



HAL
open science

Energy saving in railway timetabling: A bi-objective evolutionary approach for computing alternative running times

Rémy Chevrier, Paola Pellegrini, Joaquin Rodriguez

► To cite this version:

Rémy Chevrier, Paola Pellegrini, Joaquin Rodriguez. Energy saving in railway timetabling: A bi-objective evolutionary approach for computing alternative running times. *Transportation research. Part C, Emerging technologies*, 2013, 37, pp.20–41. hal-00742039v2

HAL Id: hal-00742039

<https://hal.science/hal-00742039v2>

Submitted on 11 Sep 2013

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.

Energy saving in railway timetabling: A bi-objective evolutionary approach for computing alternative running times

Rémy Chevrier*, Paola Pellegrini, Joaquín Rodriguez

*Université Lille Nord de France, IFSTTAR – ESTAS
20 rue Élisée Reclus, 59650 Villeneuve d'Ascq, France*

Abstract

The timetabling step in railway planning is based on the estimation of the running times. Usually, they are estimated as the shortest running time increased of a short time supplement. Estimating the running time amounts to define the speed profile which indicates the speed that the train driver must hold at each position. The approach proposed in this paper produces a set of solutions optimizing both the running time and energy consumption. The approach is based on an original method of speed profiling performed by a multi-objective evolutionary algorithm. The speed profiles found by the evolutionary algorithm are all compromises between running time on the one hand and energy consumption on the other hand. A set of results obtained on two lines are analyzed and discussed to highlight the relevance of such an approach in an practical context.

Key words: Running time, Energy saving, Railway timetabling, Multi-objective, Evolutionary algorithm, Optimization

1. Introduction

Over the years, the increase of traffic volumes in Western Europe and the increase of emission of pollutants have put a stronger and stronger accent

*Corresponding author

Email addresses: remy.chevrier@ifsttar.fr (Rémy Chevrier),
paola.pellegrini@ifsttar.fr (Paola Pellegrini), joaquin.rodriguez@ifsttar.fr
(Joaquín Rodriguez)

on the development of more eco-aware transportation systems. A direct consequence is the search for a better management of energy. In railways, this better management may be sought, first of all, in the planning phase, during which the timetables are built.

All information required for the trips are computed during the planning phase. In particular, the train running times between stations are estimated during this phase to then compute notably the timetable of each train. Based on the timetable and the predefined running times, the energy-optimal train trajectories are computed offline and then communicated to the driver in a roadmap that he must follow [1]. The corresponding speed profiles precise the speed at each position on the track and they give also the switch-points indicating to the driver when changing the driving regime. In order to efficiently switching the regimes, the train drivers are trained and accustomed to the energy-efficient driving, notably by learning it under supervision in a train simulator.

As the timetabling process is based on the predefined running times between stations, the definition of alternative running times will directly impact, of course, the traffic planning, but also the whole energy consumption. Indeed, changing the running speed or the driving regimes that the train must hold to respect the time constraints, will modify the energy consumption and lead to have potentially energy-efficient timetables. Thus, the goal is to find energy-efficient speed profiles compliant to the requirements (signaling, time constraints) so that alternative timetables are possible.

The problem of speed-profiling has been addressed in the literature in several ways, taking energy into account or not, involving one or several objectives.

For optimizing energy efficiency of train operation, there exist methods searching for a single optimal solution minimizing energy consumption for a given running time. The problem can be solved with an analytical method calculating the sequence of optimal controls and change points such as in [2], or with switch-points and differential equations systems as explained in [1].

Furthermore, due to the critical aspects of real-time railway management, a body of work has been carried out to solve the optimal speed profile according to an available running time, such as in [3] or in [4] for multi-train scheduling. It is also possible to re-optimize the running times in real-time at network scale [5].

More recently, a method based on mixed integer linear programming has been proposed to optimize running times by using an objective function which

is a tradeoff between energy consumption and riding comfort [6].

Differently to most of the works, trajectory optimization can be addressed without optimal control theory, as in [7, 8] where the authors exploit the dynamic programming to solve the problem.

Most of the methods are designed to provide a single solution to the decision-makers, though they could need more flexibility in the timetabling process.

As far as we know, there are still few multi-objective approaches to optimize speed profiles. Differential Evolution [9] has been used for mass transit systems [10] involving three objectives: punctuality, energy consumption and passenger comfort (by reducing jerks). Another evolutionary method has been developed to perform a speed-based model [11, 12], building speed profiles in a multi-objective way according to a set of predefined rules.

Although estimating the appropriate running time is crucial for the timetabling process, in practice, the running times are usually based on the fastest journey multiplied by an arbitrary factor slightly greater than 1 to have a running time supplement [13]. In general, the time supplement allows the train driver to adapt the speed to the traffic. However, the time supplements can be reduced by relaxing the timetables of some specifications (tracks, platforms, time windows, ...) for reducing the use of railway capacity [14].

If perturbations occur, it may happen that the running times have to be modified to take conflicts or delays into account. In [15], the authors propose to forecast delays to produce a distribution of possible running times to be used when the timetable cannot be respected. The authors solve systems of linear differential equations to estimate possible running times.

The time supplement could also be used to save energy. Indeed, we can identify the most energy-friendly speed profile, which benefits of this additional time for using energy-free driving regimes. The speed profile designed in this way differs from the one of the fastest journey in duration (longer) and energy consumption (weaker).

In this paper, we propose an original approach to compute train running times by concurrently minimizing both energy consumption and running time. Usually, the running time is estimated for an entire trip including a set of journeys between stations and the time supplement is spread over the trip [16]. However, as a preliminary work, we consider here the definition of running time between two stations. Since a trip includes a succession of journeys from one station to another, the approach proposed performs a running time calculation of one journey. This calculation is done by building

the speed profile according to a set of rules that we propose to determine the order of driving regimes that the train driver must follow. In particular, a braking must not be followed by an acceleration, because it is absurd from an energy consumption point of view. Moreover, the approach under consideration must be capable of providing a set of tradeoff-solutions for the decision-makers in a single run. In such a way, they will be able to choose a running time adapted to their needs in the timetabling process. Thus, the paper deals with a bi-objective optimization of speed profiling with energy saving. Given that evolutionary algorithms (EA) are well-suited to multi-objective optimization [17], our approach is based on a state-of-the-art multi-objective EA: the Indicator-Based Evolutionary Algorithm (IBEA) [18].

The paper is organized as follows. At first, Section 2 concerns the basic principles of train dynamics and running time calculation used in the optimization model. Then, Section 3 presents the problem under study and its formulation. The algorithms for building a speed profile and evaluating a solution are presented and detailed in Section 4. In Section 5, we present both the principles of multi-objective optimization and IBEA. The specific components of the algorithm are also presented in this section. Section 6 presents two case-studies (including one real railway line), as well as the results of speed profile optimization obtained on these instances. Analysis and discussion are provided for highlighting the interest of such a method in the planning process. Finally, Section 7 concludes the paper.

2. Running times and train dynamics

In this section, we explain how the running times are calculated and we recall the basic formulas of train dynamics used to perform this calculation. The reader can refer to [19, 20] to have a more detailed explanation of train dynamics. The formulas presented may be modified or replaced without changing the nature of optimization problem defined in the following. The model used here is an approximation of the reality. However, a more accurate model implying, for example, to consider the length of the train, the cars and their positions on the track, but also the axles and the wheels, requires to develop a more precise railway dynamics model.

Table 1 summarizes the main symbols used in the paper.

Table 1: Definition of the symbols used in train dynamics

T	journey duration	[s]
E	mechanical energy	[J]
v	train speed	[m/s]
$P(t)$	mechanical power at instant t	[W]
$F_T(v)$	tractive effort, function of speed v	[N]
$F_m(v)$	Maximal tractive, function of speed v	[N]
$F_R(v)$	sum of resistances	[N]
L_R	line resistance	[N]
C_R	curve resistance	[N]
$M_R(v)$	vehicle resistance, function of speed v	[N]
γ	train acceleration	[m/s ²]
b	braking	[m/s ²]
m	train mass	[kg]
ρ	mass correction factor	
β	angle of the slope	
q	gradient of the slope	[‰]
c	radius of the curve	[m]

2.1. Setting sequences of driving regimes

In order to define accurate running times, it is necessary to build speed profiles, which are indicated in the roadmaps that the train driver must follow. According to the theory of optimal control, there are four optimal regimes defined by application of the Maximum Principle (see [21, 1] for details): Acceleration at full power; Cruising at constant speed; Coasting (inertia motion while the engine is stopped); Maximum braking (according to the service braking, softer than emergency braking). Since acceleration is very energy-consuming, the inefficiency of applying unnecessary sequences of braking followed by acceleration is straightforward. Hence, it is a principle of the method that we propose in the paper. In the roadmaps to provide to the drivers, a braking must not be followed by an acceleration.

The problem we deal with consists in designing the most suited speed profile over the track. This track is composed of a sequence of *sections*, in which the speed has to be tuned. A section is defined by a length and a constant and fixed maximal speed. Consecutive sections always have different

maximal speed (see Fig. 1).

