at roulette.
4. Loaded train or port direction has more proportional value at roulette.
5. For each train randomly is chosen one of above strategies.

These techniques allow to use a good initial solution. The genes selected by the roulette are appended to the individual, and that gene is removed from the roulette for the subsequent runs.

A round is considered complete when all trains have been selected once. After this, the reference time is updated to the last movement, and all trains that have not arrived at their destination are inserted again on the roulette wheel. Any trains that have departure from their origins that can now be included on the wheel are also included during this step.

If after a round all considered trains in the set T have arrived at their destinations, the result is an individual. This individual is then inserted into the population and the process starts again.

4.3. Selection

For the selection procedure of the genetic algorithm the roulette wheel based on fitness was used, with some modifications.

In order to allow for both exploitation of good regions on the solution space while not negating new individuals in exploration, 20% of the elite population is directly transferred to the new population. The remaining population is generated by crossover and mutation. However, a niche by fitness distance of 0.01 is used to remove the closest individuals from last to first of new population until the original size of population is reached. After that, with up to the roulette wheel used did not consider all individuals. A randomly moving window was used in order to ensure a higher level of variety on the final population for the next generation.

4.4. Crossover

The crossover used is based on the Partially Matched Crossover (PMX). This crossover technique uses two random points to define the crossover. The son has a first and last part from the father. The intermediate part is inherited from the mother, keeping the priority sequence present on the mother of the missing genes. The generation of the daughter follows a similar pattern.

4.5. Mutation

During the mutation process, a random gene from the individual to be mutated is select. Then, the previous and next movement of the train on the selected gene is found. Finally, the selected gene is placed in a valid position respecting the constructive constraint of individuals.

4.6. Recovering invalid individuals

While operating on the generation, crossover or mutation of individuals, it is a common problem to run into deadlocks.

During the individual generation, any time a deadlock is found, the movement that is responsible for causing the deadlock is blocked and another movement is selected in order to become the next gene. The blocked gene is postponed until it no longer generates a deadlock.

While the individual is being mutated, after selecting the uniformly random position of insertion, the sequence is assessed to avoid deadlocks. While there is a deadlock in the system, a new position is chosen again for the selected gene.

4.7. Decoding Process

In order to decode the sequence of arrivals and departures into the objective function, the individual is then converted into a running map. Since a step is taken in order to ensure that the individual is valid, all that is needed is to have an internal clock that is updated while the first genes are being processed.

5. Computational results

Initially, in order to assess the performance of the proposed heuristic, a set of 5 small railways were generated. Since the main results are on real world sized railways, these railways will be numbered #6 - #10. The larger of these small railroads could barely be solved by exact methods on a timely manner, as can be seen on Table 3, where the pairs column refers to the number of (t_i, j_k^l) pairs for the permutation problem.

In order to compared performance, a stripped down version of the proposed genetic algorithm was run with 10 individuals and 25 generations. However, it must be noted that the deadlock avoidance, together with the initial population generation was able to achieve global optimum on the first generation on 100% of the railways tested, even with only 10 individuals generated and no processing of any of the genetic operators.

Railway		Branch&Bound		GA	
Num.	Gens	Fitness	Time	1^{st} gen	25^{th} gen
#6	16	1.0098	0.16	1.0098	1.0098
#7	24	0.9049	2.05	0.9049	0.9049
#8	26	1.1615	4.06	1.1615	1.1615
#9	36	0.9049	52.55	0.9049	0.9049
#10	40	1.1376	137.71	1.1376	1.1376

Tabela 3: Branch&Bound and GA comparison

For the railway experiment, 5 larger railways were generated. These railways are real-world sized freight railways. Each of them was tested for each combination of initial population generation strategy and niche strategy, for a total of 10 tests. The tests consisted of 30 runs of the GA in order to increase statistical significance of the results obtained.

In order to compare overall performance of niche and initial population strategy, a last test was performed where all runs were normalized with $f_n = \frac{f_o}{min(f \in T)}$, where f_n is the normalized fitness, f_o is the original fitness and T is the set of all tests run for the same railway. Therefore a normalized fitness of 1.03, for example, means that on average the result was 3% higher than the best solution found.

The results for the niche strategy can be seen on Table 4. For comparison, both ANOVA and t-tests were performed. The results marked in bold are statistically relevant ($p \leq 0.05$). In fact, all of the results had negligible values of p, always lower than 10^{-25}. It can be seen that the addition of the niche strategy improved the results significantly.

Railway	Without Niche	With Niche
#1	55.05	**54.52**
#2	26.00	**25.51**
#3	40.32	**39.72**
#4	18.32	**17.81**
#5	29.06	**28.74**
Global	1.0378	**1.0206**

Tabela 4: Average Fitness - Niche Strategy

The results for the initial population generation strategies can be seen on Table 5. A one-way ANOVA test was performed on all railways and the railways with $p \leq 0.05$ are in bold. It can be seen that the initial population has significant effect on all railways but railway #4. This leads to the significance of choosing an individual population considering all railways on the normalized global case.

	Strategy				
Railway	#1	#2	#3	#4	Combo
#1	54.94	54.78	54.77	54.73	54.70
#2	25.53	25.84	25.87	25.88	25.66
#3	39.88	40.09	40.13	40.01	40.00
#4	18.15	18.02	18.09	18.02	18.06
#5	28.82	28.96	28.93	28.94	28.87
Global	1.0276	1.0301	1.0311	1.0297	1.0277

Tabela 5: Average Fitness - Initial Pop. Strategy

It has been observed that on all railways where the choice of strategy had a relevant impact on the final result, the combo strategy was either the best or the second best. This leads naturally to a question of whether the combination of all strategies is superior to the other best option.

