



HAL
open science

Towards eco-aware timetabling: evolutionary approach and cascading initialisation strategy for the bi-objective optimisation of train running times

Aurélien Lejeune, Rémy Chevrier, Pierre Olivier Vandanjon, Joaquin Rodriguez

► To cite this version:

Aurélien Lejeune, Rémy Chevrier, Pierre Olivier Vandanjon, Joaquin Rodriguez. Towards eco-aware timetabling: evolutionary approach and cascading initialisation strategy for the bi-objective optimisation of train running times. IET Intelligent Transport Systems, 2016, 10 (7), pp. 483-494. 10.1049/iet-its.2014.0309 . hal-01578913

HAL Id: hal-01578913

<https://hal.science/hal-01578913>

Submitted on 30 Aug 2017

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.

This paper is a postprint of a paper submitted to and accepted for publication in IET Intelligent Transport Systems.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. For citation purposes, please use the doi provided on the Digital Library page.

The copy of record is available at IET Digital Library

Towards eco-aware timetabling: evolutionary approach and cascading initialization strategy for the bi-objective optimization of train running times

Aurélien Lejeune¹, Rémy Chevrier^{1,2*}, Pierre-Olivier Vandanjon³, Joaquin Rodriguez¹

¹ Université Lille Nord De France, Ifsttar – COSYS – ESTAS, F-59666 Villeneuve-d'Ascq, France

² SNCF Innovation & Recherche, 40 avenue des Terroirs de France, F-75611 Paris Cedex 12, France

³ Université LUNAM, Ifsttar – AME – EASE, F-44341 Bouguenais, France

{aurelien.lejeune ; pierre-olivier.vandanjon ; joaquin.rodriguez}@ifsttar.fr

remy.chevrier@sncf.fr

Abstract

In railway planning, the timetabling step needs, as input, the train running times, which are calculated from a train dynamic model. Usually, this model determines the most energy-efficient train trajectory for a predefined time. However, this time may not correspond to the timetable-makers' needs. They should have the choice among a set of solutions, more or less energy-consuming. This paper proposes a method capable of producing a set of alternative running times with the associated mechanical energy required. To this end, our contribution is to set up an efficient evolutionary multi-objective algorithm builds a set of well-spread and diversified solutions which approximate a Pareto front. The solutions are all compromises between running time and energy-consumption, the two minimization objectives concurrently optimized. Given that an evolutionary algorithm is strongly dependent on the initialization phase, the efficiency of the algorithm is improved through a specific and original mechanism connecting multiple initializations in cascade in order to accelerate the convergence towards the best solutions. A set of results obtained on randomly-generated instances is analyzed and discussed.

Keywords: Running time, railway timetabling, energy-saving, multi-objective, optimization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

1 Introduction

Over the last decade, the traffic volumes in Western Europe have significantly grown and have consequently increased the energy consumption as well as the emission of pollutants. In order to face these environmental challenges, public policies promote the development of more eco-aware transportation systems. In railways, these policies lead to the promotion of the improvement of the rolling stock and the materials, but also tend to improve traffic management by using computer simulation and optimization. This optimization is multi-objective : it has to minimize the journey duration, the energy consumption and to maximize passengers confort under safety constraints. This paper contributes to set up a versatile software which helps the timetable-makers to manage efficiently and rapidly the compromise between energy and travel time.

The train running times between the stations are the input of the train timetables. Based on the timetables and the predefined running times, the energy-optimal train trajectories are computed offline and then transmitted to the train drivers in a roadmap. In this roadmap, the corresponding speed profile defines the speeds to be held at certain positions and it also indicates the switch-points from which the driver changes the driving regime. The precision of this switch depends on the drivers, who are trained in eco-driving in train simulation under supervision.

Given that the timetables are based on the train running times, modifying them will necessarily impact the timetables, but also the global energy consumption of the trains involved in the timetables under consideration. The difference in energy-consumption stems from the running speeds adopted and the driving regimes used, which are more or less energy-consuming. Producing alternative running times will allow the definition of a set of alternative energy-efficient timetables. Thus, one contribution of the paper is to find train speed profiles which are compromises between running time and energy consumption.

The problem of calculating the energy-optimal trajectory has been addressed in the literature in several ways and may involve several objectives, possibly including energy.

In the literature, a set of exact methods exists, based on the Maximum Principle and searching for a single optimal solution minimizing energy consumption for a travel with a predefined time. Liu and Golovitcher [17] solve the problem with an analytical method which calculates the sequence of optimal controls and the change points. A similar method is proposed by Albrecht [1], who solves differential equation systems. In order to find a solution, which is a compromise between energy consumption and riding comfort, Wang *et al.* [26] propose an optimization model based on a mixed integer program to optimize running times.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

The critical aspects of real-time railway management have also been addressed by a body of work to solve the optimal speed profile according to an available running time, such as in [18] or in [2] for multi-train scheduling. Su *et al.* [23] propose an algorithm based on the Maximum Principle for calculating the optimal speed profile for a fixed trip time in automatic train operation. Their algorithm is applied to the Beijing subway line and can be extended for generating a timetable.

Without exploiting the optimal control theory, Miyatake *et al.* [15, 20] use dynamic programming to calculate the energy-optimal trajectory. Lu *et al.* [19] optimize a single-train trajectory and compare three optimization algorithms: ant colony optimization, dynamic programming and genetic algorithms.

The methods mentioned before are constructed with the aim of giving a single solution, whereas the timetable-makers may be interested in alternative solutions, which imply having several objectives under consideration. To the best of our knowledge, there are still few multi-objective approaches to optimizing speed profiles. Evolutionary methods are used to optimize several objectives at once, as in [5] where the authors use Differential Evolution [22] for mass transit systems involving three objectives: punctuality, energy consumption and passenger comfort (by reducing the jerks). In [6, 7], the authors use evolutionary algorithms for building speed profiles from a set a predefined rules used as templates.

During the timetable planning, time supplements and buffer times are added to the fastest running times for calculating the journey duration between stations. These time allowances enable the compensation of some processes' variations during a journey like processes of driving or alighting and boarding passengers. Moreover, these time allowances also reduce the interaction between lines following perturbations. The time supplement is a margin [13], which can be used to save energy. Indeed, energy-friendly driving regimes, like coasting, which needs more time to ride the same distance, can be inserted in the speed profile to limit the energy consumption thanks to this time supplement.

In this paper, an upper bound of the time supplements and buffer times are considered as inputs. This upper bound is a fixed percentage of the fastest running times (10 % for our experiments). The final time allowances can be allocated on the results of the original method we propose for building train speed profiles between two stations in a bi-objective way. The objectives considered are the minimization of the running time and the minimization of the energy consumption. Based on the shortest running time, the proposed method builds a set of alternative solutions for this trip, which are different in time spent and energy consumed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

To perform the optimization, we use a state-of-the-art multi-objective evolutionary algorithm: the Indicator-based Evolutionary Algorithm (IBEA) [27]. In addition to the problem-related components of the algorithm, we have developed an original initialization strategy based on a cascading initialization mechanism. As far as we know, it is the first time that such a strategy is applied to a continuous and multi-objective railway problem.

The paper is organized as follows. First, Section 2 presents the train dynamics model as well as the driving regimes used to build the speed profiles. The problem is defined in Section 3 with the original method of constructing a speed profile. Section 4 presents the evolutionary algorithm proposed to perform the optimization as well as the specific initialization strategy developed for our problem. In Section 5, results obtained on a set of instances are presented and a comparison of three initialization strategies is also reported. Finally, Section 6 concludes the paper. An appendix gives an overview of the mathematical model of the problem.

2 Elements of train dynamics

This section explains how the running time and the mechanical energy consumption are estimated by using a dynamical model of the train and a rules-based model of the driver. This section is a short introduction, a more detailed explanation is proposed in [1]. Table 1 defines the symbols used in the remainder of the paper.

2.1 Track decomposition

In order to simplify the problem, a track is divided into a sequence of *sections*. This division follows two parameters:

- Signaling: a section is first defined by a length and a fixed maximal speed.
- Topography: if the gradient in a section changes more than an input user-defined threshold, the section is divided again into additional sections. Introducing a threshold is necessary to limit the number of sections and to avoid changing the driving regimes too frequently: it is uncomfortable for the passengers and also not practical for the train driver.

2.2 Railway dynamics

In the literature, a vehicle-point model is usually preferred to a realistic model involving too many constraints. The fundamental equation of dynamics states the relation between forces (\mathbf{F}),

**This article has been accepted for publication in a future issue of this journal, but has not been fully edited.
Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.**

Table 1: Main symbols used in train dynamics.