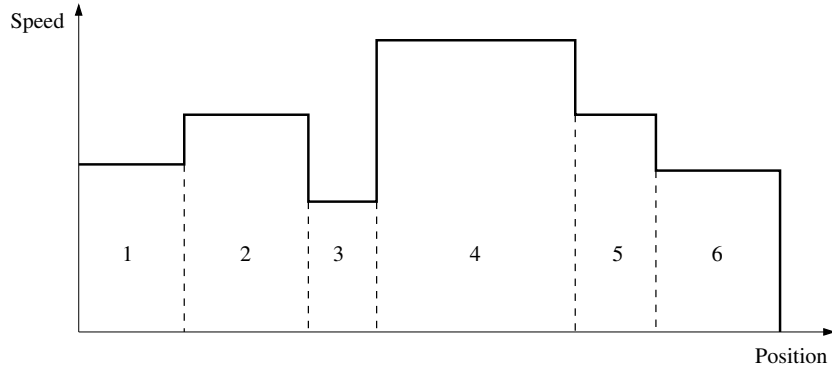


Figure 1: Decomposition of the track according to the maximal speeds

In principle, a one-section journey can be divided in four steps as depicted in Fig. 2 (we assume there is neither slope nor curve in this example). Let v_m be the maximal speed. First, the train accelerates (A) in order to reach speed v_m as quickly as possible. Then, a cruising phase (Cr) follows during which the acceleration is nil and the traction effort equals the resistance to the train advance. Given that the wheel/rail adhesion is weak, it is common to let the train coast over long distances [22, 23], e.g. points Co(1), Co(2) indicate two positions from which coasting can be started. Coasting from point Co(1) may increase the journey duration a little while reducing the use of mechanical energy. By coasting from point Co(2), the energy consumption may be further decreased with a consequent increase of journey duration. The sooner the coasting starts, the greater the economy, but the longer the journey duration. Point Co(0) indicates the last position from which the train can brake with its normal service braking (B) for being able to stop at the end of the section.

2.2. Elements of railway dynamics

The fundamental equation of dynamics states the relation between the forces F_T, F_R , speed v , mass m and acceleration γ :

$$F_T(v) - F_R(v) = \rho m \gamma, \quad (1)$$

ρ being a mass correction factor usually set to $\rho = 1.04$ [23, 1].

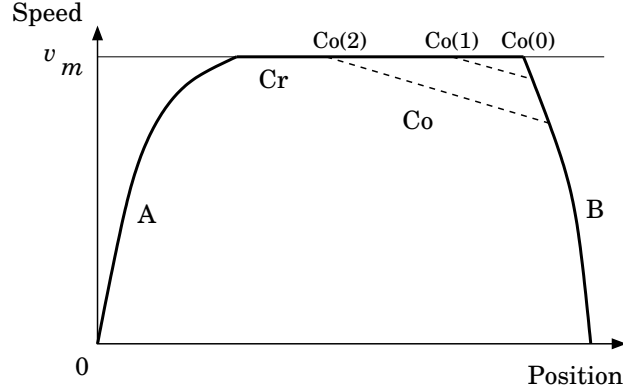


Figure 2: Usual speed profile over one section in four steps (assuming no slope): acceleration (A), cruising (Cr), coasting (Co) and Braking (B).

2.2.1. Tractive effort

Tractive effort F_T is the effort that the train produces for running, bounded by the maximum tractive effort F_m . The maximum tractive effort that a train can produce is a function of both the train characteristics and the current speed.

2.2.2. Constraints

Train's speed, tractive effort and braking are bounded. Let v_m^c be the maximum speed of train c , $F_m(v)$ the maximum tractive effort that the train can exert when traveling at speed v , and b_m the maximum service braking.

$$v \leq v_m^c \quad (2)$$

$$F_T(v) \leq F_m(v) \quad (3)$$

$$b \leq b_m. \quad (4)$$

Given that the deceleration depends also on the gradient of the track, it may happen that using the maximum service braking b_m is not sufficient to slowdown the train if it runs in a steep descent. The respective effect in acceleration may appear if the train runs in a climb so steep that it cannot accelerate though using the maximum tractive effort $F_m(v)$. However, such gradients are unusual in practice, given that the tracks are designed in such a way that the rolling stocks can move without difficulty. The problem may appear with materials not planned to run on some tracks.

2.2.3. Resistances

The resistance to the train advance F_R corresponds to the sum of line (L_R), curve (C_R) and vehicle (M_R) resistances:

$$F_R(v) = L_R + C_R + M_R(v). \quad (5)$$

Line resistance L_R depends on the train mass and the slope angle β :

$$L_R = m g \sin \beta, \quad (6)$$

g being the gravity constant: $g = 9.81$ N/kg. However, line resistance L_R is often approximated as:

$$L_R = m g q, \quad (7)$$

with $q = \tan \beta$. This is considered a good approximation since, for small values of β as the ones considered, $\sin \beta$ and $\tan \beta$ are very similar. The quantity q is a gradient measured in meters per thousand.

Concerning curve resistance, the value of C_R is approximated by

$$C_R = m g \frac{700}{c}. \quad (8)$$

Finally, vehicle resistance M_R combines both rolling resistance and air resistance. The former linearly increases as a function of the adhesion and the wheel rims. The latter quadratically increases as a function of the train velocity. Resistance M_R depends on the physical properties of the train and its current speed. In order to simplify its calculation, we use the Davis' equation [24] proposed in the 1920s and still used today. This formula introduces constants A , B and C , specific to the rolling stock, and allows to approximate the vehicle resistance M_R as follows:

$$M_R(v) = A + Bv + Cv^2. \quad (9)$$

2.2.4. Mechanical energy consumption

Let E be the mechanical energy needed to move the train. It can be calculated as the integral of the mechanical power over the running time T [1]. For convenience, let $F(t)$ and $v(t)$ be the tractive effort and the train's speed at instant t , respectively:

$$E = \int_0^T P(t) dt \text{ with } P(t) = F(t) v(t). \quad (10)$$

Power $P(t)$ generated by the train at instant t is calculated in function of the regime adopted, within function `apply_regime` (Algorithm 1 in Section 4.1).

If we want to consider aspects as electrical-mechanical energy conversion (for instance, motors and inverters), it is possible to use another railway dynamic model including the principles mentioned. In no case, the optimization model will be affected because it uses the objective values T, E . However, changing the dynamic model will impact the objective value E , and consequently the speed profiles generated.

2.3. Description of the driving regimes

As mentioned in Section 2.1, according to the Maximum Principle, four regimes can be adopted by the train, when power recovery (regenerative braking) is not used [8]: acceleration, cruising phase, coasting and braking. Energy consumption and running time evolve differently during each of them.

As mentioned by Miyatake and Ko in [8], there are difficulties in efficient utilization of regenerative braking. In particular, an accelerating train must be close to the braking one so that the former can absorb the regenerative energy. However, only one train is actually considered in our method, whereas the problem of recovering energy by another train is relevant when considering several trains at once. Moreover, such a problem implies the optimization of the synchronization of the trains in the construction of the timetable, which is out of the scope of the paper. Hence, we do not consider the regenerative braking.

2.3.1. Acceleration

During this phase, the train accelerates at full power. The delivered power, P , depends on both tractive effort F_T and speed v . Hence, during an acceleration:

$$F_T(v) = F_m(v) \tag{11}$$

$$P(v) = F_T(v) v. \tag{12}$$

2.3.2. Cruising phase

This regime consists in maintaining the speed constant, i.e., the acceleration is nil: $\gamma = 0$. In fact, the resistance is counterbalanced by the minimum necessary tractive effort: $F_T(v) = F_R(v)$. In other words, the train must adapt its effort to the resistance either by partially braking or producing an effort according to the gradient and the resistances (line and vehicle).

Since γ results from both the acceleration due to the forces of motion and the braking, we denote a the acceleration due to the forces of motion and b the service braking, so that: $\gamma = a - b$.

Let σ be the gradient representing the threshold under which the descent may allow the train to accelerate without effort. We can formalize the effort in the two cases defined below.

1. If $q \geq \sigma$, the train has to produce effort to maintain the speed. First, we determine the resistance $F_R(v)$ and then we set the tractive effort $F_T(v)$ that the train should exert to counterbalance $F_R(v) \geq 0$:

$$F_T(v) = F_R(v) \Leftrightarrow \gamma = 0. \quad (13)$$

Last but not least, the power delivered can be deduced as follows:

$$P(v) = F_T(v) v. \quad (14)$$

2. If $q < \sigma$, the train has to partially brake to maintain the speed. First, we state resistance $F_R(v)$. Given that $F_R(v) < 0$ in this case, we determine the resulting acceleration $a > 0$ to deduce the braking b necessary to keep $\gamma = a - b = 0$.

$$b = a = \frac{-F_R(v)}{\rho m}. \quad (15)$$

Finally, as no power recovery is considered, we set the power delivered as nil:

$$P(v) = 0. \quad (16)$$

2.3.3. Coasting

The coasting corresponds to an inertia motion, while the engine is stopped. The tractive effort is therefore nil:

$$F_T(v) = 0. \quad (17)$$

As a consequence, the energy consumption during coasting is nil and $P(v) = 0$.

2.3.4. Braking

The braking is computed using the maximum service braking b_m . When braking, the tractive effort is therefore nil:

$$F_T(v) = 0. \quad (18)$$

Like in the coasting regime, the energy consumption during coasting is nil and $P(v) = 0$.

Table 2: Symbols used in the problem definition

S	sequence of sections composing the train path
n	number of sections
i	section index: $1 \leq i \leq n$
p_i	starting position of section i with respect to a reference point
l_i	length of section i
$v_{m,i}$	maximum speed in section i (remark that $v_{m,i} > 0$ for all $i = 1, \dots, n$)
$l_{f,i}$	length of the first part of section i
$l_{s,i}$	length of the second part of section i
$t_{f,i}$	duration of the first part of section i
$t_{s,i}$	duration of the second part of section i
$e_{f,i}$	energy spent in the first part of section i
$e_{s,i}$	energy spent in the second part of section i

3. Problem definition

In order to build a speed profile between two stations, we build the speed profile within each section covered by the train, sequentially. Each section is then decomposed according to a set of speeds for choosing the appropriate driving regimes. The main symbols used in the following are defined in Table 2.

