# Real-Time Optimization of Energy Consumption in Railway Networks

Federico Naldini, Paola Pellegrini, Joaquin Rodriguez

Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 00 (2021) 000–000

Transportation Research Procedia

www.elsevier.com/locate/procedia

24th Euro Working Group on Transportation Meeting, EWGT 2021, 8-10 September 2021, Aveiro, Portugal

# Real-Time Optimization of Energy Consumption in Railway Networks

Federico Naldini[a,*], Paola Pellegrini[b], Joaquin Rodriguez[c]

[a]*DEI "Guglielmo Marconi", Alma Mater Studiorum Università di Bologna, Viale Risorgimento 2, 40136 Bologna, Italy*
[b]*COSYS-LEOST, Univ Gustave Eiffel, IFSTTAR, UnivLille, F-59650 Villeneuve d'Ascq, France*
[c]*COSYS-ESTAS, Univ Gustave Eiffel, IFSTTAR, UnivLille, F-59650 Villeneuve d'Ascq, France*

**Abstract**

In railway traffic, perturbations may give rise to conflicts, causing delays. As a countermeasure, effective re-scheduling and re-routing decisions can be taken by addressing the real-time Rail Traffic Management Problem (rtRTMP). One of its subproblems is the real-time Energy Consumption Minimization Problem (rtECMP). The latter enforces the train routing and precedences computed by a rtRTMP solver and defines train timings and speed profiles. The objective is to minimize the weighted sum of train energy consumption and total delay. In this paper, we propose an Ant Colony Optimization algorithm for the rtECMP and we test it on the French Pierrefitte-Gonesse control area with dense mixed traffic. The results show that, in a very short computing time, a remarkable exploration of the search space is performed before convergence.



*Keywords:* Energy consumption; Multiple trains; Ant Colony Optimization; Traffic Management; Rail Transport;

## 1. Introduction

Managing traffic perturbations in railway networks to reduce the overall delay is known as *real-time railway traffic management problem* (rtRTMP) (Pellegrini et al., 2014). Its solution is a feature of modern rail *traffic management systems* (TMS). The task of managing traffic to reduce delay is entrusted to a dispatcher, who is in charge of a control area, i.e., a limited size railway network. The *real-time Energy Consumption Minimization Problem* (rtECMP), as introduced by Montrone et al. (2018), has the objective of minimizing the weighted sum of train energy consumption and total delay by deciding speed profiles in a given control area and time horizon. It takes as input the decisions on train routing and precedences made by a rtRTMP solver. In addition, having to define energy-efficient speed profiles for multiple interacting trains, it takes into accounts infrastructure characteristics, operational constraints and

* Corresponding author.
*E-mail address:* federico.naldini4@unibo.it

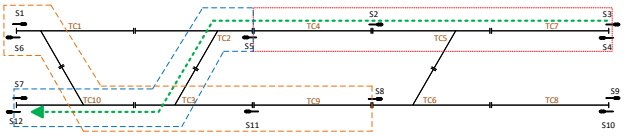



Fig. 1: Example of a simple infrastructure in which *T*C denotes track-circuits, *S* denotes signals. The dashed green arrow highlights a train route composed of block sections S4-S5 (dotted red) and S5-S12 (dashed blue). Block section S1-S8 (dashed orange) is incompatible with S5-S12 as they share TC3 and TC10.

train dynamics. The rtECMP ouputs include arrival, departure, passing through and dwelling times along with speed profiles.

Minimization of energy consumption in train operations is the main concern of the literature on Energy-Efficient Train Control (EETC) and Energy-Efficient Train Timetabling (EETT) of which Scheepmaker et al. (2017) provided a comprehensive review. EETC methods focus on minimizing the traction energy consumption of one or multiple trains given a timetable to fulfill and, therefore, they aim at computing energy-efficient speed profiles. ETTT methods focus on scheduling one or multiple trains. Here, energy saving by EETC is considered along with either timetabling objectives, in case of nominal traffic, or re-scheduling objectives, in case of perturbed traffic. For perturbed traffic, we refer the reader to, e.g., Wang and Goverde (2017), Galapitage et al. (2018), Luan et al. (2018), Montrone et al. (2018), Li et al. (2020). For nominal traffic, we refer the reader to, e.g., Goverde et al. (2016), Wang and Goverde (2019), Xu et al. (2020).