Symbol	Definition	Unit
$s(t)$	Position of the train at instant t	[m]
$v(t)$	Speed of the train at instant t	[m/s]
$a(t)$	Acceleration of the train at instant t	[m/s ²]
$u(t)$	Command of the train	-
$F_T(t)$	Tractive effort at instant t	[N]
$F_M(v)$	Maximal tractive effort, function of speed v	[N]
$F_R(t)$	Total resistance at instant t	[N]
$F_{Rw}(v)$	Running resistance, function of speed v	[N]
$F_{Rl}(s)$	Line resistance, function of position s	[N]
$F_B(t)$	Braking Force at instant t	[N]
F_{Bmax}	Maximum Braking Force	[N]
$q(s)$	Gradient of the slope, function of position s	-
$r(s)$	Radius of the curve, function of position s	[m]
m	Train mass	[kg]
ρ	Rotating mass correction factor	-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

mass (m) and acceleration (\mathbf{a}) applied to a point:

$$\sum \mathbf{F} = m \mathbf{a} \quad (1)$$

By taking into account motion resistances F_R , tractive efforts F_T and braking forces F_B , it yields:

$$F(t) - F_R(t) = \rho m a(t) \quad (2)$$

where ρ is the rotational inertia coefficient, representing effort consumed by the rotating parts of the vehicle. It usually ranges between 1.04 to 1.10 [25].

Where $F(t)$ refers to the following alternatives :

$$\begin{aligned} F(t) &= F_T(t) && \text{while the engine is providing traction} \\ F(t) &= F_B(t) && \text{while braking} \\ F(t) &= 0 && \text{while coasting} \end{aligned} \quad (3)$$

The tractive efforts F_T , the resistances F_R and braking F_B are modeled in the following subsections.

2.2.1 Tractive Effort

The vehicle engine generates a tractive effort F_T to move the train. This effort is usually limited by several conditions like the overheat limit or the adhesion of the wheel rim on the rail. Data about the tractive effort are usually given as the adhesion diagram (an example is shown in Figure 1). This diagram gives the adhesion limit of a train regarding the effort produced and the current speed v . So the maximum available effort $F_M(v)$ regarding a speed v can be deduced from that diagram as the limit of adhesion.

2.2.2 Braking system

In the timetabling step, emergency braking is not considered. Braking is carried out with service brakes i.e. dynamic brakes based on the dissipation of excess energy in rheostat devices. In this frame, the adhesion limit is compatible with the maximum service braking i.e. wheels are never blocked. Then, the maximum service braking is a constant, F_{Bmax} .

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

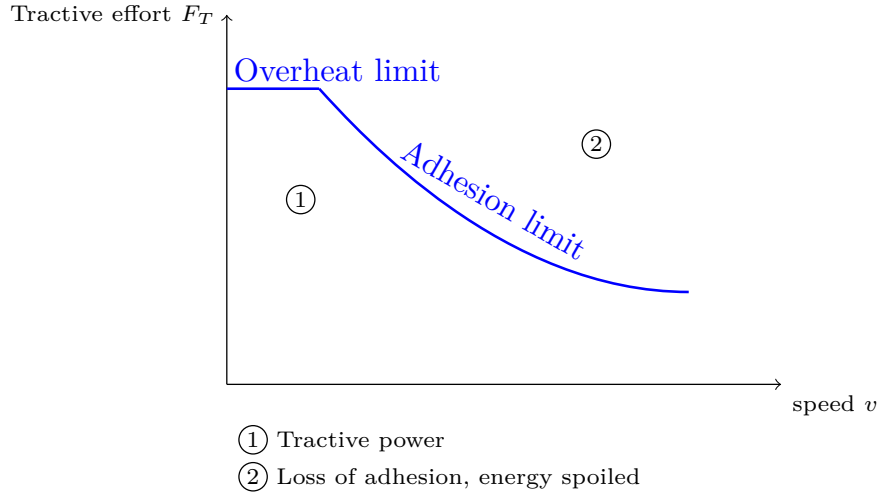


Figure 1: Schematic tractive effort curve: $F_M(v)$

2.2.3 Resistances

One of the most important losses of energy is due to the running resistance and the line resistance. So, we state the total resistance as the sum of the two previous resistance:

$$F_R(t) = F_{Rw}(v(t)) + F_{Rl}(s(t)) \quad (4)$$

The running resistance F_{Rw} is the sum of the air resistance (which is quadratic to the speed v) relative to the wind and the rolling resistance caused by adhesion loss and similar reasons. Those resistances can be described by the following parabola equation (known as the Davis' formula with mass included [8]):

$$F_{Rw}(v(t)) = A + Bv(t) + Cv^2(t) \quad (5)$$

where A is the mass-related coefficient of mechanical resistance (in N), B the viscous coefficient of mechanical resistance (Ns/m), and C the coefficient of aerodynamic resistance (Ns²/m²). Those parameters depend on the train characteristics.

Moreover, the line resistance F_{Rl} is caused by the topography. It depends on the curves and the slopes at position $s(t)$:

$$F_{Rl}(s(t)) = F_{Rlg}(s(t)) + F_{Rlc}(s(t)) \quad (6)$$

The resistance $F_{Rlg}(s(t))$ caused by the slope at position $s(t)$ is usually computed with the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

following equation with $\alpha(s(t))$ the angle of the slope at position $s(t)$:

$$F_{Rlg}(s(t)) = m g \sin(\alpha(s(t))) \quad (7)$$

However, given that $\tan \alpha$ and $\sin \alpha$ are almost equal for very small values of α , the resistance $F_{Rlg}(s(t))$ is often approximated as:

$$F_{Rlg}(s(t)) = m g q(s(t)) \quad (8)$$

with q the gradient of the slope ($q = \tan \alpha$). Those values are usually provided in meter per thousand in European technical data.

Finally, as the curve resistance F_{Rlc} increases as the curve radius decreases, it is usually expressed as a constant K divided by the radius of the curve r [21]:

$$F_{Rlc}(s(t)) = m g \frac{K}{r(s(t))} \cdot 10^{-3} \quad (9)$$

K values range between 500 to 1,200, depending on rail systems. According to this last equation, a curve of a K -meter radius ($r(s(t)) = K$) is equivalent to a 1‰ slope.

2.2.4 Energy consumption

The mechanical energy E consumed by the train during the trip is the integral of the mechanical power over the running time T :

$$E = \int_{t_0}^T F_T(t) v(t) dt \quad (10)$$

2.3 Driving behaviour

Based on the theory of optimal control, there are four optimal regimes defined by the maximum principle [1]: acceleration at full power (*Acc*), cruising at constant speed (*Cru*), coasting (inertia motion, *Coa*) and the maximum service braking (*Brk*).

Let $u(t)$ be the command used by the train at instant t and limited to these four regimes: {*Acc*, *Cru*, *Coa*, *Brk*}. The scheme depicted in Figure 2 describes a motion of the train using those four regimes over a sample section. Firstly, the train accelerates at full power until it reaches the speed v^{\max} . Then, it maintains its speed during the cruising phase. Afterwards, a coasting motion is engaged until the train has to brake to stop at the end of the section. By coasting from the position s_2 , the train saves more energy than coasting from s_1 but this driving increases the running time (the sooner the coasting, the greater the energy saved, but the longer the running time).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

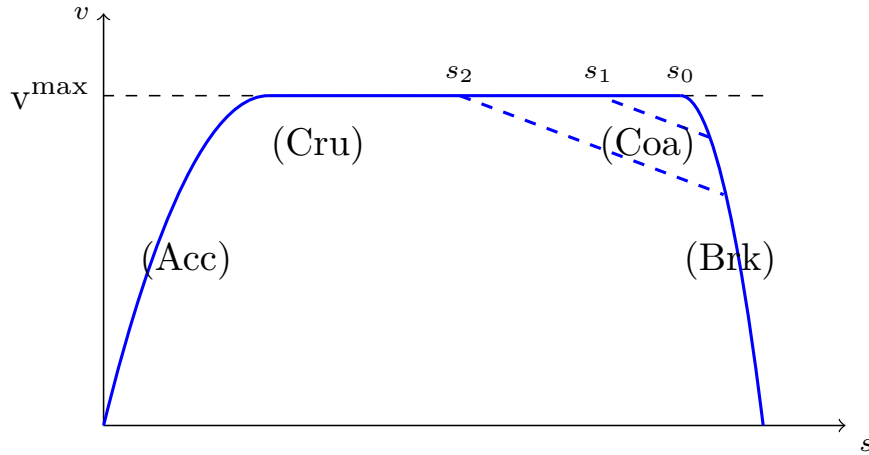


Figure 2: Sample speed profile over a section. Four regimes are noted: acceleration (Acc); cruising (Cru); coasting (Coa); braking (Brk).

2.3.1 Acceleration

In an acceleration phase, all the available tractive effort is used to reach a target speed. So, the tractive effort F_T is equal to the maximum effort available $F_M(v)$ depending on the tractive effort curve (Fig. 1). The command $u(t)$ of this regime is denoted Acc. The rule is hence defined:

$$u(t) = \text{Acc} \quad \Rightarrow \quad F_T(t) = F_M(v(t)) \quad (11)$$

2.3.2 Cruising phase

In a cruising phase, the objective is to maintain the speed constant (*i.e.*, $a(t) = 0$). The effort produced by the train must counterbalance the resistance: $F_T(t) = F_R(t)$.