3.1. Objectives formulation

The problem under study can be formulated as a set Φ of two objective functions to be minimized while respecting constraints. The first objective function represents the minimization of journey duration T , and the second the reduction of energy consumption E .

$$\Phi = (\min T, \min E). \quad (19)$$

Objective values (T_u, E_u) of solution u are assessed by `eval_solution`, which is a function defined by Algorithm 3 in Section 4.2.

3.2. Speed-based decomposition of a section

3.2.1. Using target-speeds as decision variables

As the train path is decomposed into a set of n sections, the speed profile is successively built in each section. This construction is based on the use of target-speeds, which allow the decomposition of each section into a sequence of driving regimes. For each section $i = 1, \dots, n$ we define the following speeds: $v_{e,i}, v_{x,i}, v_{a,i}, v_{b,i} \in \mathbb{R}$. The speeds $v_{e,i}$ and $v_{x,i}$ are, respectively, the entrance and exit speeds of section i and they are determined while building the speed profile.

The speeds $v_{a,i}, v_{b,i}$ are the decision variables searched for by the algorithm and they are at the basis of the decomposition of the section. They represent two speed-levels that the train must reach while running over the section. The main idea is to allow the introduction of driving regimes following a set of rules which depend on the speeds, as explained in the following.

3.2.2. Principle of decomposition of a section

Speed profiling is done in two phases, each depending on a set of speeds. Figure 3 depicts the decomposition of the speed profile over one single section, as well as the corresponding time-position diagram. The main idea consists in splitting the section into two parts: a first part in which the acceleration at full power (the most energy-consuming driving regime) can be used and a second part for using energy-friendly (cruising) or energy-free (coasting, braking) driving regimes.

During the first part, the train enters at speed $v_{e,i}$ and has to reach the first target-speed $v_{a,i}$ by braking or accelerating. Then, during the second part, the train tries to reach speed $v_{b,i}$, initially by coasting. Additional regimes may be used to reach speed $v_{b,i}$ (braking) and complete the rest of the section (cruising). Even if some of these driving regimes are not used, globally, their use follows the order: coasting, cruising, braking. Building the profile between the entrance speed $v_{e,i}$ and the first target-speed $v_{a,i}$ allows the identification of the length and the time necessary for the first part: $l_{f,i}$ and $t_{f,i}$, respectively. This building also allows the deduction the length of the second part of the section: $l_{s,i} = l_i - l_{f,i}$. The construction of the second part starts from position $p_i + l_{f,i}$ and depends on target-speeds $v_{a,i}$ and $v_{b,i}$. The produced sequence of driving regimes leads to determine the exit speed $v_{x,i}$ of section i and, obviously, $v_{e,i+1} = v_{x,i}, i < n$. The details of the construction are explained right below.

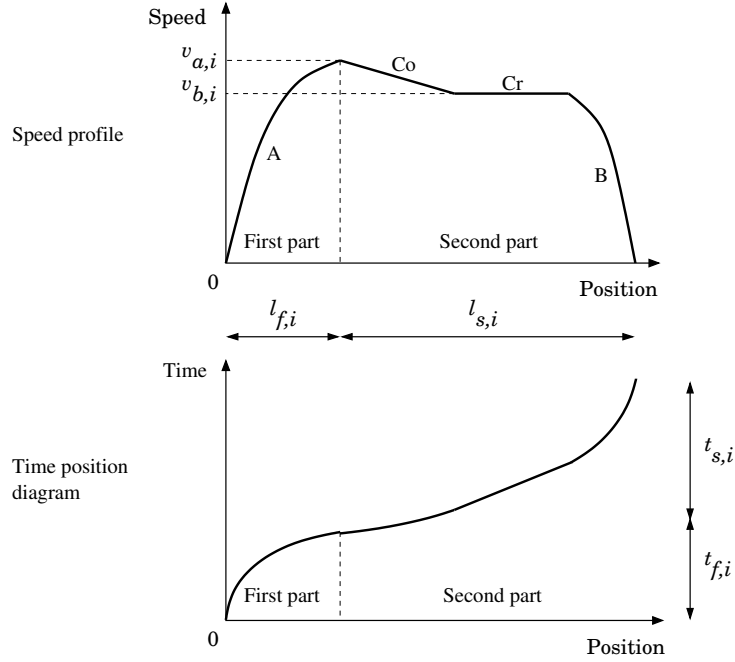


Figure 3: Decomposition of a section into two parts: (a) Speed profile describing the sequence of the driving regimes according to the target-speeds; (b) Time position diagram describing the corresponding train path as well as the required times to cover the section.

Constraints. During the solution construction, we impose Constraints (20) to (22) to the decision variables for each section.

$$v_{a,i} \geq v_{b,i} \quad \forall i = 1, \dots, n \quad (20)$$

$$v_{m,i} \geq v_{a,i} \quad \forall i = 1, \dots, n \quad (21)$$

$$v_{b,i} > 0 \quad \forall i = 1, \dots, n \quad (22)$$

As explained in the following, the value of $v_{b,i}$, $i = 1, \dots, n$, may be changed during the evaluation of the objective function, in case the original one results unfeasible.

4. Solution assessment and running time calculation

In this section, the algorithms for building the speed profile and for assessing a solution are given and detailed. But, beforehand, we provide algorithms

for calculating train dynamics corresponding to a driving regime. These algorithms are essential to compute distance covered, energy consumed and time spent during a driving regime. After this description, we will give the algorithms of speed profiling as a function of the target-speeds defined in each section. In the following, symbols `ac`, `cr`, `co`, `br`, respectively, represent acceleration, cruising, coasting and braking.

4.1. Calculation of driving regime

Based on the description of the possible driving regimes, Algorithm 1 defines function `apply_regime` which calculates time spent, length, energy, and speed at each position in function of the characteristics of the track and the train, and also of the speeds given in input. The principle at the basis of this iterative function is to determine efforts, resistances, acceleration, speed, power and energy at each instant t (let Δt be the time-slot).

Since function `apply_regime` needs to be interrupted when changing the driving regime, function `end_reached` (Algorithm 2) indicates when the current regime is implemented. The main reasons to interrupt a regime are either that the target-speed is reached or that the limit position beyond which the regime used must be changed is attained.

4.2. Objectives computation

For computing T and E , we apply the function `eval_solution` described in Algorithm 3. Within this function, T and E are calculated for each section consecutively by the function `eval_section` defined in Section 3.2.

Algorithm 4 describes the function named `eval_section`. Based on the characteristics of a section and the values of the decision variables, this function returns the time and the energy spent in the section itself. Within this function, we use two additional sub-functions `first_part` and `second_part`, which respectively build the speed profile on the first and the second part of the section under consideration (Algorithms 5 and 6 respectively).

Construction of the speed profile in the first part. The first part corresponds to the entrance in the section and depends on two speeds: the entrance speed $v_{e,i}$ and the target-speed $v_{a,i}$. The latter is a decision variable of the problem and it is searched for by the evolutionary algorithm. The construction of the speed profile is carried out through Algorithm 5, which identifies the regime to be used: acceleration if $v_{a,i} > v_{e,i}$, braking otherwise. If the two speeds are equal ($v_{e,i} = v_{a,i}$), length, time and energy spent are nil: $l_{f,i} = 0, t_{f,i} = 0, e_{f,i} = 0$; function `end_reached()` returns 1 in such a case.

Algorithm 1: Function `apply_regime`(v_1, v_2, l, p, r)

Data: v_1 : initial speed, v_2 : speed to reach, l : distance to cover, p : start position, r : driving regime to use

Result: (t, l, e, R) : a vector containing the time spent, the length and the energy used during the motion. R is a vector containing the pairs (p_t, v_t) .

Initialization
 $t = h; l, e = 0; R = ()$
 $v_t = v_1; p_t = p$

begin

```
while not end_reached( $v_t, v_1, v_2, p, p + l, r$ ) do
    Calculating  $L_R, C_R, M_R$  as function of  $v_t, p_t$  (Eq. 7, 8, 9)
     $F_R = L_R + C_R + M_R$ 

    if  $r == \{ br \text{ or } co \}$  then
         $F_T = 0$ 
        if  $r == br$  then
             $b = b_m$ 
        else
             $b = 0$ 
    else
        if  $r == cr$  then
             $F_T = \max(0, \min(F_m, F_R))$ 
             $b = \max(0, -F_R / \rho \cdot m)$ 
        else
            --  $r == ac$ 
             $F_T = F_m$ 
             $b = 0$ 

         $a = \frac{F_T - F_R}{\rho \cdot m} + b$ 
         $v_t = v_t + a \Delta t$ 

         $p_t = p_t + v_t \Delta t$ 
         $l = l + v_t \Delta t$ 

         $e = e + F_T v_t \Delta t$ 

         $R = R \cup (p_t, v_t)$ 
         $t = t + \Delta t$ 
end
```

Construction of the speed profile in the second part. This part depends on both variables defined for section i , namely $v_{a,i}$ and $v_{b,i}$ and it depends on the gradient, the maximum speed of the following section and the capability to coast all over length $l_{s,i}$. Let l_{co} be the length of coasting, l_{cr} the length of cruising, l_{br} the length of braking. Algorithm 6 describes the construction of the second part. It has to be noted that two additional functions are used in the algorithm. The first is `search_for_intersection` which computes the changing between two regimes by searching for the intersection of the speed curves representing the driving regimes under consideration. The second is `apply_reverse_regime` which is the counterpart of `apply_regime` but it