Montrone et al. (2018) proposed a mixed-integer linear programming formulation for the rtECMP. However, the solution of such formulation quickly becomes impractical for realistic instances, due to too long computing times. In this paper, we extend the research of Montrone et al. (2018) by proposing a graph-based rtECMP model that we solve with an Ant Colony Optimization (ACO) algorithm, to which we refer as ACO-rtECMP. An experimental analysis is conducted on a real life railway infrastructure (the Pierrefitte-Gonesse junction, located in France) subject to various traffic perturbations.

The rest of the paper is structured as follows: Section 2 defines the problem by also introducing operational constraints and characteristics of train speed profiles; Section 3 describes our graph-based model for the rtECMP; Section 4 discusses the proposed ACO algorithm; Section 5 explains our experimental setup; Section 6 analyzes the results; Section 7 exposes our conclusions.

## 2. Problem

As discussed in the introduction, solving the rtECMP requires defining an energy-efficient speed profile for each train when traffic perturbations occur. The objective is to jointly minimize the overall delay and energy consumption. It is crucial for the chosen speed profiles to be compliant with traffic management decisions (routing and precedences) and railway safety requirements. The problem constraints depend on two main aspects: infrastructure characteristics and train physical capabilities impacting possible speed profiles, and operational rules.

For what concerns possible speed profiles, we consider four alternative driving regimes which have been proven to be convenient based on the *Pontryagin's maximum principle*: maximum traction, cruising, coasting and maximum braking. For further details, see, e.g., Scheepmaker et al. (2017).

For what concerns the modeling of operational rules, we consider a microscopic representation of rail infrastructures. Specifically, the infrastructure in a control area is composed of *track-circuits*. These are track elements equipped with an electric device for sensing the presence of a train. Track-circuits are grouped into *block sections*, which are track stretches that can be traversed by only one train at a time to maintain safe distancing. Every train is assigned a route in the infrastructure by a TMS. More precisely, we define a route as a sequence of block sections linking an origin-destination pair. Control areas are equipped with an *interlocking system* that ensures a coherent route formation. To indicate whether a block section can be accessed or not, the interlocking system employs signals, which are designed to give visual instructions to drivers. In particular, every block section is delimited by two signals placed at its entrance and exit location (see for instance Pachl 2002). Figure 1 shows an example of a simple infrastructure.

In the simplest case, a signal can display three so-called *aspects*: green, yellow and red. A red aspect forbids the access to the following block section. A signal displaying yellow allows the access and demands a slow down so that a full train stop is possible before the following signal. Green grants the access to a block section and implies that a driver can safely enter the following one at full speed, if suitable. Each signal in a route may have a different number of aspects, up to some $n \geq 3$. This means that, when a $n$-aspect signal is crossed with a green, at least $(n-2)$ block sections are available in front of it. In addition to red, yellow and green, $n$-aspect signals can display further *restrictive aspects*. They progressively demand greater speed reductions to be reached by the end of the block section considered, up to a stop before a red signal. The interlocking system ensures that the length of $(n-2)$ block sections is sufficient for a train to brake after a signal with $n$ aspects, provided that the train type is authorized to use the infrastructure.
Given a block section $b$, we designate all the block sections sharing one or more track-circuits with $b$ as *incompatible* (in Figure 1, block sections S5-S12 and S1-S8 are incompatible as they share track-circuits TC3 and TC10). When $b$ is available, having its opening signal displaying an aspect different from red, a train can enter it.
In addition to safety, the interlocking system also allows the enforcement of traffic management decisions. As for routing decisions, it is in charge of setting block sections for trains so that the planned route is indeed traversed. As for precedence decisions, it allows unlocking block sections along a route only if all the included track-circuits are to be used, first of all, along the routes itself.