As mentioned before, the resistance $F_R(t)$ is constituted by the line resistance $F_{Rl}(t)$ which depends on the gradient at position $s(t)$: $q(s(t))$. Figure 3 represents the line resistance F_{Rl} in function of the gradient q .

In some cases, it may happen that the objective of maintaining the speed is unreached when the train engages in a steep climb. In such a situation, the effort produced at full power may be insufficient to counterbalance the line resistance. Let q^+ be the threshold-gradient (and R the resulting resistance) over which the speed cannot be maintained due to an insufficient engine power, *i.e.*, $q(s(t)) > q^+$ and $F_M(v(t)) < F_R(t)$ (see Fig. 3, situation (3)). The train has to put the engine at full power to keep the speed as fast as possible. The driving rule is defined as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

follows:

$$(u(t) = \text{Cru}) \wedge (q^+ < q(s(t))) \Rightarrow (F_T(t) = F_M(v(t))) \wedge (a(t) < 0) \quad (12)$$

Conversely, a descent may be sufficiently steep to make the train accelerate without effort (according to the equation 2, $F_R(t) < 0$). Let q^- be the threshold-gradient under which the line resistance is negative: $q(s(t)) < q^-$ and $F_{RL}(s(t)) < 0$ (see Fig. 3, situation (1)). It has to be noted that the line resistance is nil when $q(s(t)) = q^-$.

If $q(s(t)) < q^-$, the train has to partially brake to maintain the speed. As the engine is not used, the energy consumption is nil. The driving rule is defined as follows:

$$(u(t) = \text{Cru}) \wedge (q(s(t)) < q^-) \Rightarrow (F_T(t) = 0) \wedge (a(t) = 0) \wedge (F_B(t) = F_R(t)) \quad (13)$$

In the interval between q^- and q^+ depicted in Figure 3 by the situation (2), the train can counterbalance the line resistance with a sufficient tractive effort. The driving rule is defined as follows:

$$(u(t) = \text{Cru}) \wedge (q^- < q(s(t)) < q^+) \Rightarrow (F_T(t) = F_R(t)) \wedge (a(t) = 0) \quad (14)$$

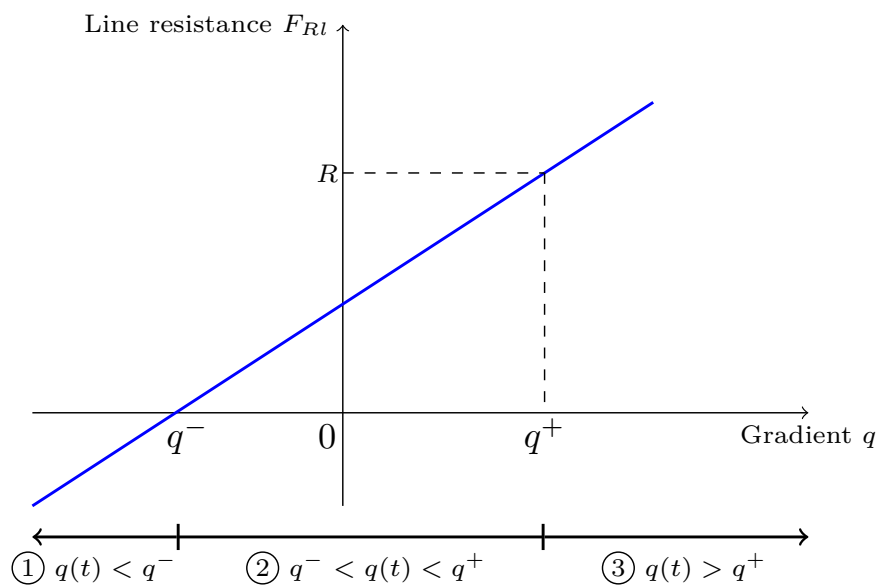


Figure 3: Line resistance F_{RL} in function of the gradient q , and the threshold-values q^-, q^+ . R reports the resistance limit that the tractive effort can counterbalance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

2.3.3 Coasting

While coasting, the vehicle moves by using only its own inertia. Neither the engine nor the braking are used. The coasting can be defined by the following equation:

$$u(t) = \text{Coa} \quad \Rightarrow \quad F_T(t) = 0 \wedge F_B(t) = 0 \quad (15)$$

As no traction is applied, no tractive energy is consumed in this phase.

2.3.4 Braking

While braking, the maximum service braking is used. This also means that no tractive effort is exerted:

$$u(t) = \text{Brk} \quad \Rightarrow \quad (F_T(t) = 0) \wedge (F_B(t) = F_{Bmax}) \quad (16)$$

2.3.5 Running time

By using the previous rules and equation 2 of the dynamic model, the running time, T , is computed according to the following equation.

$$\begin{aligned} T &= \int_0^T 1 dt & (17) \\ \dot{s}(t) &= v(t) \quad \dot{v}(t) = a(t) \quad a(t) = \frac{1}{\rho m} (F(t) - F_R(t)) \\ s(0) &= s_1^o \quad s(T) = s_n^d \quad v(0) = 0 \quad v(T) = 0 \end{aligned}$$

with s_1^o is the beginning and s_n^d is the end of the line.

3 Problem formulation

In this section, the model for building a speed profile sequentially within each section is presented in details. The first subsection formulates the problem objectives. The mathematical model based on a two-phase model is detailed in the second subsection. Finally, the generation of the speed profile following our model is explained in the third subsection. For the sake of clarity, Table 2 summarizes the main symbols used in this section. This section and the previous section exhibit a detailed presentation of the problem. An overview is proposed in the appendix.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

Table 2: Main symbols used in the problem formulation. Symbols in roman letter are parameters of the problem.

Symbol	Definition	Unit
T	Journey duration	[s]
E	Traction energy consumed at the end of the journey	[J]
n	Number of sections	-
i	Section subscript, $i \in [1..n]$	-
s_i^o	Beginning of the section i	[m]
s_i^d	End of the section i , $s_i^d = s_{i+1}^o$	[m]
v_i^{\max}	Speed limit of the section i	[m·s ⁻¹]
χ_i^1	Energy consuming phase	-
χ_i^2	Energy free phase	-
$\chi_i^{1 \rightarrow 2}$	Switch-position between the phases χ^1, χ^2 in the section i	[m]
v_i	Cruising speed along the section i	[m·s ⁻¹]

3.1 Formulation of objectives

The problem studied is formulated as a set Φ of two objective functions. The former is the minimization of the journey duration (T), and the latter represents the minimization of the traction energy consumed (E).

$$\Phi = \{\min T, \min E\} \quad (18)$$

In the case of mono-objective optimization, the minimum is a single value. In the actual case of two objectives, Φ is a curve in 2 dimension which is called the Pareto front.

3.2 Two-phase decomposition

The optimization problem consist in finding optimal controls $F(t)$ which minimize the previous objectives. This type of problem has been addressed from the optimal control theory point of view. In the field of the railway timetable design, It has been proven (see [1] for a complete explanation) that, by applying the Pontryagin's maximum principle, the optimal control proceeds in a section through the phases of Acceleration, Cruising, Coasting and Braking. With this result, it is shown that the initial optimization problem according to a function $F(t)$ becomes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

an optimization in a finite dimension space according to a finite number of decision variables which parameterizes $F(t)$. In the following, the parameterization of the optimal trajectory by two decisions variables by section is described.

We focus on practical solutions whose representation can be trivially transformed into indication for the driver. In this frame, each section i is divided into two phases: the energy-consuming phase (denoted χ_i^1) and the energy-free phase (χ_i^2). On the one hand, during the energy-consuming phase, the train only performs energy-consuming driving regimes, i.e., acceleration and cruising. On the other hand, coasting and braking are only performed in the energy-free phase.

With the aim of splitting each section i into these two phases, a pair of decision-variables is introduced. Let $\chi_i^{1 \rightarrow 2}$ be the phase switch-point within the section i and v_i the cruising speed to be reached in the first phase of the section i .