Algorithm 2: Function `end_reached(v_1, v_2, p_1, p_2, r)`

Data: v_1 : current speed, v_2 : target-speed, p_1 : current position, p_2 : limit position, r : current driving regime

Result: $reached = \{0|1\}$

```
begin
  reached = 1;
  switch r do
    case br
      | if  $v_1 > v_2$  or  $p_1 < p_2$  then reached = 0
    case co
      | if  $v_1 < v_2$  or  $p_1 < p_2$  then reached = 0
    case cr
      | if  $p_1 < p_2$  then reached = 0
    case ac
      | if  $v_1 < v_2$  or  $p_1 < p_2$  then reached = 0
  end
```

Algorithm 3: `eval_solution($u = (v_{a,1}, v_{b,1}, \dots, v_{a,n}, v_{b,n})$)`

Data: for each section $i = 1, \dots, n$: $v_{a,i}, v_{b,i}, p_i, l_i, v_{m,i}$

Result: vector (T, E) including the total running time and total energy consumption

$(T, E) = (0, 0)$;

$(T, E) = (T, E) + \text{eval_section}(0, v_{a,1}, v_{b,1}, v_{a,2}, p_1, l_1, v_{m,1}, v_{m,2})$;

for $i = 2, \dots, n - 1$ **do**

$(T, E) = (T, E) + \text{eval_section}(\min\{v_{b,i-1}, v_{m,i}\}, v_{a,i}, v_{b,i}, v_{a,i+1}, p_i, l_i, v_{m,i}, v_{m,i+1})$;

$(T, E) = (T, E) + \text{eval_section}(\min\{v_{b,n-1}, v_{m,n}\}, v_{a,n}, v_{b,n}, 0, p_n, l_n, v_{m,n}, 0)$

Algorithm 4: `eval_section($v_e, v_a, v_b, p, l, v_m, v_n$)`

Data: v_e : entry speed, v_a : target speed in the first part, v_b : target speed in the second part, p : entry position, l : section length, v_m : maximum speed of the section, v_n : maximum speed of the next section

Result: vector (t, e) including the total running time and total energy consumption in the section

```
1 begin
2    $(t_a, l_a, e_a, R_a) = \text{first\_part}(v_e, v_a, l, p)$  ;
3    $(t_{co}, t_{cr}, t_{br}, e_{co}, e_{cr}, e_{br}, l_{co}, l_{cr}, l_{br}, R_{co}, R_{cr}, R_{br}) = \text{second\_part}(v_a, v_b, p, l_a, v_m, v_n)$  ;
4    $t = t_a + t_{co} + t_{br} + t_{cr}$  ;
5    $e = e_a + e_{co} + e_{br} + e_{cr}$  ;
6 end
```

computes the phase from the end-point to the beginning. This function is used when no beginning-point is known for the driving regime to use, but the end-point of the phase is known. Given that these two functions can be retrieved easily, they are not defined in this paper.

If $v_{a,i} = v_{b,i}$, the speed profile consists of a cruising phase at speed $v_{a,i}$ all along the section if $v_{m,i+1} \geq v_{a,i}$ (Fig. 4(a)), i.e., if the maximum speed

Algorithm 5: `first_part(v_e, v_a, l, p)`

Data: v_e : entry speed, v_a : target speed in the first part, p : start position, l : section length, v_m : maximum speed of the section

Result: vector (t_a, l_a, e_a, R_a) including the total running time, length, energy consumption and regime used in the section.

```
1 begin
2   if  $v_e < v_a$  then
3      $(t_a, l_a, e_a, R_a) = \text{apply\_regime}(v_e, v_a, l, p, \mathbf{ac})$  ;
4   else
5      $(t_a, l_a, e_a, R_a) = \text{apply\_regime}(v_e, v_a, l, p, \mathbf{br})$  ;
6 end
```

of the following section is higher than the current target-speed. Otherwise it consists in a cruising phase at speed $v_{a,i}$ followed by a braking to reach speed $v_{m,i+1}$ (Fig. 4(b)).

Let v_l be the last speed measured at the end of the coasting and returned by function `exit_coast` (not defined in the paper). If $v_{a,i} > v_{b,i}$, we try to insert a coasting phase:

- If $q \geq \sigma$ the train decelerates by coasting, and thus $v_l < v_{a,i}$ because of the slowdown due to the resistive efforts while coasting,
- If $q < \sigma$: there is a steep descent.

If the coasting permits to reach $v_{b,i}$, then $v_l = v_{b,i}$. Otherwise, a number of cases must be distinguished to be treated differently. When $v_l < v_{a,i}$, we distinguish four cases depending on the possibility to coast along a distance smaller than or equal to the distance $l_{s,i}$:

- if $v_{b,i} \leq v_{m,i+1}$
 - i. if the train may reach speed $v_{b,i}$, starting at speed $v_{a,i}$, by coasting along the length $l_{s,i}$, the speed profile includes the coasting followed by a cruising regime at speed $v_{b,i}$ in the remaining distance (Fig. 4(c)). The exit speed $v_{x,i}$ equals $v_{b,i}$,
 - ii. if the train covers the section by coasting and it never reaches speed $v_{b,i}$, then we set: $v_{b,i} = v_l$ (Fig. 4(d)). In addition, $v_{x,i} = v_{b,i}$.
- if $v_{b,i} > v_{m,i+1}$
 - iii. if the train may reach speed $v_{b,i}$, starting at speed $v_{a,i}$, by coasting along the length $l_{s,i}$, the same speed profile described in (i) is

imposed, but a final braking is necessary to enter the following section at speed $v_{m,i+1}$ (Fig. 4(e)). In this case, $v_{x,i} = v_{m,i+1}$,

- iv. if the train covers the second part of the section by coasting and it never reaches speed $v_{b,i}$, then a final braking is imposed for attaining this speed (Fig. 4(f)). Let v_c be the speed measured when starting braking, i.e., v_c is obtained after calling function `search_for_intersection`. Since speed $v_{b,i}$ cannot be attained, it is then corrected by replacing its value with: $v_{b,i} = v_c$. Last, $v_{x,i} = v_{m,i+1}$.

As discussed above, it may happen that a coasting results in an acceleration if $q < \sigma$, in this case:

- if $v_{b,i} \leq v_{m,i+1}$, the coasting is interrupted by a braking to leave the section at speed $v_{b,i}$ (Fig. 5(a)),
- if $v_{b,i} > v_{m,i+1}$, the train stops coasting and brakes to leave the section at speed $v_{m,i+1}$. Speed $v_{b,i}$ is therefore corrected to $v_{m,i+1}$: $v_{b,i} = v_{m,i+1}$ (Fig. 5(b)).

4.3. Post processing: smoothing the speed profiles

Although the construction of speed profiles aims to avoid sequences composed of braking followed by acceleration, a post-processing is necessary for guaranteeing that it is always the case. In fact, if the slope in the second part of the section is sufficiently steep to make the train accelerate while coasting, then a braking is introduced to reach speed $v_{b,i}$. If $v_{b,i} < v_{m,i+1}$, this braking could be followed by an acceleration (if $v_{a,i+1} > v_{b,i}$).

For avoiding this, we use a smoothing post-processing to eliminate two types of sequences: (Braking, Acceleration at full power); (Braking, Acceleration while coasting). Whatever the sequence under consideration, we can distinguish two cases for which we determine a cruising phase replacing one part of the sequence depending on speeds v_c (defined as speed measured when starting braking) and $v_{a,i+1}$ (Fig. 6(a, c)). The cruising speed corresponds to the minimum between them: $\min(v_c, v_{a,i+1})$. Finally, Figures 6(c, d) are respectively the smoothed profiles of Figures 6(b, d).

All along this inserted cruising phase, it is necessary to compute the effort necessary to maintain the speed constant. This effort will replace the one previously computed for the acceleration phase in the evaluation of

Algorithm 6: $\text{second_part}(v_a, v_b, p, l, v_m, v_n)$

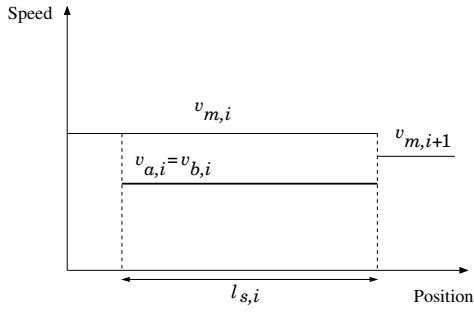
Data: v_a : target speed in the first part, v_b : target speed in the second part, p : start position, l : section length, v_m : maximum speed of the section, v_n : maximum speed of the next section

Result: vector $(t_{co}, t_{cr}, t_{br}, e_{co}, e_{cr}, e_{br}, l_{co}, l_{cr}, l_{br}, R_{co}, R_{cr}, R_{br})$ including the total running time, the total energy consumption and the total run length of each regime used in the second part of the section.