## 3. Model

In this section, we propose a graph-based model for the rtECMP as defined in Section 2. In the model definition, a train speed profile is defined as a combination of *partial speed profiles*. For each train and for each block section in the train's route, we pre-compute a discrete set of partial speed profiles.
We are given a set of trains $i \in T$ and set of block sections composing a route for each of the trains $B_i$. We refer to a generic block section in the route of train $i$ as $j \in B_i$. Let $S$ be the set containing train-block pairs $(i, j) : i \in T, j \in B_i$, such that either $j$ is the last block section in the route of $i$ (before the train leaves the control area), or $j$ is where a scheduled stop for $i$ is planned. For each $(i, j) \in S$, we denote the scheduled departure time as $\lambda_{i,j}$. For each $i \in T$ and $j \in B_i$, we are provided with train precedence in the form of a set $\mathcal{P}_{i,j}$ containing train-block pairs $(i^*, j^*)$. Only when every train $i^*$ has released its block $j^*$, $i$ can access $j$. Let $G = (V, A)$ be a graph with $V$ and $A$ set of vertices and arcs respectively. For each block section $j \in B_i$, we pre-compute a discrete set of partial speed profiles with the following attributes: an initial speed; a final speed; a value of energy consumed; a running time. Except for two special vertexes, *source* ($s$) and *terminal* ($t$) (which indicate the beginning and end of every route), each vertex in graph $G$ represents a partial speed profile for a train-block pair. The attributes of a generic $v \in V$ are denoted by: $i_v$ (train to which the node refers); $j_v$ (the considered block section of $B_{i_v}$); $k_v$ (initial speed); $k'_v$ (final speed); $r_v$ (running time of $i_v$ in $j_v$); $E_v$ (energy consumption).
The set of arcs $A$ of $G$ is defined as follows:

- For all $a, b \in V$ such that $i_a = i_b$, arc $(a, b)$ exists if all following statements are true: $j_a$ directly precedes $j_b$ in the train's route; the final speed associated to node $a$ is equal to the initial speed associated to node $b$, i.e., $k'_a = k_b$.
- For all $a, b \in V$ such that $i_a \neq i_b$, arc $(a, b)$ exists if one of the following conditions is verified: $(i_a, j_a)$ is in $\mathcal{P}_{i_b,j_b}$; $j_a$ and $j_b$ are compatible.
- For all $v \in V$, arc $(s, v)$ exists if train-block pair $(i_v, j_v)$ verifies the following conditions: $j_v$ is the first block section in the route of $i_v$; no other train precedes $i_v$ in $j_v$, i.e., $\mathcal{P}_{i_v,j_v} = \emptyset$.
- For all $v \in V$, arc $(v, t)$ exists if $j_v$ is the last block section of the route of train $i_v$.

An example $G(V, A)$ is shown in Figure 2, where two trains, denoted as T1 and T2, are considered. The route of T1 comprises block sections A and B. The route of T2 comprises block sections C and D. Let us suppose that only block sections D and B are incompatible, and that T2 must have cleared D before T1 can enter B, i.e., $(T2, D) \in \mathcal{P}_{T1,B}$. Nodes are partitioned in four clusters (colored rectangles), each associated to a given train-block pair: (T1, A); (T1, B); (T2, C); (T2, D). The black arcs connect compatible block section speed profiles for a train in terms of initial/final speed. In addition, they connect the source to the first block sections' nodes. Similarly, black arcs connects the last

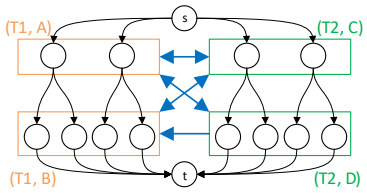



Fig. 2: Example $G(V, A)$ described in Section 3.