3.2.1 Insertion of a phase switch-point in the sections

The decision-variable $\chi_i^{1 \rightarrow 2}$ indicates the switch-point within the section i and is bounded as follows:

$$s_i^o \leq \chi_i^{1 \rightarrow 2} \leq s_i^d, \quad \forall i = 1, \dots, n \quad (19)$$

$\chi_i^{1 \rightarrow 2}$ allows the introduction of two distinct phases as illustrated by Figure 4:

1. Phase χ_i^1 : Only acceleration or cruising can be performed:

$$u(t) \in \{\text{Acc}, \text{Cru}\} \iff s_i^o \leq s(t) \leq \chi_i^{1 \rightarrow 2} \quad (20)$$

2. Phase χ_i^2 : Only coasting or braking can be performed (braking is engaged to respect the exit speed of the considered section):

$$u(t) \in \{\text{Coa}, \text{Brk}\} \iff \chi_i^{1 \rightarrow 2} \leq s(t) \leq s_i^d \quad (21)$$

3.2.2 Limitation of the cruising speed

A second variable v_i , the cruising speed, is introduced for each section. The cruising speed is defined as the maximum speed reachable with an acceleration during the first phase of the section. This defines the following constraints:

$$v_i \leq v_i^{\max}, \quad \forall i = 1, \dots, n \quad (22)$$

$$v(t) \leq v_i, \quad s_i^o \leq s(t) \leq \chi_i^{1 \rightarrow 2} \quad (23)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

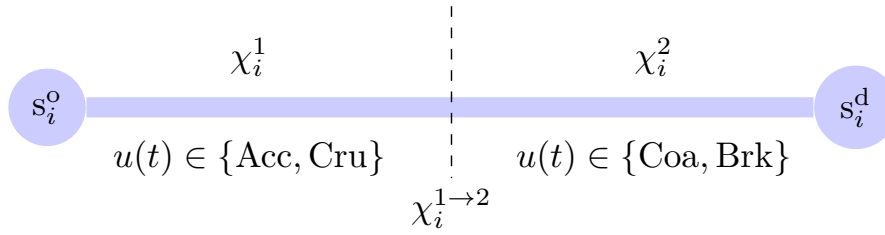


Figure 4: Section decomposition into two phases: χ_i^1, χ_i^2 . $\chi_i^{1 \rightarrow 2}$ is the switching position.

The cruising regime is applied in the first phase only if the speed v_i is reached. This yields the following equation:

$$u(t) = \text{Cru} \quad \iff \quad s_i^o \leq s(t) \leq \chi_i^{1 \rightarrow 2} \wedge v(t) = v_i \quad (24)$$

If the cruising speed v_i cannot be reached, the solution will be badly scored during the evaluation in the optimization process. Therefore, the solution will be discarded.

It has to be noted that the model is designed in a such way that the entrance speed in a section is always less than, or equal to, the cruising speed v_i . The traction energy consumed corresponds to efforts produced for the acceleration and the cruising regimes during the whole phase.

3.3 Speed profile generation

The speed profile is computed sequentially for each section. Given that the equation of motion is a second order *ordinary differential equation* (ODE), $\ddot{s} = f(u, s, v, t)$, and f is not an explicit function with a single variable, we choose a numerical method to generate the speed profile. As evaluating a speed profile needs to be computed quickly, a first order method (Euler integration), with a fixed step size Δt is chosen to approximate the solution. Δt has to be sufficiently small to ensure the stability of the method.

For the sake of clarity, the speed profile generation is illustrated by an example. Figures 5 describe a speed profile $v(t)$ versus the longitudinal position $s(t)$. The speed profile starts from the entering state (t_i^0, s_i^0, v_i^0) , which is the state of the system at the exit of the previous section ($i - 1, i > 0$). The two phases are computed sequentially as the state at the end of χ_i^1 has to be determined to compute the state χ_i^2 .

To compute the χ_i^1 phase, the train accelerates until it reaches the cruising speed v_i . After that, two alternative solutions are possible:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

- either the engine power can maintain the cruising speed until it reaches $s(t) = \chi_i^{1 \rightarrow 2}$ (Fig. 5a);
- or it is not possible to maintain the cruising speed due to the steep climb (Fig. 5b). In this case, a full engine power motion is performed until the gradient decreases sufficiently to maintain the speed, or until the position $\chi_i^{1 \rightarrow 2}$ is reached. If the cruising speed cannot be reached before $s(t) = \chi_i^{1 \rightarrow 2}$, no speed profile can be generated and the solution is, therefore, considered infeasible.

Then, the χ_i^2 phase is determined. Let v^{d*} be the exit speed-limit determined as follows:

- if the train has to stop at the end of the section, the exit speed-limit is set to 0: $v^{d*} = 0$;
- otherwise, the exit speed-limit is set to the next v_i value: $v^{d*} = v_{i+1}$.

A coasting is inserted until the end of the section (s_i^d) and two situations may occur:

- either the train speed at the end of the section is smaller than v^{d*} and the profile over the section is computed as depicted in Figure 5c;
- or braking is necessary after coasting to slow-down to the speed v^{d*} at the end of the section (Fig. 5d).

4 Optimization method

4.1 Multiobjective Evolutionary Optimization

In order to perform the construction and the production of multiple solutions at once, the optimization method is directly inspired from that proposed by Chevrier et al. [7]. In their work, the authors have chosen a state of the art evolutionary algorithm (EA) for the multiobjective optimization as the basis of their method: the *Indicator-Based Evolutionary Algorithm* (IBEA) which has the best performances according to Zitzler and Künzli [27]. The variation operators used in the cited paper are the same as those that we have chosen: the Simulated Binary Crossover (SBX) and the polynomial mutation [10].

Unlike methods optimizing a single value corresponding to a weighted sum of the objectives, a multiobjective algorithm offers the capability of optimizing the objectives concurrently. Concretely, in a multiobjective optimization, a solution is assigned a vector of objective values. Since

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

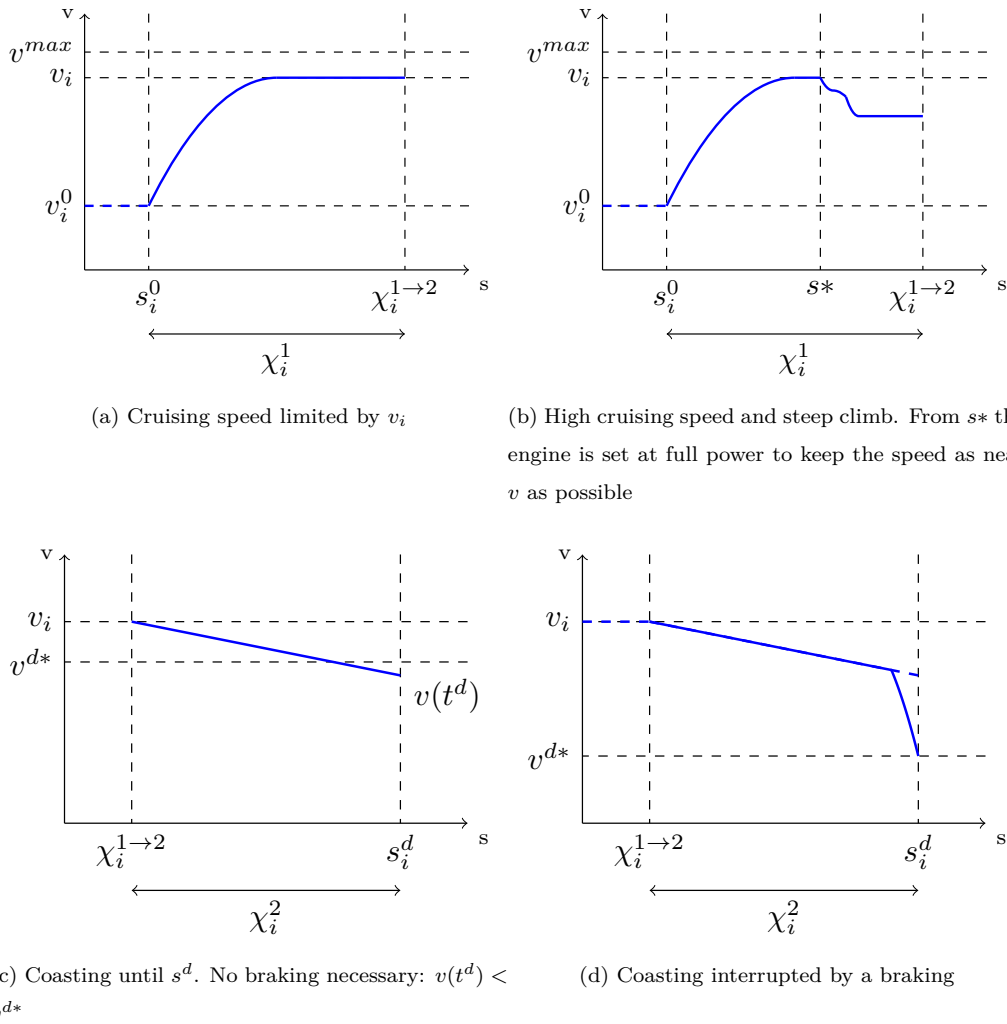


Figure 5: Motion occurring in χ_i^1 stage (a, b) and in χ_i^2 stage (c, d), according to a flat gradient

two objectives are to optimize in the problem under study, a solution matches a pair of values $\langle T, E \rangle$ in the two-dimension objective space (energy vs time). That allows the optimization of the objectives without having to use weights or to normalize objectives because they are of different nature, i.e., different units. Moreover, as optimizing a problem with more than one objective has no single solution but a set of Pareto-solutions [9, 11, 24], it is preferable to use a multiobjective optimization based on a Pareto approach, which produces a set of alternative solutions, which are all compromises between the objectives.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

4.1.1 Solution representation and evaluation

As previously stated, the decision variables of the problem are the phase switch-points $\chi_i^{1 \rightarrow 2}$, and the cruising speed v_i , of each section i . With a track composed of n sections, a vector of n pairs $(\langle \chi_1^{1 \rightarrow 2}, v_1 \rangle, \dots, \langle \chi_i^{1 \rightarrow 2}, v_i \rangle, \dots, \langle \chi_n^{1 \rightarrow 2}, v_n \rangle)$ defines a solution.