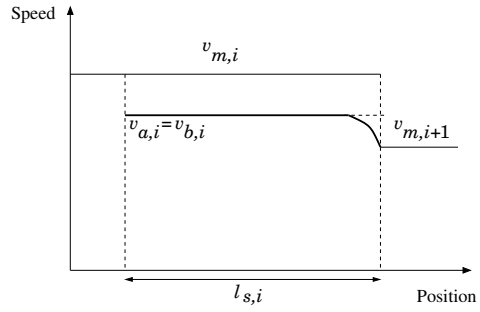
```
1 begin
2   if  $v_a == v_b$  then
3     if  $v_a < v_n$  then
4        $(t_{cr}, l_{cr}, e_{cr}, R_{cr}) = \text{apply\_regime}(v_a, v_b, l - l_a, p + l_a, \mathbf{cr})$ ;
5     else
6        $(t_{br}, l_{br}, e_{br}, R_{br}) = \text{apply\_reverse\_regime}(v_b, v_n, l - l_a, p + l, \mathbf{br})$ ;
7        $(t_{cr}, l_{cr}, e_{cr}, R_{cr}) = \text{apply\_regime}(v_a, v_b, l - l_a - l_{br}, p + l_a, \mathbf{cr})$ ;
8   else
9     //  $v_a > v_b$ 
10     $v_l = \text{exit\_coast}(v_a, v_b, l - l_a, p + l_a)$ ;
11     $(t_{co}, l_{co}, e_{co}, R_{co}) = \text{apply\_regime}(v_a, v_b, l - l_a, p + l_a, \mathbf{co})$ ;
12    if  $v_l > v_a$  then
13      if  $v_n \leq v_b$  then
14         $(t_{br}, l_{br}, e_{br}, R_{br}) = \text{apply\_reverse\_regime}(v_m, v_n, l - l_a, p + l, \mathbf{br})$ ;
15         $(t_{co}, t_{br}, l_{br}, l_{co}, e_{co}, e_{br}, R_{br}, R_{co}) = \text{search\_for\_intersection}(R_{co}, R_{br})$ ;
16         $v_b = v_n$ ;
17      else
18         $(t_{br}, l_{br}, e_{br}, R_{br}) = \text{apply\_reverse\_regime}(v_m, v_b, l - l_a, p + l, \mathbf{br})$ ;
19         $(t_{co}, t_{br}, l_{co}, l_{br}, e_{co}, e_{br}, R_{co}, R_{br}) = \text{search\_for\_intersection}(R_{co}, R_{br})$ ;
20    else
21      if  $v_l > v_b$  then
22        if  $v_n \leq v_l$  then
23           $(t_{br}, l_{br}, e_{br}, R_{br}) = \text{apply\_reverse\_regime}(v_m, v_n, l - l_a, p + l, \mathbf{br})$ ;
24           $(R_{co}, R_{br}, v_c) = \text{search\_for\_intersection}(R_{co}, R_{br})$ ;
25           $v_b = v_c$ ;
26        else
27           $v_b = v_l$ ;
28      else
29         $v_t = \min(v_b, v_n)$ ;
30         $(t_{br}, l_{br}, e_{br}, R_{br}) = \text{apply\_reverse\_regime}(v_b, v_t, l - l_a, p + l, \mathbf{br})$ ;
31         $(t_{cr}, l_{cr}, e_{cr}, R_{cr}) = \text{apply\_regime}(v_b, v_b, l - l_a - l_{co} - l_{br}, p + l, \mathbf{cr})$ ;
32     $t = t_a + t_{co} + t_{br} + t_{cr}$ ;
33     $e = e_a + e_{co} + e_{br} + e_{cr}$ ;
34 end
```

the second objective of the optimization. The same holds for the running time associated to the speed profile. The advantage of inserting a cruising phase instead of an inappropriate sequence is to reduce the journey duration while also reducing the quantity of energy consumed, because acceleration is replaced by a regime far less energy-consuming.

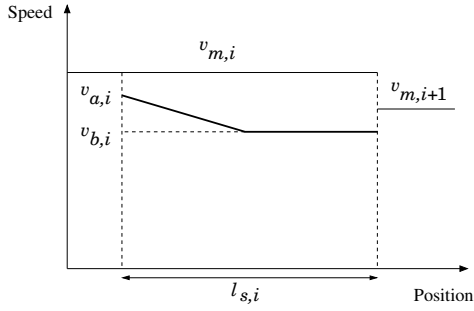
It has to be noted that the post processing is applied to any solution as



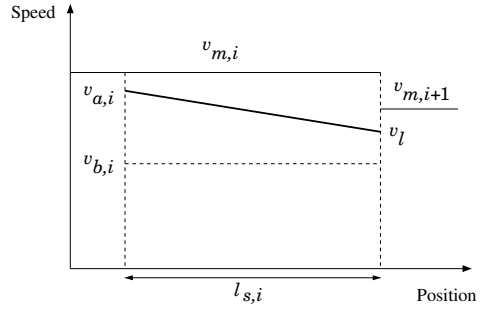
(a) Algorithm 6 line 4: $v_{a,i} = v_{b,i}$ and $v_{a,i} \leq v_{m,i+1}$



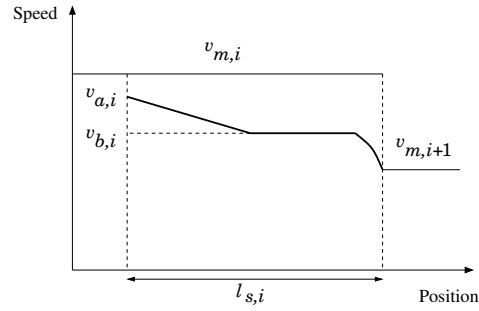
(b) Algorithm 6 line 6: $v_{a,i} = v_{b,i}$ and $v_{a,i} > v_{m,i+1}$



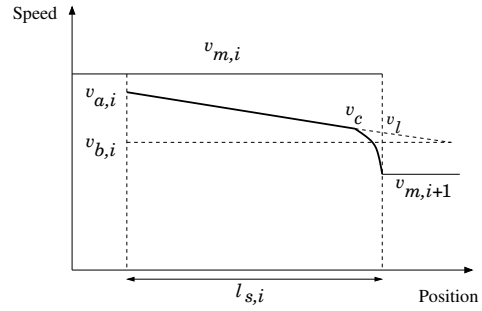
(c) Algorithm 6 line 29: $v_{b,i} \leq v_{m,i+1}$



(d) Algorithm 6 line 27: $v_l \leq v_{m,i+1}$



(e) Algorithm 6 line 29: $v_{b,i} > v_{m,i+1}$



(f) Algorithm 6 line 23: $v_l > v_{m,i+1}$

Figure 4: Description of the possible situations in the second part of a section

soon as it is generated.

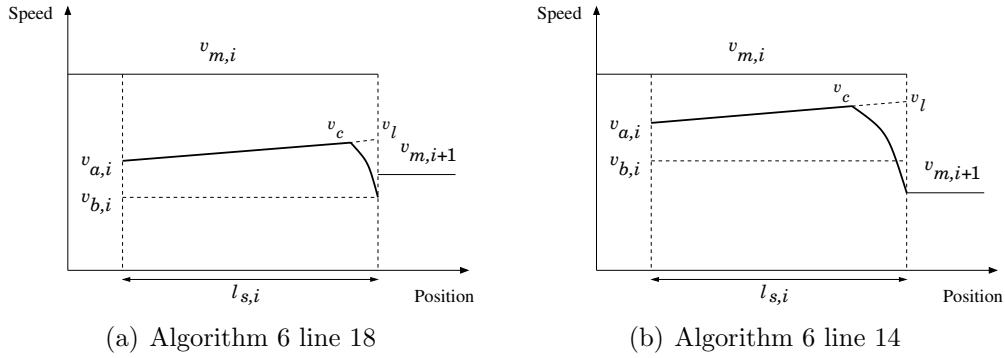


Figure 5: Description of particular situations in the second part of a section, when gradient q is negative and the descent is such that a train can accelerate without effort.

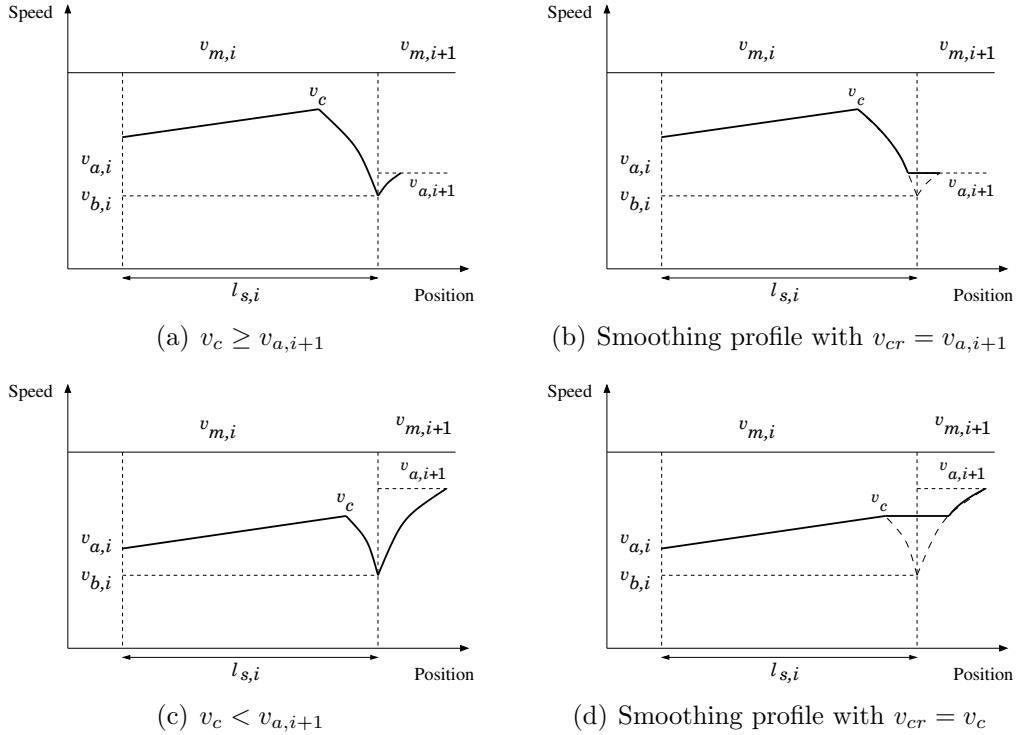


Figure 6: Description of smoothing of speed profiles: profiles (a) and (b) have a braking followed by an acceleration (at full power or by coasting in descent); profiles (c) and (d) are the respective smoothed speed profiles.