block sections' nodes to the terminal. As A and C are compatible, each vertex in (T1,A) is bidirectionally connected to each vertex in (T2,C). The corresponding arcs are replaced by a single blue bidirectional arc connecting the corresponding rectangles, for readability. The same applies for the bidirectional arcs between: (T1, A)-(T2, D), (T1, B)-(T2, C), and (T1, A)-(T2, D). Due to the precedence constraint, nodes in (T2, D) are connected to nodes in (T1, B), but not viceversa. The blue arrow linking the two rectangles is hence monodirectional.
A path $h$ in $G$ that satisfies the following set of constraints is a feasible solution of the rtECMP:

- **C1** For each $(i, j) : i \in T, j \in B_i$, exactly one vertex such that $(i_v, j_v) = (i, j)$ must be in $h$. This makes sure that a unique choice of partial speed profile is effected per train-block pair.
- **C2** For each $v, w \in h$ referring to two consecutive block sections $j_v$ and $j_w$ that have to be traversed by the same train ($i_v = i_w$), it must be true that $k'_v = k_w$. This ensures speed profile continuity on the boundaries between every pair of consecutive block sections traversed by a train.
- **C3** For each $v \in h$, $k'_v$ must be coherent with the signal aspect opening block section $j_v$ when $i_v$ enters it. This constraint models the interlocking system functioning as discussed in this section.

The optimal solution is the one minimizing the weighted sum of normalized energy consumption and delay.

## 4. Algorithm

We propose an Ant Colony Optimization (ACO) algorithm (Dorigo and Stützle, 2004) to tackle the graph model introduced in Section 3. Precisely, we adopt a $\mathcal{MAX}$–$\mathcal{MIN}$ *ant system* ($\mathcal{MM}$AS) algorithm (Stützle and Hoos, 2000), to which we refer as *ACO-rtECMP*.
ACO is a population-based meta-heuristic modeled after the pheromone-based cooperative behavior of ants. In ACO, ants are computational agents implementing a stochastic constructive heuristic for the problem at study. The elements of a solution are the nodes of a so-called *construction graph*. To build a solution, each ant moves across the graph, one node at a time, selecting, at each decision step, a node in the neighborhood of the current one. The selection of each node is done through the so-called pseudo-random proportional rule. It is biased by two factors: *heuristic information* and *pheromone trails*. The former is a problem-specific quantity associated to nodes or arcs that represents a greedy measure of their attractiveness. The latter is a quantity representing a collective memory on the quality of previously visited solutions containing the node or arc on which pheromone is deposited. Pheromone constitutes an indirect mean of communication between ants as it is deposited by them throughout a run of the algorithm and it guides further ants' individual search. The algorithm is iterative: at each iteration, a group of ants, named colony, builds solutions. Based on these solutions, the pheromone trail is updated to lead future ants towards promising regions of the search space.
The rtECMP, by nature, is well suited to be addressed with a constructive, incremental approach such as ACO. Precisely, from a chronological standpoint, the initial decisions made have a strong impact on the subsequent ones both in terms of feasibility and in terms of resulting quality. In an ACO algorithm, decisions can easily be made following both the chronological order and the one imposed by the precedence constrains. This allows ants to choose the speed profile of each block section being aware of the signal aspect opening it. Therefore, only the candidate profiles matching the possibly necessary braking are considered and infeasible solutions are avoided entirely. Furthermore, the rtECMP involves many complex space-time relations arising between groups of interacting trains. Here, another advantage of ACO is that it does not require these relations to be formalized into *a priori*. Indeed, they are autonomously learned through the evolution of the pheromone trails.

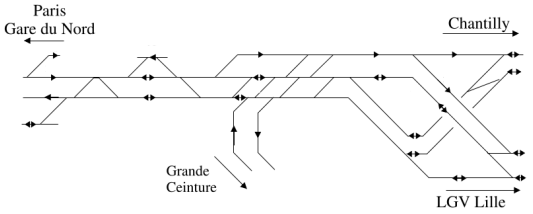



Fig. 3: Network layout of the infrastructure in the Pierrefitte-Gonesse control area.