In order to perform such test, the two best strategies for each railway except #4 and for the global result were compared on a t-test for equivalence of means. The results can be seen on Table 6, where the significant results ($p \leq 0.05$) are in bold.

Railway	Best Strategy	Avg Fitn.	Combo Fitn.
#1	Strat. #4	54.73	54.70
#2	Strat. #1	**25.53**	25.66
#3	Strat. #1	**39.88**	40.00
#5	Strat. #1	28.82	28.87
Global	Strat. #1	1.0276	1.0277

Tabela 6: Average Fitness - Initial Strategy Comparison

It is possible then to conclude that although the selection of strategy for the initial population has importance as shown on Table 5, Strategy #1 was better than the combo strategy for two railways but not for the combination of all railways, thus suggesting that for new instances choosing either strategy #1 or the combo should be recommended.

6. Conclusion

The train timetable of this work considered new and current trains on a railway such as takeover on trains moving in same direction increasing the search space and put the need to repair unfeasible schedules caused by deadlocks. The schedule for theses trains is obtained with a Genetic Algorithm that includes a guided process to build the initial population based on 5 heuristics strategies defined in section 4.2. These heuristics improve lightly the final results comparing the full random strategies. The results of the computational experience, point out that GA is an appropriate method to explore the search space of this complex problems and that further research in the design of efficient GA is justified. The GA approach proposed in this work might be improved with the use of local search able to intensify performance around promising regions of local optima.

Although the approach of niche inside Genetic Algorithm improve the search process, For further work, other heuristics can be verified to put together and add a better performance to the whole process, such as Evolucionary Cluster Search (Oliveira e Lorena [2004]), Tabu Search and Simulated Annealing.

Referências

Abril, M., Salido, M. A., e Barber, F. (2008). Distributed search in railway scheduling problems. *Engineering Applications of Artificial Intelligence*, 21(5):744–755.

Arenas, D., Chevirer, R., Hanafi, S., e Rodriguez, J. (2014). Solving the periodic timetabling problem using a genetic algorithm. *arXiv preprint arXiv:1411.6998*.

Barrena, E., Canca, D., Coelho, L. C., e Laporte, G. (2014). Single-line rail rapid transit timetabling under dynamic passenger demand. *Transportation Research Part B: Methodological*, 70:134–150.

Cacchiani, V., Caprara, A., e Fischetti, M. (2012). A lagrangian heuristic for robustness, with an application to train timetabling. *Transportation Science*, 46(1):124–133.

Cacchiani, V. e Toth, P. (2012). Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727–737.

Caprara, A., Fischetti, M., e Toth, P. (2002). Modeling and solving the train timetabling problem. *Operations research*, 50(5):851–861.

Epstein, D., Lu, Q., Zhao, J., e Leachman, R. (2005). An exact solution procedure for determining the optimal dispatching times for complex rail networks. Technical report, Technical report. Los Angeles, CA, USA: Department of Industrial and Systems Engineering, University of Southern California.

Ghoseiri, K., Szidarovszky, F., e Asgharpour, M. J. (2004). A multi-objective train scheduling model and solution. *Transportation research part B: Methodological*, 38(10):927–952.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Jamili, A., Shafia, M. A., Sadjadi, S. J., e Tavakkoli-Moghaddam, R. (2012). Solving a periodic single-track train timetabling problem by an efficient hybrid algorithm. *Engineering Applications of Artificial Intelligence*, 25(4):793–800.

Kraay, D. R. e Harker, P. T. (1995). Real-time scheduling of freight railroads. *Transportation Research Part B: Methodological*, 29(3):213–229.

Lu, Q., Dessouky, M., e Leachman, R. C. (2004). Modeling train movements through complex rail networks. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 14(1):48–75.

Mao, H. e Li, Z. (2013). Train schedule adjustment strategies for train dispatch. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 11(5):2526–2534.

Niu, H. e Zhang, M. (2012). An optimization to schedule train operations with phase-regular framework for intercity rail lines. *Discrete Dynamics in Nature and Society*, 2012.

Oliveira, A. C. e Lorena, L. A. (2004). Detecting promising areas by evolutionary clustering search. In *Advances in Artificial Intelligence–SBIA 2004*, p. 385–394. Springer.

Pachl, J. (2011). Deadlock avoidance in railroad operations simulations. In *Transportation Research Board 90th Annual Meeting*, number 11-0175.

Peña, M., Ramos, A., Fernández, A., e Cucala, A. P. Train timetabling problem for complex railway systems.

Ping, L., Axin, N., Limin, J., e Fuzhang, W. (2001). Study on intelligent train dispatching. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, p. 949–953. IEEE.

Pinheiro, E. (2016). RailWay File Repository. `https://github.com/ehfp/ttp/`. [Online; accessed 03-Feb-2016].

Sun, X., Zhang, S., Dong, H., e Zhu, H. (2013). Optimal train schedule with headway and passenger flow dynamic models. In *Intelligent Rail Transportation (ICIRT), 2013 IEEE International Conference on*, p. 307–312. IEEE.

Tormos, P., Lova, A., Barber, F., Ingolotti, L., Abril, M., e Salido, M. A. (2008). A genetic algorithm for railway scheduling problems. In *Metaheuristics for Scheduling in Industrial and Manufacturing Applications*, p. 255–276. Springer.