5. Evolutionary Multi-objective Optimization

The problem under study is a multi-objective continuous optimization problem. To tackle it, we propose to use an evolutionary algorithm because

this kind of algorithm is known to be well-suited to multi-objective problems [17]. First, we present multi-objective optimization principles and concepts. Then, we introduce the state-of-the-art evolutionary algorithm which is used in the experimental analysis. Finally, we present the mechanisms specific to our application.

5.1. Multi-objective Optimization

A general Multi-objective Optimization Problem (MOP) can be defined by a set of k objective functions $f = (f_1, f_2, \dots, f_k)$ and a set U of feasible solutions in the decision space. Let Z be the objective space $Z = f(U)$. Without loss of generality, we assume here that each objective function is to be minimized. To each solution $u \in U$ is assigned an objective vector $z \in Z$ with $z = \{z_1, z_2, \dots, z_k\}$ computed on the basis of the vector function $f : U \rightarrow Z$ with $z = f(u) = (f_1(u), f_2(u), \dots, f_k(u))$. An objective vector $z \in Z$ is said to *dominate* another objective vector $z' \in Z$ iff $\forall i \in \{1, 2, \dots, k\}, z_i \leq z'_i$ and $\exists j \in \{1, 2, \dots, k\}$ such that $z_j < z'_j$. A decision vector $u \in U$ *dominates* a decision vector $u' \in U$ if $f(u)$ dominates $f(u')$. An objective vector $z \in Z$ is said to be *non-dominated* iff no other objective vector $z' \in Z$ exists such that z' dominates z . A solution $u \in U$ is said to be *efficient*, or *Pareto optimal*, if its mapping in the objective space results in a non-dominated point.

Due to the complexity of the underlying problem, the overall goal is often to identify a good approximation of the efficient set. Population-based metaheuristics in general, and evolutionary algorithms in particular, are commonly used to this end, as they are capable of finding multiple and well-spread non-dominated solutions in a single run [17].

5.2. Indicator Based Evolutionary Multi-objective Algorithm

Over the last decades, a very large number of evolutionary algorithms for MOP solving have been proposed in the literature [17, 25]. These approaches can be seen as frameworks in which problem-related components have to be defined. In this work, we have used a state-of-the-art evolutionary multi-objective optimization algorithm, namely the Indicator-Based Evolutionary Algorithm (IBEA) [18]. This algorithm follows the main steps illustrated in the flowchart of Fig. 7.

IBEA characterizes the trend in evolutionary computation dealing with indicator-based search which has become popular over recent years. The main idea is to introduce a total order between solutions by means of a binary quality indicator. In multi-objective optimization, ‘quality’ represents

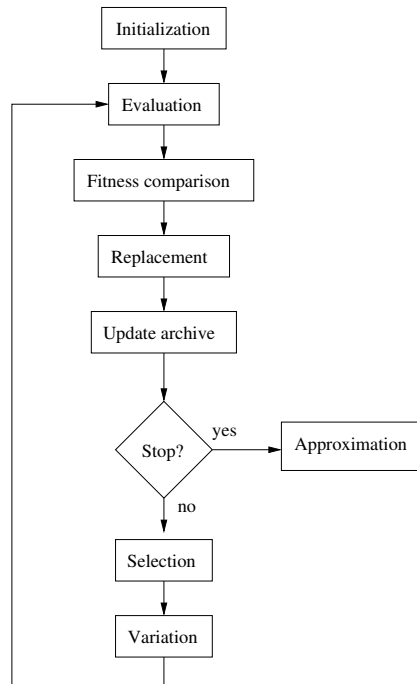


Figure 7: Flowchart of an evolutionary multi-objective optimization algorithm.

the well-spread aspect of the solutions in a front. Each solution contributes to the spread of the front, and hence to its quality [18].

The fitness assignment scheme is based on a pairwise comparison of solutions from the current population with regard to the indicator $I_{\epsilon+}$ [26]. To each individual u is assigned a fitness value $\phi(u)$ which is to be maximized, and measuring the contribution of the solution u and hence the ‘loss in quality’ if u is removed from the current population Q , *i.e.*, $\phi(u) = \sum_{u' \in Q \setminus \{u\}} (-e^{-I(u',u)/\kappa})$, where $\kappa > 0$ is a user-defined scaling factor.

The variation step comprises recombination (crossover) and mutation. The selection for crossover consists of a binary tournament between randomly chosen individuals and the selection for replacement consists in iteratively removing the worst solution from the current population until the required population size is reached. The fitness information of the remaining individuals is updated whenever one is deleted. Moreover, all new non-dominated solutions found during the process are archived in a separate population. This archived population is updated every iteration in function of the new

non-dominated solutions for discarding the dominated ones.

5.3. Solution representation and initialization

A solution is defined by a vector of speeds: $\langle v_{a,1} v_{b,1} \dots v_{a,n} v_{b,n} \rangle$. Given that two speeds are necessary to represent a section, the number of components of solution u equals twice the number n of sections: $\#u = 2n$.

To avoid too many unfeasible solutions at the beginning of the optimization, a specific initialization strategy is developed based on the fastest journey, as described in Section 5.3.2.

5.3.1. Determination of the reference solution

In order to have a reference solution for further comparisons, we search for the solution which minimizes the running time, denoted u^* . Concretely, it consists in driving as fast as possible with respect to the speed constraints of the track.

A complete description of this calculation is given in [20]. In few words, the speed profile is built in three steps. First, the method consists in determining all necessary braking at the end of the sections for respecting the maximal speed of consecutive sections. Second, it consists in determining the maximal acceleration at the beginning of each section. Third, cruising phases are added between accelerations and brakings to complete the speed profile. The obtained solution represents the lower-bound \underline{T} of running time and serves also as basis of comparison for the energy consumption. The decision-maker will be able to limit the possible range of running time by upper-bounding it to a duration equal to $x \times \underline{T}$, by setting parameter $x > 1$.

5.3.2. Initialization of the population

The solutions are based on solution u^* and are successively initialized. The initial population as well as the following ones are composed of a fixed number N of solutions. Within each initialization of solution $\mu \in [1, N]$, the values of $v_{a,i}^\mu, v_{b,i}^\mu$ are determined from $v_{a,i}^{u^*}, v_{b,i}^{u^*}$ in such a way that the solution initialized is automatically longer and less energy-consuming than the reference solution u^* . At every solution initialization, the solution is longer than the previous one. In our implementation, we have chosen a very easy way to do this. First, we assume that the population is limited to 50 solutions: $N = 50$. Then, the interval I_i for decreasing speeds $v_{a,i}^\mu, v_{b,i}^\mu$ per

section i corresponds to half the maximal speed reachable:

$$\forall i \leq n, \quad I_i = \frac{v_{a,i}^{u^*}}{2} \quad (23)$$

$$\forall \mu \in [1, N], \forall i \leq n, \quad v_{a,i}^\mu = v_{a,i}^{u^*} - (I_i \times i \times 0.01) \quad (24)$$

$$(25)$$

This strategy is clearly improvable. Nevertheless, it is satisfactory for producing original solutions and the development of a new initialization strategy is out of the scope of the paper. After initialization, the population is composed of a set of N distinct solutions, which will be used to produce new and improved ones.

5.4. Solution evaluation

Each solution u is evaluated during its construction. In particular, for each section i , the running time t_i and the energy consumption e_i are computed as explained in Section 3.

If a speed profile does not satisfy the constraints or cannot be built, then the solution is not feasible and it is discarded. As the objectives have to be minimized, their fitnesses are assigned huge values so that the solution will not appear in the next population.

5.5. Solution variation

5.5.1. Crossover

Crossover is the mechanism allowing solutions to recombine with each other in order to produce new solutions. As the solution space is continuous, we use an operator adapted to the continuous search: the Simulated Binary Crossover (SBX) [27]. It generates two new solutions from two initial ones belonging to the current population. Its mechanism is inspired from the single-point crossover used in the binary-coded genetic algorithm [27]. A crossover rate indicates the percentage of individuals to recombine.