In the ACO-rtECMP, we use $G$ as the construction graph as we tackle the model in Section 3. Hence, an ant is required to construct a path $h$ in $G$ subject to constraints **C1**-**C3**.
Each vertex $v \in V$ is assigned a heuristic information value (denoted as $\eta_v$) defined as:

$$\eta_v = w_0 \left( \frac{\epsilon_{max}^{(j_v)} - E_v}{\epsilon_{max}^{(j_v)} - \epsilon_{min}^{(j_v)}} \right) + w_1 \left( \frac{\gamma_{max}^{(j_v)} - r_v}{\gamma_{max}^{(j_v)} - \gamma_{min}^{(j_v)}} \right) \qquad \forall v \in V, \tag{1}$$

where

- $\epsilon_{min}^{(j_v)}$ and $\epsilon_{max}^{(j_v)}$ are respectively the minimum and maximum energy-consumption values w.r.t. all the available vertices associated to block section $j_v$ and train $i_v$,
- $\gamma_{min}^{(j_v)}$ and $\gamma_{max}^{(j_v)}$ are respectively the minimum and maximum running-time values w.r.t. all the available vertices associated to block section $j_v$ and train $i_v$.
- parameters $w_0$ and $w_1$ are the weights of relative importance of, respectively, energy and delay. The same weights are used in the objective function.

The so defined $\eta$ varies between 0 and 1. It favors vertices representing faster and less energy-consuming partial speed profiles.
A pheromone update is applied after each iteration. First, a *pheromone evaporation* phase takes place, causing a fraction $\rho < 1$ of the current pheromone trails $\tau$ to be removed. Second, a *pheromone deposit* is performed on the arcs of the *best-so-far* (*bsf*) or the *iteration-best* (*ib*) solution. The use of one or the other is driven by parameter $f_{ib}$. Moreover, every $\tau$ is bounded between $\tau_{min}$ and $\tau_{max}$ (Stützle and Hoos, 2000).

## 5. Experimental setup

We test the ACO-rtECMP on the French Pierrefitte-Gonesse control area (Figure 3). It consists of an infrastructure with 89 track-circuits grouped into 79 block sections. The possible train routes are 39 and the control area does not contain stations. The traffic is dense: 336 trains are scheduled to traverse the control area in a classic week-day timetable. Three kinds of trains are used here, with different rolling-stock features. To construct a set of rtECMP instances, we first generate random perturbations of the timetable and construct the corresponding rtRTMP instances. To do so, following the literature (Pellegrini et al., 2015), we select a random sample containing 20% of the trains planned in a real week-day timetable. We then create 100 daily perturbations by delaying the nominal entry time in the control area, for each of the sampled trains, of a random amount between 5 and 15 minutes. For each perturbation, we consider the same 1-hour time horizon (06:00-07:00AM), which represents the morning peak time. This leads to a set of 100 rtRTMP instances comprising either 15 or 16 trains. By employing the rtRTMP solver RECIFE-MILP (Pellegrini et al., 2015) to address the 100 perturbations, we finally obtain the routing and precedences required to construct the corresponding 100 rtECMP instances. They correspond to the best (possibly proven optimal) set of routes and schedules to minimize total delays found in the available computational time.

| $w_0$ | $w_1$ | $\bar{E}$ $[10^3 J]$ | $\bar{D}$ $[s]$ | $\bar{I}$ | *iter* | *time* $[s]$ | $\bar{B}$ |
|---|---|---|---|---|---|---|---|
| 0.9999 | 0.0001 | 855 | 13,409 | 64% | 92 (252) | 12s | 86% |
| 0.5 | 0.5 | 2,119 | 8,248 | 28% | 108 (190) | 19s | 48% |
| 0.0001 | 0.9999 | 6,946 | 3,291 | 20 % | 104 (170) | 21s | 26% |

Table 1: ACO-rtECMP results on 50 instances (with 3 different objective function weight $(w_0, w_1)$ configurations) and benchmarks against the corresponding random-search results. Time limit: 30 seconds.