In order to evaluate a solution, the corresponding speed profile is built according to the procedure explained before. At the end, the values of the solution in the objective space are assigned to the vector $\langle T, E \rangle$ with T the running time and E the traction energy consumed during the journey. If a speed profile cannot be built regarding the constraints, the solution is marked as infeasible, and huge values are assigned to the objective vector in order to reject the solution (as the algorithm has to minimize the objective values).

4.1.2 Construction of the reference solution with a tight schedule

In order to ensure that the algorithm begins with at least one feasible solution, the fastest speed profile is added in the initial population. This tight-schedule solution (S_{TS}) can easily be computed according to the method described in [4]. Roughly speaking, it consists in driving as fast as possible, with respect to the speed limitations. It is done in three stages: first, by backwardly calculating the braking for each section from the last to the first. Secondly, the full power acceleration is calculated forward from the first section to the last. Finally, the gaps are filled with cruising motions. In the solution vector, v_i is the maximum reachable speed regarding the engine power and the speed limitation of the section i , and $\chi_i^{1 \rightarrow 2}$ is the position of braking (as coasting is never performed). An example of a tight schedule solution for two sections is provided in Figure 6.

4.1.3 Construction of the remaining population

The remaining population is simply filled with randomly generated solutions to add diversity to the initial population. A random solution is generated by randomly choosing each variable value inside its bounds. The upper-bound of each v_i variable is defined as the equivalent value in the S_{TS} solution (as we know we cannot go faster). A lower-bound greater than zero is arbitrarily chosen to avoid too slow motion (a good solution is to choose the lowest speed limit of the journey). The bounds of the $\chi_i^{1 \rightarrow 2}$ variables are simply set to s_i^o and s_i^d , respectively the beginning and end positions of each section. Last but not least, the S_{TS} solution represents the lower-bound of the running time objective, and the upper-bound of the energy objective. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

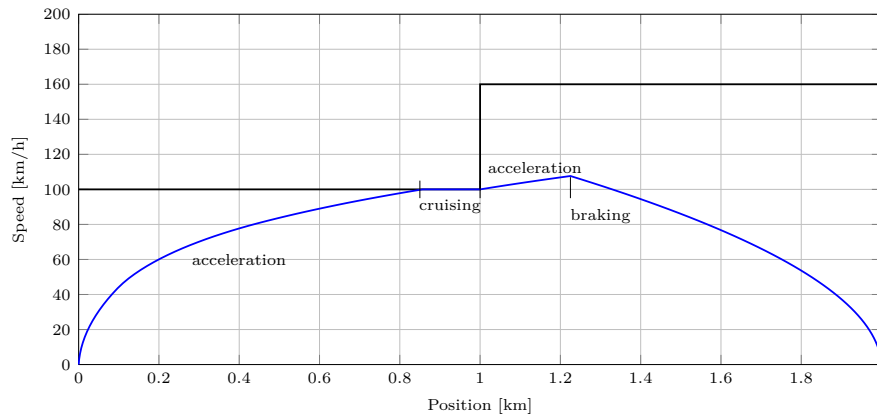


Figure 6: Tight schedule on a very short line. In the second section, the train cannot reach the speed limit because it needs to stop at the end.

upper-bound of the trip duration has to be introduced by the decision-maker in order to avoid ineffective solutions (too slow journey).

4.2 Cascading Initialization Strategy

4.2.1 Rationale and principles

In order to improve the convergence of the EA, it is preferable to start with an initial population including feasible solutions instead of solutions fully-randomly generated and maybe infeasible. Beginning an optimization with a good knowledge of the decision space increases the quality of the final solution set. However, properly exploring the decision space during the optimization is a time-consuming operation. Therefore, having an adapted initialization strategy often leads to better results in the same period of time.

The specific initialization strategy that we have developed and we call ‘Cascading Initialization Strategy’ (CIS) allows the construction of an enhanced initial population. It is a particular method of population initialization based on an iterative improvement of the solutions randomly generated according to the procedure described in Section 4.1.3 and a step-size relaxation in the ODE solved in the solution evaluation.

As we solve an ODE, the integration accuracy is dependent on the integration step Δt . Within the CIS process, we propose to start the optimization with a large Δt in order to converge more quickly to the most efficient part of the solution space. The integration is thus rough but rapid with a large Δt . It becomes more accurate but slower with a shorter integration step. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

process is performed a certain number of times by shortening the integration step each time.

Given that EAs are designed to offer good exploration possibilities, we use the EA itself for performing the CIS and initializing the population, but with a larger integration step (Δt) in the ODE. This initialization is done by optimizing a random population with a rougher precision. This optimization is obviously costly, but, as a major part of the exploration will have been performed, a non-negligible amount of time will be saved for processing the main optimization. One of the results of this paper is to demonstrate that this compromise is fruitful for our application.

To the best of our knowledge, it is the first time that such a strategy is used on this kind of problem.

4.2.2 Cascading connection of multiple initializations

Let P_N be the population obtained after N iterations (refinements) to be used in the main optimization process, and P_k the population at the iteration $k \in [1..N]$.

Roughly speaking, the CIS starts with a randomly generated population P_0 . Then, the CIS improves it by applying k times the EA previously described. At each iteration $k > 0$, the population P_{k-1} is used as input of the EA and the integration step $(\Delta t)_k$ is decreased by a relaxation function in such a way that the population P_k is evaluated more accurately than at the iteration $k - 1$. With such a mechanism, the CIS can explore a lot of solutions at the beginning and hence notice the bests of them. By performing several times the EA with a greater and greater accuracy in the construction of the speed profile, the CIS refines the solutions which require more and more computation time to be evaluated.

At the end of the initialization, the population P_N is composed of feasible solutions which can evolve in the main optimization EA with a fixed integration step Δt .

4.2.3 Parameter settings

In order to perform multiple initialization iterations, two parameters have to be defined for each iteration: the integration step size Δt , and the computation time available for an optimization process D^{opt} . As we deal with an initialization process, micro-tuning parameters, such as variation probability, do not seem useful. So, for each iteration, all the parameters related to the EA are the same as in the main optimization process.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

4.2.4 Main algorithm

The CIS algorithm is described by the algorithm 1. In the first step, an initial random population is generated with the initial step size $(\Delta t)_0$. Let f be a relaxation function defined as follows:

$$f(k) \rightarrow ((\Delta t)_k, D_k^{opt})$$

As the relaxation function is a decreasing function, the step size $(\Delta t)_0$ is the biggest defined in f . Then, the tight schedule solution (S_{TS}) is computed and added to this initial population P_0 . Afterwards, an optimization process is performed on this population. This process is repeated on the obtained population as often as necessary. At each iteration (k), the step size $(\Delta t)_k$ is updated according to the relaxation function f .

Once the exploration stage is finished, the initial population has to be built using the best individuals from the previously computed population. The best solutions have to be corrected according to a smaller integration step. This is a critical point of the proposed method. Indeed, the values v_i may risk getting outside the bounds and leading to an infeasible solution. Furthermore, as it is an initialization procedure, the correction has to be done efficiently. Therefore, the correction only prunes the values of the solution vector to its bounds.

Furthermore, a new S_{TS} is computed to determine the new bounds of the relaxed problem, and added to the current population.

Lastly, the final population P_N is used as the initial population of the main algorithm.

We can observe that neither the step-size relaxation nor the cascade-connection makes assumptions about the nature of the problem formulation. Thereby, this initialization strategy can be applied to any problem including ODE.

5 Experimental results

In this section, several cascading initialization strategies are compared with the basic initialization strategy. This is done by applying our proposed dynamic model on different railway lines. In the first part, details are given about the implementation and the set-up of parameter values presented before. Thereafter, a description of the lines and rolling stock is provided. Then, the different relaxation functions used in this work are presented and a comparison of the initialization strategies is presented and discussed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited.
Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

Data: EA : an evolutionary algorithm,
 $f(k) \rightarrow ((\Delta t)_k, D_k^{opt})$: relaxation function,
 N : number of refinements,
 p : population size,
 D_{All}^{opt} : time allowed to the whole optimization process

Result: P_N : result population to be used as input of the main optimization process,
 D^{opt} : time allowed to the main optimization process

- 1 Generate P_0 randomly according to $\Delta t = (\Delta t)_0$ and $|P_0| = p - 1$;
- 2 Compute and add $S_{TS}((\Delta t)_0)$ to P_0 ;
- 3 Call $f(1)$ to compute $(\Delta t)_1$ and D_1^{opt} ;
- 4 **for** $k = [1..N]$ **do**
- 5 Apply EA to P_{k-1} during D_k^{opt} ;
- 6 Let P_k be the set of $p - 1$ best solutions at the end of the k th optimization;
- 7 Evaluate and correct P_k according to $(\Delta t)_k$;
- 8 Compute and add $S_{TS}((\Delta t)_k)$ to P_k ;
- 9 Call $f(k + 1)$ to compute $(\Delta t)_{k+1}$ and D_{k+1}^{opt} ;
- 4 **end**
- 10 $D^{opt} \leftarrow D_{All}^{opt} - \sum D_k^{opt}$;
- 11 **return** (P_N, D^{opt}) ;

Algorithm 1: Cascading Initialization Strategy by relaxing the integration step size

5.1 Implementation

The algorithms previously presented are implemented by using the ParadisEO framework [16]. This framework works as a ‘white box’ with several optimization components implemented, and only problem-related components have to be designed by the users. So, the SBX, the polynomial mutation and IBEA are already implemented in ParadisEO. The computations are performed on a Quad-Core Xeon with 3Ghz CPU and 6GiB memory running Linux operating system.