By assuming that there is one binary variable in the problem to solve, the binary one point crossover states that the average value of the parents is the same as that of the offspring. In the example of Figure 8, p_1, p_2 are the parents used to generate children c_1, c_2 . After recombination, we can see that $\bar{p} = \bar{c}$ and the values c_1, c_2 are spread around the average value \bar{c} . The spread factor β is defined as the ratio of the spread of the child points to

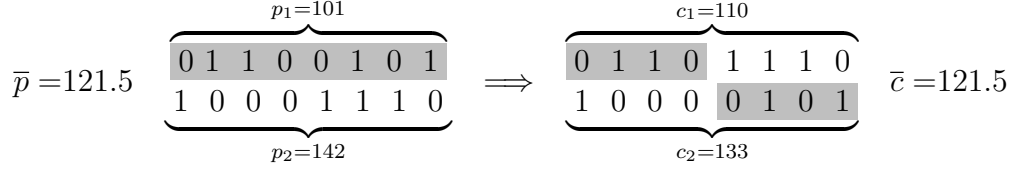


Figure 8: Illustration of the binary one point crossover

that of the parent points: $\beta = \left| \frac{c_1 - c_2}{p_1 - p_2} \right|$. Finally, the values of the offspring can be expressed as follows:

$$c_1 = \bar{p} - \frac{\beta \times |p_1 - p_2|}{2} \text{ and } c_2 = \bar{p} + \frac{\beta \times |p_1 - p_2|}{2} \quad (26)$$

The SBX uses these properties: the average value and the spread factor properties. As the SBX deals with real variables, the spread factor β is defined in \mathbb{R}^+ . Moreover, the SBX is defined for scalar values, whereas the crossover has to be performed on real vectors. To do that, the SBX is uniformly applied over each scalar values of the decision vectors, with regard to a probability, as illustrated in Figure 9. The spread factor β is randomly defined according to a probability density function defined as follows:

$$P(\beta) = \begin{cases} 0.5 \times (\eta_c + 1) \times \beta^{\eta_c} & \beta > 1 \\ 0.5 \times (\eta_c + 1) \times \frac{1}{\beta^{\eta_c+2}} & \beta \leq 1 \end{cases} \quad (27)$$

where the perturbation value η_c is a given positive parameter. Further details and explanation can be found in [27].

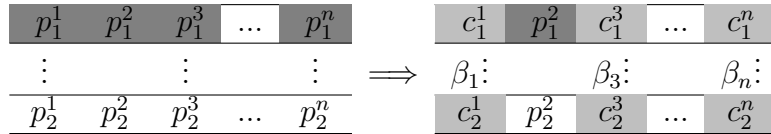


Figure 9: Illustration of the Simulated Binary Crossover (SBX).

5.5.2. Mutation

Mutation consists in providing diversity for the population by modifying a solution randomly chosen. For the same reason mentioned for the crossover,

we use an operator adapted to the continuous search: a polynomial mutation [17, 27]. A mutation rate indicates the percentage of individuals to mutate.

The polynomial mutation is uniformly applied over each scalar values of the decision vector, with regard to a probability: $1/2n$. Let δ be the deviation from the original value. Each scalar value is deviated using the following probability distribution:

$$P(\delta) = \frac{(\eta_m + 1)(1 - |\delta|)^{\eta_m}}{2}, \quad \delta \in [-1, 1] \quad (28)$$

where η_m is a given positive parameter.

6. Experimental analysis

6.1. Implementation

We implemented the algorithm by using the ParadisEO framework [28]. The ParadisEO framework is a ‘white box’ in which several algorithmic components are already implemented, and must be combined and integrated by the user. In addition to the problem-related modules that we have developed, we use the ParadisEO implementations of SBX operator and polynomial mutation. We performed the experiments on a PC (3.0 GHz with 6 GB) running Linux release of ParadisEO framework.

Parameter settings. The population is composed of 50 solutions and evolves over 60 seconds of computation, which is the stopping criterion. Crossover and mutation rates are respectively set to 0.9 and 0.5. Specific parameter κ for IBEA is set to 0.0001. We selected these values based on some preliminary experiments.

6.2. Instances and rolling stock used in the analysis

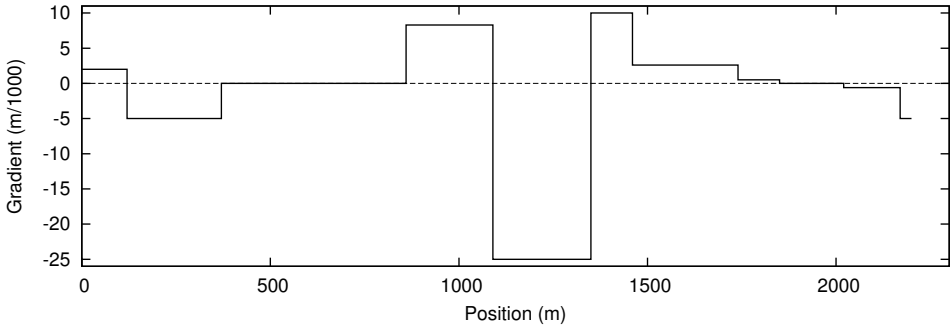
The lines that we use for the experiments correspond to two lines described in Figure 10(a) which reports the length and the maximal speed of each section. The track slopes are depicted in Figure 10(b, c) and given as gradient [%].

Instance 1 corresponds to a line described and used in [7, 8]. It is 2.2 km long and includes five sections. Instance 2 is the *Saint-Étienne–Rive de Giers* line in France. It is 20.2 km long and includes five sections. It is interesting to note that the gradient is mostly negative and thus the outward journey is in descent for the most part of the line.

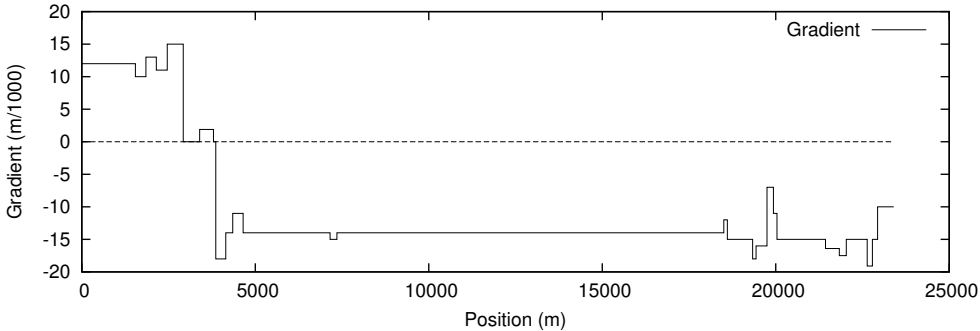
Line 1					
section	1	2	3	4	5
length (m)	1350	160	250	240	200
maximum speed (km/h)	95	70	40	25	40

Line 2					
section	1	2	3	4	5
length (m)	3500	3900	3900	6600	2300
maximum speed (km/h)	90	110	105	120	105

(a) Length and maximum speed allowed on each section



(b) Description of the line 1 gradient



(c) Description of the line 2 gradient

Figure 10: Description of the studied lines

In both lines, the train is an AGC¹. For computing energy consumption, the relevant parameters about AGC, as well as the tractive effort curve, are reported in Fig. 11.

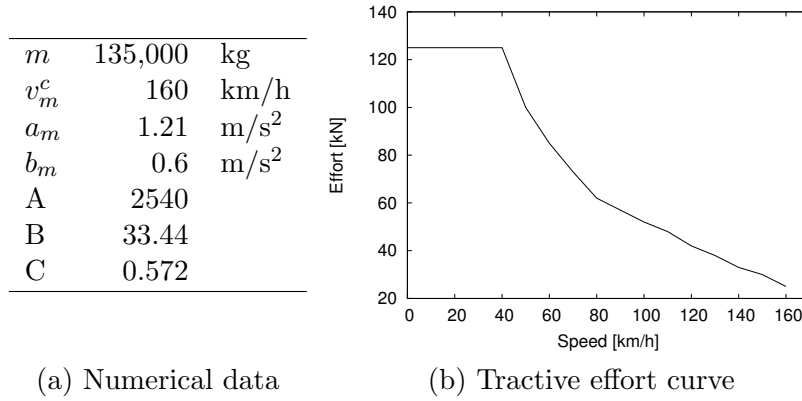


Figure 11: Technical parameters of train AGC

6.3. Results

Here, we analyze the results obtained on the two lines tackled in this paper, separately detailed and discussed in the cases studied hereafter. For each line, we consider two instances representing outward and inward journeys, respectively.

6.3.1. Line 1

Outward journey. Figure 12(a) depicts the objective space of Instance 1a and the sets of produced solutions at different times. Three populations are represented: the initial one (time $t=0$ second), the sets of non-dominated solutions produced at time $t=30$ seconds and the corresponding set at the end of the process (time $t=60$ seconds). The reference solution $I1a^*$ corresponds to the fastest journey and is represented to have a basis of comparison. Note that the limit of journey duration is fixed to $1.5 \times T_{I1a^*}$.

Compared to the initial solutions, it clearly appears that the sets of solutions improved during the process. Moreover, according to the stretched

¹*Autorail Grande Capacité*

shape of the sets, we can say that the solutions have been well diversified during the search. Now, if we compare the set at time $t=30$ with that at time $t=60$, we can see that the solutions have not been strongly improved during the last 30 seconds. Hence, with the adopted parameter settings, the largest part of the optimization has been performed during the first half of the available time.

In Figure 12(b), three speed profiles are drawn: reference solution I1*, and two others: I1.1 and I1.2. Solutions I1.1 and I1.2 are two alternative solutions obtained in the same run. Duration and energy consumption of each solution, as well as deviations from solution I1*, are reported in Table 3. As could be assumed, the decrease percentage of energy consumption is much higher than the increase percentage of running time.

Table 3: Numerical results of three solutions obtained on instance 1a: reference solution I1a* and two alternative solutions I1a.1, I1a.2.