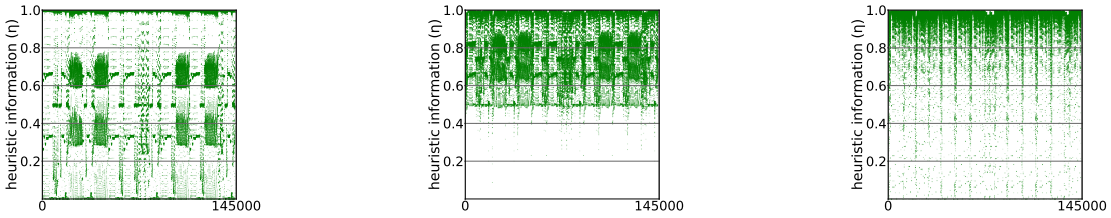
To ensure real-time applicability, the solution of both the rtRTMP and the rtECMP must be completed within a short computing time. Considering that, three minutes are often allocated for real-time traffic management (Samà et al., 2016), we assign 30 seconds to ACO-rtECMP imagining that the remaining 150 seconds are devoted to the rtRTMP. Such computing times, although to some extent arbitrary, are in line with the currently envisaged deployment of a TMS (Shift2Rail, 2019). In particular, it is in line with human-in-the-loop specifications: as dispatchers are meant to assess and possibly modify optimization decisions before their implementation, some minutes are to be appointed to decision making. Three of these minutes can reasonably be used for optimization.
To choose the appropriate parameter setting of ACO-rtECMP, we use IRACE (López-Ibáñez et al., 2016). It implements a set of machine learning tools for automatically configuring algorithms. Half of the rtECMP instances is used to tune the algorithm's parameters. The values tested for these parameters are: exponential weight of $\eta$ and $\tau$ in the pseudo-random proportional rule in $\{1, \ldots, 10\}$, the number of ants in $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ and $\rho \in \{0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.4, 0.5, 0.6, 0.8\}$. We perform one *ib*-based pheromone update every 50 iterations, i.e., $f_{ib} = 1/50$. We allow a total of 181,500 algorithm executions. The chosen algorithm configuration confirms our intuition on the aimed algorithm behavior. Indeed, the heuristic measure is a very good lead for ants' decisions. Its weight in the pseudo-random proportional rule is hence rather high compared to the pheromone's one (10 vs 2). Moreover, pheromone evaporation rate is low (0.05) so as to give time to colonies to build on the common knowledge cumulated throughout several iterations. A high number of ants (100) compose each colony to preserve a remarkable level of exploration.
Experiments are run on the 50 instances that were not used used for tuning. We test three configurations of weights $(w_0, w_1)$ in the objective function: $(0.9999, 0.0001)$ where mostly the energy-consumption term matters; $(0.5, 0.5)$ where both energy and delay are equally weighted; $(0.0001, 0.9999)$ where mostly the delay term matters. We perform the tuning considering weights $(0.5, 0.5)$ and we consider the selected configuration for all cases. Preliminary analysis showed that the configuration performance is not very sensitive to these weights. We benchmark the results we obtain against those of a random search performed on the same instances and with the same computing time limit.
We implement ACO-rtECMP in C++ and run the experiments on a AMD Ryzen™ 7 2700X 8-core, 3.7GHz CPU with 64 GB of ram and Ubuntu 18.10 operating system.

## 6. Results

The results of our experimental analysis are reported in Table 1, in which the following aggregate indicators (w.r.t. the 50 tackled instances) are used:

- $\bar{E}$: mean energy consumption in the final solution, expressed in megajoules;
- $\bar{D}$: mean delay in the final solution, expressed in seconds;
- $\bar{I}$: mean percentage improvement of the final solution w.r.t. the corresponding first *ib* solution;
- *iter*: mean iteration at which ACO-rtECMP finds the best solution;
- *time*: mean time at which ACO-rtECMP finds the best solution;
- $\bar{B}$: mean percentage improvement of the final solution w.r.t. the best solution found by random search.