5.2 Parameters setting

The population is composed of 100 individuals and evolves during 30 seconds of computation. The parameter values of the variation operators have been determined empirically with preliminary experiments: the mutation probability p_m and the crossover probability p_c are respectively fixed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

to 0.5057 and 0.8853 . Let \underline{T} be the shortest running time, i.e., the lower-bound of the running time objective. The upper-bound \overline{T} of the running time objective is set to 1.1 times the shortest travel time: $\overline{T} = 1.1 \times \underline{T}$

5.3 Rolling stock description

Two different kinds of trains have been chosen: AGC¹ for regional lines [3] and TGV for high speed lines [14]. The relevant characteristics of these trains are given in Figure 7.

5.4 Generation of the instances

To evaluate our algorithm and compare it with a random generation of solutions, ten instances have been randomly generated. These instances are separated into two classes:

- 5 regional line instances (with an AGC),
- 5 high speed line instances (with a TGV).

The lines are randomly generated by varying both the speed limitations along the track and the gradient to exhibit a rugged topography and a lot of sections. At the very beginning of the track and at the end (corresponding to the departure/arrival from/at the station), the speed limitations are quite low to those observed along the track, e.g., less than 100 km/h vs around 250 km/h, for a high-speed line. It has to be noted that the number of speed limitation changes implies the number of sections. The gradient varies randomly between -40% and 40% .

For a regional line, the instances are nearly 100 km long divided into approximatively twenty-five sections. The speed is limited to 160 km/h at most.

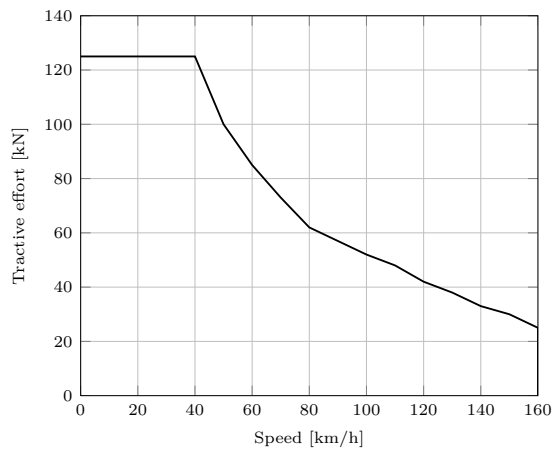
The TGV instances are three times longer (300 km) and divided into nearly sixty sections. The speed is limited to 300 km/h at most.

¹ *Autorail Grande Capacité*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

Parameter	Value	Unit
m	135,000	[kg]
ρ	1.04	-
A	2540	[N]
B	3.34	[N s/m]
C	0.49	[N s ² /m ²]

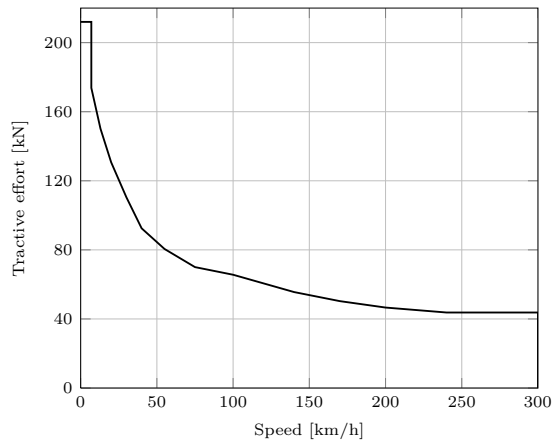
(a) Numerical parameters of AGC



(b) Tractive effort diagram of AGC

Parameter	Value	Unit
m	380,000	[kg]
ρ	1.04	-
A	2540	[N]
B	33.44	[N s/m]
C	0.572	[N s ² /m ²]

(c) Numerical parameters of TGV



(d) Tractive effort diagram of TGV

Figure 7: Technical data of the rolling stock used in the experiments

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

5.5 Comparison of the relaxation functions

In this subsection, three relaxation functions are compared on a simple random initial population. These functions are defined according to three values:

- $(\Delta t)_n$ the integration step of the main optimization process;
- D_{All}^{opt} the total computation time given for the whole optimization process (with the initial population generation included),
- and $n \in \mathbb{N}^+$ the number of cascading connections.

5.5.1 $f_{\log 2}$

This function relaxes the integration step (Δt) and determines the computation time D^{opt} according to based-2 logarithm functions:

$$(\Delta t)_k = 2^{(n-k)} \times (\Delta t)_n \quad (25)$$

$$\begin{cases} D_n^{opt} = D_{All}^{opt}/2 \\ D_k^{opt} = D_{k+1}^{opt}/2 \\ D_1^{opt} = D_{All}^{opt}/2^{n+1} \end{cases} \quad (26)$$

With this function, the time available for the main optimization is $D^{opt} = D_{All}^{opt}/2$.

5.5.2 f_{lin}

This function linearly decreases the value of (Δt) and D^{opt} :

$$(\Delta t)_k = (n - k + 1) \times (\Delta t)_n \quad (27)$$

$$D_k^{opt} = D_{All}^{opt}/(n + 1) \quad (28)$$

5.5.3 f_{lin2}

This function linearly decreases the value of (Δt) with $(\Delta t)_k \in [(\Delta t)_n, 2 \cdot (\Delta t)_n]$. The definition of D_k^{opt} is the same as the previous function:

$$(\Delta t)_k = (2 - k/n) \times (\Delta t)_n \quad (29)$$

$$D_k^{opt} = D_{All}^{opt}/(n + 1) \quad (30)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

5.5.4 Methodology

The three relaxation functions are compared using random initial populations. Since beginning with a standard random population fails to give good results in 30 seconds of optimization, we add the tight schedule solution to this initial population. This method is denoted *Random* in the following. Furthermore, the three relaxation functions are compared using a different number of cascading connections, with n varying between 1 and 3. This leads to ten different initialization strategies to be analyzed. Moreover, for each instance and for initialization function, ten runs are performed. At all, each instance is solved 100 times.

In order to compare these initialization strategies, the protocol proposed by Fonseca [12] has been followed. For a given instance, let \mathcal{Z}^{all} define the union of every solution obtained at the end of the experiments. Then, we compute \mathcal{P}^* the set of all the non-dominated solutions in \mathcal{Z}^{all} . Two reference points are then defined: z^{min} and z^{max} which are respectively the lower and upper-bounds of the solutions in \mathcal{Z}^{all} . Finally, all the objective values are normalized according to z^{min} and z^{max} .

The quality of a given solution set \mathcal{S} is computed regarding \mathcal{P}^* . In Figure 8, the ‘+’ (set \mathcal{P}^*) or ‘o’ (set \mathcal{S}) plots represent solutions in the normalized objective space whatever the indicator under consideration. The set of the ‘o’ solutions is dominated by the set of the ‘+’ solutions. The indicators work as follows:

- the unary additive epsilon metric ($I_{\epsilon+}^1$) computes the minimal translation to apply on \mathcal{S} to dominate \mathcal{P}^* (Fig. 8a illustrates this translation). ;
- the hypervolume difference indicator I_{HD} calculates the volume of objective space (bounded by z^{max}) dominated by \mathcal{P}^* but not by \mathcal{S} (Fig. 8b).

The reader interested in more details about the indicators chosen and the performance assessment can refer to the following references: [27, 12].

5.5.5 Results

Table 3 exhibits the difference in quality between each method regarding the two previous indicators. For each instance, the quality value indicated is the average quality value of the ten runs. Without loss of generality, since the objectives have to be minimized, the best quality values are the smallest.