As expected, we can see in Table 1 that energy consumption decreases as more weight is given to it in the objective function. The opposite holds for delay. The highest value of $\bar{I}$ is obtained with $w_0 = 0.9999$, i.e., when all importance is given to energy consumption. This strongest improvement within the run of the algorithm does not depend on the




(a) $w_0 = 0.9999$ (b) $w_0 = 0.5$ (c) $w_0 = 0.0001$

Fig. 4: Scatter plots of the heuristic information ($\eta$) in a representative instance for each configuration of objective function weights.

total number of iterations performed. Indeed, on average, the total number of iterations completed are 252, 190 and 170 for $w_0 = 0.9999$, 0.5 and 0.0001, respectively. However, the best solution is found sooner in the first case, i.e., at iteration 92 on average. We can observe a trend on $\bar{I}$, *iter* and *time*: the highest the importance of delay, the smaller the improvement of the algorithm with respect to the first *ib* solution and the latest the final solution is found, both in terms of number of iterations and of time elapsed. In all cases, the algorithm does not find any improving solution for about one third of the computational time available, hence longer runs would be useless. Column $\bar{B}$ shows that ACO-rtECMP always outperforms the random search, suggesting that the instances are not trivial. However, the difference is again smaller when delay is more important.

The lower $\bar{I}$ values for $w_0 = 0.5$ and $w_0 = 0.0001$ can be explained by observing the values of the heuristic information $\eta$ with the different weight configurations. Indeed, this measure assumes very similar values when most weight is given to delay. As an example, Figure 4 reports a scatter plot of the values of $\eta$ across all nodes for each chosen setting of $w_0$ in a representative instance. In Figure 4a, where $w_0 = 0.9999$, $\eta$ varies in the whole range $[0, 1]$ and its values tend to be noticeably different from each other, although some denser clusters can be seen around values 0.35 and 0.65. In Figure 4b, where $w_0 = 0.5$, the range of $\eta$ is compressed towards the upper bound so that $\eta$ varies between approximately 0.5 and one for almost all nodes. Although being less evident, the same clusters appearing in Figure 4a are visible, thanks to the still remarkable importance of energy consumption which allows discriminating between nodes. In Figure 4c, in which $w_0 = 0.0001$, the extent of the compression is even more severe. Despite the presence of slightly more values lower than 0.5, the majority of points are shifted towards one.

Apparently, for $w_0 = 0.5$ and $w_0 = 0.0001$, the reduced variation of $\eta$ impairs its capability of guiding the search as it fails to bias the node selection probabilities. This forces each ant to perform a somehow more uniformly random selection of nodes until enough pheromone has been deposited and some convergence is achieved. This is confirmed by the fact that the increase of $w_1$ corresponds to a very large number of explored arcs (denoted as $n_a$): in case of $(w_0, w_1) = (0.9999, 0.0001)$, $n_a = 5$ millions arcs; in case of $(w_0, w_1) = (0.5, 0.5)$, $n_a = 18$ millions arcs; in case of $(w_0, w_1) = (0.0001, 0.9999)$, $n_a = 23$ millions arcs.

## 7. Conclusion

In this research, we developed a graph model and a meta-heuristic algorithm to address the rtECMP, which is the problem of minimizing the weighted sum of total delay and energy consumption given fixed train routing and precedences. In real-time traffic management, whenever operations are perturbed, an rtRTMP solver computes new routing and precedences that can be fed into an rtECMP solver. The latter computes a speed profile for each train traversing a control area in a given time horizon.

We modeled the rtECMP as a graph. A path that fulfills a set of constraints is a feasible solution. Each node of the graph is associated to a pre-computed partial speed profile for a train in a block section.

To solve the problem, we introduced ACO-rtECMP, an ant colony optimization algorithm that employs the aforementioned graph to construct solutions.

We tested ACO-rtECMP on instances derived from the peak-hour traffic in the French Pierrefitte-Gonesse control area. For each instance, we studied the improvement of the best ACO-rtECMP solution w.r.t. the best one found in the first iteration with a time limit of 30 seconds. First, to indicate the non-trivial nature of this problem, we showed how ACO-rtECMP results significantly outperform those of a random-search. Second, we assessed the impacts of

shifting the optimization priority from energy consumption to delay reduction. By focusing on energy consumption minimization, the most remarkable improvements are obtained. This seems to be due to the range of values assumed by the heuristic information used to guide ants' exploration of the search space. This is due to the fact that running times associated to many train-block section speed profiles being very similar to each other. This implies the inability of the algorithm to spot better and better feasible space regions throughout the run, and thus its stagnation in a low quality local optimum.

In future research, we will address whether a different definition of the heuristic information may improve the algorithm performance.

Another research direction consists of extending ACO-rtECMP with a local-search algorithm, which is generally known to improve solution quality in conjunction with meta-heuristics as ACO.

Future work will also be devoted to benchmarking the ACO-rtECMP results against an exact rtECMP solver, such as TDRC-MILP (Montrone et al., 2018), on different infrastructures.

Finally, an actual multi-objective definition of the rtECMP may be considered, instead of relying on a weighted sum of its two objectives.

## References

Dorigo, M., Stützle, T., 2004. Ant Colony Optimization. The MIT Press.

Galapitage, A., Albrecht, A.R., Pudney, P., Vu, X., Zhou, P., 2018. Optimal real-time junction scheduling for trains with connected driver advice systems. Journal of Rail Transport Planning & Management 8, 29 – 41.

Goverde, R.M., Bešinović, N., Binder, A., Cacchiani, V., Quaglietta, E., Roberti, R., Toth, P., 2016. A three-level framework for performance-based railway timetabling. Transportation Research Part C: Emerging Technologies 67, 62 – 83.

Li, W., Peng, Q., Wen, C., Wang, P., Lessan, J., Xu, X., 2020. Joint optimization of delay-recovery and energy-saving in a metro system: A case study from china. Energy 202, 117699.

Luan, X., Wang, Y., De Schutter, B., Meng, L., Lodewijks, G., Corman, F., 2018. Integration of real-time traffic management and train control for rail networks - part 2: Extensions towards energy-efficient train operations. Transportation Research Part B: Methodological 115, 72 – 94.

López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., Stützle, T., 2016. The irace package: Iterated racing for automatic algorithm configuration. Operations Research Perspectives 3, 43 – 58.

Montrone, T., Pellegrini, P., Nobili, P., 2018. Real-time energy consumption minimization in railway networks. Transportation Research Part D: Transport and Environment 65, 524 – 539.

Pachl, J., 2002. Spacing trains. Railway Operation & Control , 38–90.

Pellegrini, P., Marlière, G., Pesenti, R., Rodriguez, J., 2015. Recife-milp: An effective milp-based heuristic for the real-time railway traffic management problem. IEEE Transactions on Intelligent Transportation Systems 16, 2609–2619.

Pellegrini, P., Marlière, G., Rodriguez, J., 2014. Optimal train routing and scheduling for managing traffic perturbations in complex junctions. Transportation Research Part B: Methodological 59, 58 – 80.

Samà, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., Pacciarelli, D., 2016. Ant colony optimization for the real-time train routing selection problem. Transportation Research Part B: Methodological 85, 89 – 108.

Scheepmaker, G.M., Goverde, R.M., Kroon, L.G., 2017. Review of energy-efficient train control and timetabling. European Journal of Operational Research 257, 355 – 376.

Shift2Rail, 2019. X2rail-4: Advanced signalling and automation system - completion of activities for enhanced automation systems, train integrity, traffic management evolution and smart object controllers. https://cordis.europa.eu/project/id/881806/results. Last accessed April 6,2021.

Stützle, T., Hoos, H.H., 2000. Max–min ant system. Future Generation Computer Systems 16, 889 – 914.

Wang, P., Goverde, R.M., 2017. Multi-train trajectory optimization for energy efficiency and delay recovery on single-track railway lines. Transportation Research Part B: Methodological 105, 340 – 361.

Wang, P., Goverde, R.M., 2019. Multi-train trajectory optimization for energy-efficient timetabling. European Journal of Operational Research 272, 621 – 635.

Xu, Y., Jia, B., Li, X., Li, M., Ghiasi, A., 2020. An integrated micro-macro approach for high-speed railway energy-efficient timetabling problem. Transportation Research Part C: Emerging Technologies 112, 88 – 115.