It can be seen that the *Random* generation gives most of the time one of the worst sets of Pareto approximation. Moreover, the *Random* method can be unable to give feasible solutions,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

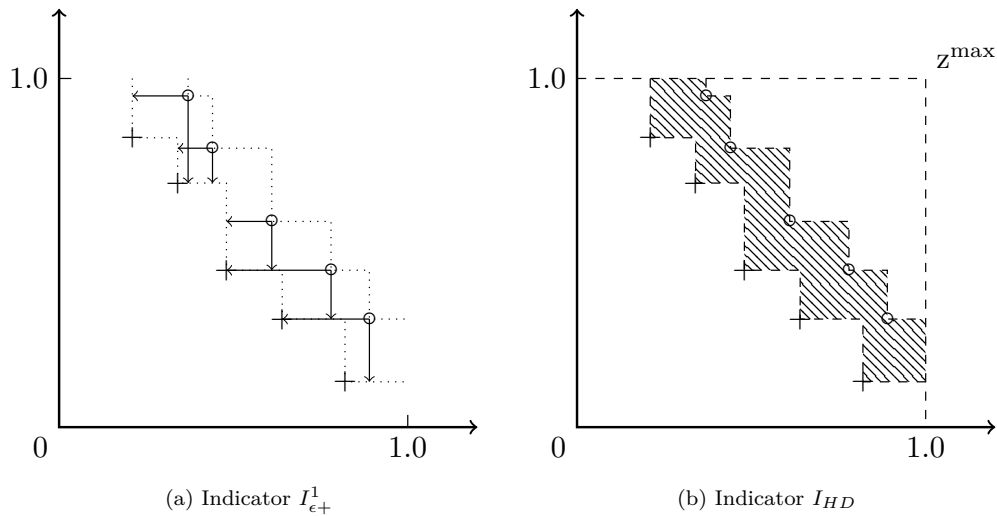


Figure 8: Comparison of quality of two Pareto front. Two quality indicators are depicted: $I_{\epsilon+}^1$ and I_{HD} . The ‘o’ set (\mathcal{S}) is compared with a reference front ‘+’ (\mathcal{P}^*).

except the tight schedule solution S_{TS} in the time granted, e.g., the results obtained with the random initialization on the TGV-4 instance: 1.0 for the $I_{\epsilon+}$ indicator and 1.2 for the I_{HD} indicator, which are the worst values indicating that no front has been found (at least two solutions).

In contrast, the best results are often given with the 3-step f_{log2} relaxation. Also, we can see that the ranking given by the two indicators is nearly the same. But reducing the evaluation time with the (Δt) relaxation allows a greater exploration of the decision space, and getting more feasible solutions, even if, in some cases, the *Random* method can give the best results (see instance AGC-4). Relaxing the integration step greatly increases the quality of the final solution set.

Globally, we can see that all the problem instances have been solved in 30 seconds by using the CIS and the sets of solutions have been produced in this period of time. As example, the solutions of the AGC-1 instance, depicted in Figure 9, have been obtained over the 10 runs by using the 3-step f_{log2} relaxation. These solutions are all non-dominated and are well-spread and diversified. That means a lot of solutions, compromises between the two objectives under consideration, have been found and are spread along a line which approximates the Pareto front. That characterizes a good quality of the set of solutions. With such a tool, which computes a lot of alternative solutions optimizing two different objectives, a decision-maker will be able to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

choose the speed-profile, and the running time associated, the most adapted to his needs when constructing a timetable. Indeed, the decision-maker will be able to examine the objective values and choose a solution following rules which cannot be mathematically formalized, as politics for example.

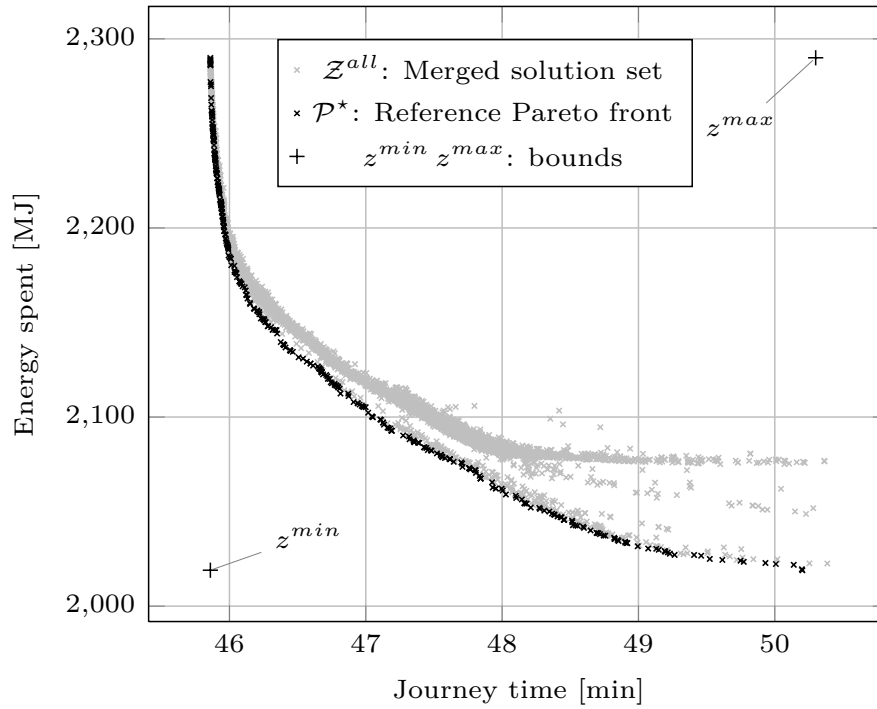


Figure 9: The reference Pareto front determined from merging all efficient solutions obtained when solving the AGC-1 instance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

Table 3: Comparison of the random initialization and three {1; 2; 3}-step relaxation functions regarding the $I_{\epsilon+}$ and I_{HD} indicators. In green the best values, in red the worst. The ‘{AGC; TGV}-avg’ rows are the average of the 5 instance values.

Ind. $I_{\epsilon+}$	<i>Random</i>	f_{log2}			f_{lin2}			f_{lin}		
		1	2	3	1	2	3	1	2	3
AGC-1	0.212	0.212	0.199	0.046	0.212	0.223	0.244	0.213	0.225	0.208
AGC-2	0.514	0.431	0.457	0.400	0.433	0.467	0.451	0.430	0.497	0.488
AGC-3	0.357	0.317	0.434	0.448	0.320	0.297	0.332	0.317	0.419	0.441
AGC-4	0.249	0.259	0.255	0.286	0.259	0.263	0.260	0.258	0.267	0.268
AGC-5	0.314	0.262	0.259	0.062	0.263	0.255	0.185	0.272	0.238	0.215
AGC-avg	0.329	0.296	0.321	0.248	0.298	0.301	0.294	0.298	0.329	0.324
TGV-1	0.579	0.579	0.580	0.528	0.580	0.574	0.546	0.579	0.581	0.583
TGV-2	0.415	0.174	0.184	0.188	0.176	0.308	0.354	0.179	0.136	0.158
TGV-3	0.535	0.426	0.338	0.189	0.427	0.409	0.371	0.428	0.317	0.326
TGV-4	1.000	0.229	0.150	0.141	0.206	0.216	0.404	0.206	0.174	0.107
TGV-5	0.417	0.357	0.172	0.188	0.356	0.268	0.272	0.356	0.154	0.173
TGV-avg	0.487	0.353	0.285	0.247	0.349	0.355	0.389	0.350	0.272	0.269

Ind. I_{HD}	<i>Random</i>	f_{log2}			f_{lin2}			f_{lin}		
		1	2	3	1	2	3	1	2	3
AGC-1	0.131	0.133	0.125	0.031	0.133	0.143	0.159	0.133	0.143	0.135
AGC-2	0.342	0.270	0.290	0.231	0.276	0.299	0.289	0.269	0.317	0.318
AGC-3	0.272	0.242	0.326	0.336	0.242	0.226	0.251	0.239	0.315	0.330
AGC-4	0.154	0.159	0.157	0.177	0.159	0.163	0.163	0.159	0.164	0.169
AGC-5	0.247	0.201	0.199	0.043	0.202	0.193	0.137	0.208	0.173	0.159
AGC-avg	0.229	0.201	0.219	0.164	0.202	0.205	0.200	0.202	0.222	0.222
TGV-1	0.402	0.402	0.402	0.363	0.402	0.399	0.377	0.402	0.403	0.405
TGV-2	0.194	0.060	0.068	0.069	0.062	0.158	0.162	0.064	0.050	0.063
TGV-3	0.364	0.273	0.213	0.115	0.274	0.266	0.238	0.275	0.192	0.205
TGV-4	1.200	0.145	0.115	0.107	0.133	0.157	0.273	0.135	0.115	0.082
TGV-5	0.216	0.181	0.071	0.080	0.180	0.121	0.131	0.180	0.047	0.077
TGV-avg	0.294	0.212	0.174	0.147	0.210	0.220	0.236	0.211	0.161	0.166

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

6 Conclusion and Perspectives

In this paper, we deal with the problem of calculating the train running times. This calculation is the first step in railway planning, because it precedes train timetabling. In order to contribute to the construction of energy-efficient train timetables, we have proposed to use alternative train running times which are more or less energy-consuming. Determining these alternative running times is possible by using the method proposed in this paper, which is based on two main contributions. The first consists in constructing speed profiles by using an original method based on an evolutionary computation able of optimizing the two antagonist objectives at once. The second contribution concerns the algorithm performance increased by the specific cascading initialization strategy that we have conceived.

The quality of the results obtained allows us to consider the integration of this approach into the planning process. It is indeed interesting to envisage developing timetabling methods taking the energy criterion into account, i.e., the alternative running times which are more or less energy-consuming. With such methods, the decision-makers will be helped by new decision tools showing the advantages and drawbacks of the possible solutions for the railway service with the associated costs, energy consumed, and quality of service.

Another possible extension consists in adding the power recovery in braking. This extension assumes to have data on the catenary and the train's rheostat devices in particular, but also on the primary power-supplier if it is fed by the energy recovered. Applied to the traffic management, the braking-power recovery is relevant in the case of a multi-train scheduling, where we search for feeding one train with the power produced by another train which is braking in a close range. Such a functionality will lead to a new field of traffic simulation and management methods.

References

- [1] Thomas Albrecht. Energy-efficient train operation. In Ingo Arne Hansen and Jörn Pachl, editors, *Railway Timetable & Traffic*, chapter 5, page 83–105. Eurailpress, 2008 edition, 2008.
- [2] Thomas Albrecht and Steffen Oettich. A new integrated approach to dynamic schedule synchronization and energy-saving train control. *Publication of: WIT Press*, 2002.
- [3] Jean-Marc Allenbach. *Traction électrique*. PPUR Presses Polytechniques, 2008.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

- [4] Olaf Brünger and Elias Dahlhaus. Running Time Estimation. In Ingo Arne Hansen and Jörn Pachl, editors, *Railway Timetable & Traffic*, chapter 4. Eurailpress, 2008 edition, 2008.
- [5] Cheng-Shang Chang, D. Y. Xu, and H. B. Quek. Pareto-optimal set based multiobjective tuning of fuzzy automatic train operation for mass transit system. *IEE Proceedings Electric Power Applications*, 146(5):577–583, 1999.
- [6] Rémy Chevrier. An Evolutionary Multi-objective Approach for Speed Tuning Optimization with Energy Saving in Railway Management. In *ITSC'2010, 13th IEEE Intelligent Transport Systems Conference*, page 279–284, Madeira, Portugal, 2010.
- [7] Rémy Chevrier, Paola Pellegrini, and Joaquín Rodríguez. Energy saving in railway timetabling: A bi-objective evolutionary approach for computing alternative running times. *Transportation Research Part C*, 37:20 – 41, 2013.
- [8] W.J. Davis Jr. Tractive resistance of electric locomotives and cars. *General Electric Review*, 29:685–708, 1926.
- [9] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.
- [10] Kalyanmoy Deb and Ram B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:1–34, 1994.
- [11] Matthias Ehrgott and Xavier Gandibleux. *Approximative Solution Methods for Multiobjective Combinatorial Optimization*, volume 12. 2004.
- [12] Carlos M. Fonseca, Joshua D. Knowles, Lothar Thiele, and Eckart Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. In *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 216, 2005.
- [13] Rob Goverde. *Punctuality of railway operations and timetable stability analysis*. PhD thesis, TU Delft, 2005.
- [14] Alain Jeunesse and Michel Rollin. La motorisation du TGV POS. *Revue Générale des Chemins de Fer*, 126, 2004.
- [15] Hideyoshi Ko, T. Koseki, and Masafumi Miyatake. Application of dynamic programming to optimization of running profile of a train. *Computers in Railways IX*, page 103–112, 2004.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

- [16] Arnaud Liefoghe, Laëtitia Jourdan, and El Ghazali Talbi. A unified model for evolutionary multi-objective optimization and its implementation in a general purpose software framework. In *Computational intelligence in multi-criteria decision-making, 2009. mcdm'09. ieee symposium on*, page 88–95. IEEE, 2009.
- [17] Rongfang Rachel Liu and Iakov Mikhailovich Golovicher. Energy-efficient operation of rail vehicles. *Transportation Research Part A: Policy and Practice*, 37(10):917–932, 2003.
- [18] Marco Lüthi. *Improving the Efficiency of Heavily Used. Railway Networks through Integrated. Real-Time Rescheduling*. PhD thesis, ETH Zurich, 2009.
- [19] Shaofeng Lu, Stuart Hillmansen, Tin Kin Ho, and Clive Roberts. Single-train trajectory optimization. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):743–750, 2013.
- [20] Masafumi Miyatake and Hideyoshi Ko. Optimization of Train Speed Profile for Minimum Energy Consumption. *IEEJ Transactions on Electrical and Electronic Engineering*, 5(3):263–269, 2010.
- [21] Vassilios A Profillidis. *Railway engineering*. Avebury Technical/Ashgate Publishing Limited, Aldershot, 2000.
- [22] Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
- [23] Shuai Su, Xiang Li, Tao Tang, and Ziyou Gao. A subway train timetable optimization approach based on energy-efficient operation strategy. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):883–893, 2013.
- [24] El Ghazali Talbi. *Metaheuristics: from design to implementation*. Wiley, 2009. 624 p.
- [25] Vukan R. Vuchic. *Urban public transportation: systems and technology*. Englewood Cliffs: Prentice-Hall, 1981.
- [26] Yihui Wang, Bart De Schutter, Bin Ning, Noortje Groot, and Ton J.J. van den Boom. Optimal trajectory planning for trains using mixed integer linear programming. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, page 1598–1604. IEEE, 2011.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

- [27] Eckart Zitzler and Simon Künzli. Indicator-based selection in multiobjective search. *Parallel Problem Solving from Nature (PPSN) VIII*, 3242(i):832–842, 2004.

Appendix: synthesis

This section displays a global mathematical view of the problem. For the sake of the clarity, the vector $(\chi_1^{1 \rightarrow 2}, v_1, \dots, \chi_i^{1 \rightarrow 2}, v_i, \dots, \chi_n^{1 \rightarrow 2}, v_n)$ is summarized by the notation $(\chi_i^{1 \rightarrow 2}, v_i)_{i=1..n}$

The energy objective E and the running time objective T are:

$$E((\chi_i^{1 \rightarrow 2}, v_i)_{i=1..n}) = \int_0^T F_T(t)v(t)dt \quad T((\chi_i^{1 \rightarrow 2}, v_i)_{i=1..n}) = \int_0^T 1dt$$

The optimization program is to find the set of solutions, called the Pareto front:

$$\Phi = \{ \min E((\chi_i^{1 \rightarrow 2}, v_i)_{i=1..n}), \min T((\chi_i^{1 \rightarrow 2}, v_i)_{i=1..n}) \}$$

The decision variables are: $\chi_i^{1 \rightarrow 2}, v_i$ with $i = 1..n$. They obey to the following constraints.

$$\chi_i^{1 \rightarrow 2} \in [s_i^o, s_i^d] \quad v_i \in [0, v_i^{\max}] \quad \forall i = 1, \dots, n$$

$[s_i^o, s_i^d]$ is the i^{th} section.

The computation of E and T is given by the following dynamic equations:

$$\dot{s}(t) = v(t) \quad \dot{v}(t) = a(t) \quad a(t) = \frac{1}{\rho m} (F(t) - F_R(t))$$

with the initial and final conditions:

$$s(0) = s_1^o \quad s(T) = s_n^d \quad v(0) = 0 \quad v(T) = 0$$

$F(t)$ and, by the way $F_T(t)$, are computed section by section according to the decision variables. In the i^{th} section (when $s(t) \in [s_i^o, s_i^d]$), the optimal trajectory proceeds sequentially through the phases of Acceleration, Cruising, Coasting and Braking. Each phase specifies the values of $F_T(t)$ and $F(t)$. The switches between the phases are controlled by a comparison between the state of the system and the decision variables according to the following rules.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

$$s(t) \leq \chi_i^{1 \rightarrow 2}$$

$$v(t) < v_i \left\{ \text{Acceleration} \right\} \quad F(t) = F_M(v(t)) \quad F_T(t) = F_M(v(t))$$

$$v(t) = v_i \left\{ \text{Cruising} \right\} \quad F(t) = F_R(t) \quad F_T(t) = \max(0, F_R(t))$$

$$s(t) \geq \chi_i^{1 \rightarrow 2}$$

$$v(s(t)) < v_i^{ls}(s(t)) \left\{ \text{Coasting} \right\} \quad F(t) = 0 \quad F_T(t) = 0$$

$$v(s(t)) = v_i^{ls}(s(t)) \left\{ \text{Braking} \right\} \quad F(t) = F_{Bmax} \quad F_T(t) = 0$$

$v_i^{ls}(s)$ is a limit velocity which is computed from the limit trajectory when the service brake is maximum. This limit trajectory starts from the end of the section i and is computed backwardly by using the maximum braking service according to the following equations:

$$\dot{s}_i^l(t) = v_i^l(t) \quad \dot{v}_i^l(t) = a_i^l(t) \quad a_i^l(t) = \frac{1}{\rho m} (F_R(t) - F_{Bmax})$$

$$s_i^l(0) = s_i^d \quad v_i^l(0) = v_{i+1}$$