Solution	Duration [s]	Dev. [%]	Energy [kJ]	Dev. [%]
I1*	175		4.9790×10^6	
I1.1	189	+8.0	3.6917×10^6	-25.85
I1.2	213	+21.7	2.6327×10^6	-47.12

By analyzing the speed profile of solution I1a.1 (Figure 12(b)), we can observe that, in the first section, after an acceleration at full power for attaining $v_{a,1} = 21$ m/s, the train coasts for reaching the target-speed $v_{b,1}$. Then, in the second part, we can see that the train accelerates between positions 1100 m and 1300 m due to the steep descent of the track, even if $v_{b,1} = 18.48$ m/s: the train cannot reach the target-speed and the situation identified here corresponds to that described in Algorithm 6 line 18 (Fig. 5(a)). In the second section (from 1350 m to 1510 m), a braking covers the whole available distance. It corresponds in fact to a braking when the train enters in the section, followed by the same situation as before (Fig. 4(e)), but coasting and cruising distances are equal to 0. In the third section, the train enters at speed $v_{e,3} = v_{a,3} = 11$ m/s so that the length of the first part is nil. In the second part, the train coasts and then brakes to leave at the maximum speed of the next section. With speeds $v_{a,3} = 11$ m/s and $v_{b,3} = 9.76$ m/s, that corresponds to the situation described in Algorithm 6 line 27 (Fig. 4(e)) with a cruising distance equal to 0. Finally, the train leaves section 3 at speed

$v_{x,3} = 6.94$ m/s and enters section 4 at speed $v_{e,4} = v_{x,3}$.

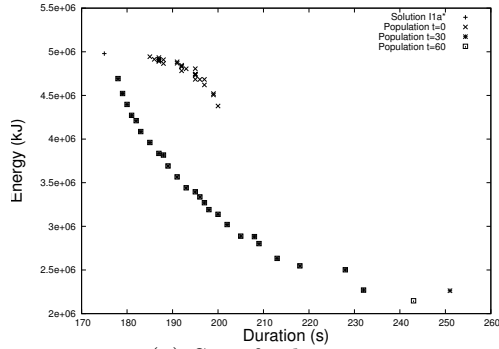
In section 4, the train drives at constant speed and this situation corresponds to the one described in Figure 4(a). It is interesting to note that the train runs at speeds lower than 6.94 m/s. In practice, it is obvious that the train will run at the maximum speed of the section if the limitation is very low (25 km/h in the paper mentioned). However the notion of low speed is not taken into account in our method. Whatever the speed limitation in the section under consideration, it is inherent to the optimization model to find alternative speed profiles even in low-speed sections. Thus, even if the speed limitation is low, the method can propose less energy-consuming speed profiles taking more time than the solution running at the speed limitation. It is therefore normal that other speed profiles are proposed by the method. In order to avoid them, the notion of low speed should be defined for not running at lower speeds. But, the question of what a low speed is, has to be answered first.

Last, in section 5, the train accelerates to reach the first target-speed before coasting until the compulsory braking when arriving at the end of the path (see Fig. 5(b)).

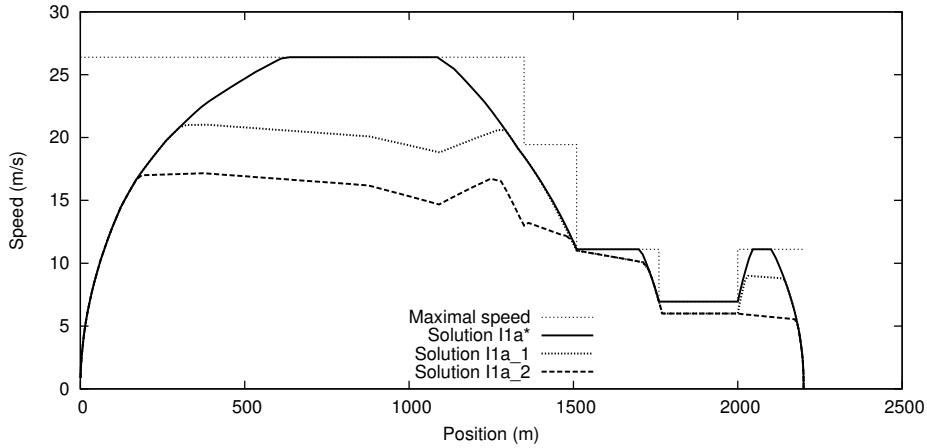
The speed profile of solution I1.2 follows approximately the same driving regimes but with lower target-speeds. Its consumption is weaker compared to solution I1.1 but with, of course, a longer journey duration.

Inward journey. The sets of solutions obtained during the optimization are represented in Figure 13(a). Even if there is a little improvement of the solutions at time 60 seconds, we can say however that the best solutions have been been often by time 30 seconds. Indeed, the improvement of the solutions between times 30 and 60 seconds is quite weak.

Figure 13(b) represents three speed profiles obtained for the instance I1b. The corresponding numerical results are reported in Table 4. In this direction, there is now a steep climb which prevents the train to accelerate strongly, as depicted by the reference solution I1b* (Fig. 12(b)) where the speed increases less quickly between position 870 m and 1200 m. The solution I1b.1 introduces coastings and in particular in the last section around the position 1000 m. Due to the steep climb, the slowdown is quite strong compared to the coasting in the first section. The slowdown is directly followed by a little acceleration in the descent around position 1100 m. But from the position around 1300 m, the train decelerates gently until it has to brake when arriving at the destination (position 2000 m).



(a) Set of solutions



(b) Reference solution and two tradeoffs

Figure 12: Results obtained on instance 1a: (a) Fronts of solutions at times $t=\{0, 30, 60\}$ for outward journey; (b) Examples of speed profiles for outward journey.

The solution I2b.2 follows approximately the same trajectory than the solution I2b.1, but it needs more time and saves more energy.

Through these examples, the relevance of a multi-objective approach of speed profiling optimization is highlighted. Indeed, in a short computation time, the approach has been capable of producing a set of distinct solutions. Train-practitioners will have the possibility to choose a solution adapted to their needs instead of working with the shortest running time increased of a supplement. In this way, the proposed approach would help them to decide what a good tradeoff is between time supplement and energy consumption in the planning under consideration.

Table 4: Numerical results of three solutions obtained on instance 1b: reference solution I1b* and two alternative solutions I1b_1, I1b_2.

Solution	Duration [s]	Dev. [%]	Energy [kJ]	Dev. [%]
I1b*	174		7.59525×10^6	
I1b_1	191	+9.7	3.91279×10^6	-48.48
I1b_2	200	+14.9	3.49579×10^6	-53.97

6.3.2. Line 2

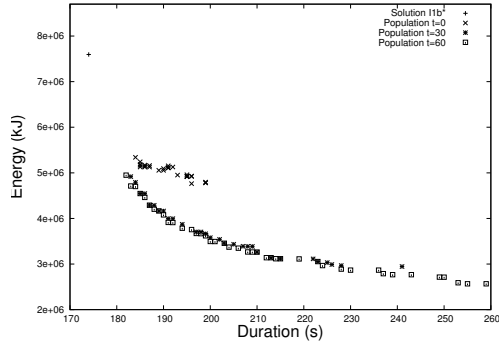
Outward journey. In this case, steep descents are present in the track profile. The interest of using a smoothing method will be highlighted in such a context to produce optimized speed profiles.

Figure 14(a) presents the sets of solutions at different times ($t=\{0, 30, 60\}$). As for the previous instances, the initial population ($t=0$ second) is compared to populations at $t=30$ and $t=60$ seconds. The reference solution I2* is also represented to have a basis of comparison. The journey duration is limited to $1.5 \times T_{I2^*}$. Similarly to what observed in instances I1a and I1b, the comparison of the three sets clearly indicates that the largest part of the optimization is done during the first half of the computation.

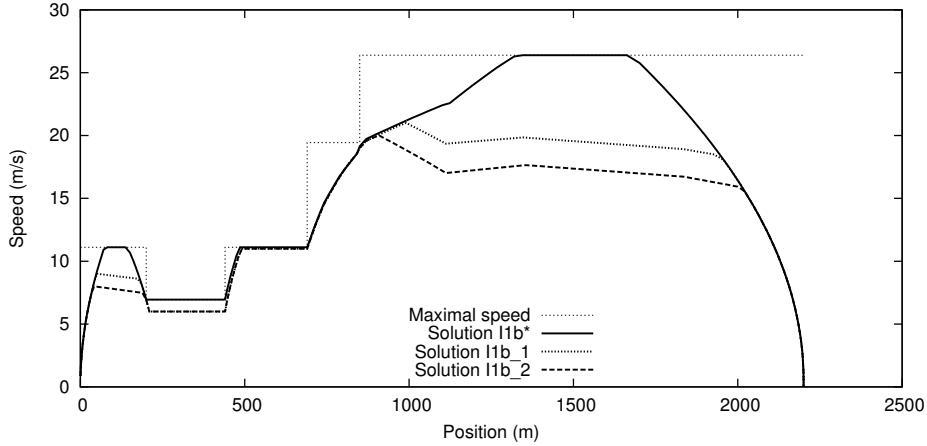
The gap in energy consumption between the reference solution and the others can be explained by the particular topology of the track. In this example, given that the train can move with low effort by using coasting, the engine can be utilized only little, so that the consumption falls dramatically.

In order to clarify this explanation, we can focus on the speed profile of solution I2a_1 in Figure 14(b). A coasting is introduced in all sections except the first. Moreover, still with the exception of the first section, the train does not use very much the acceleration regime to increase speed: indeed it can accelerate by coasting in descent. As soon as the train reaches the maximum speed by coasting in sections 2, 3, 4, 5, it must partially brake to maintain the speed, whereas in solution I2* the train has to accelerate at full power to reach the maximum speed. It is the reason why solution I2* requires far more energy than the other reported ones.

The acceleration occurring while entering in section 3 is due to the model proposed: $v_{e,3} = v_{x,2} < v_{a,3}$, and the sequence (Braking, Acceleration) is smoothed according to the method explained in Section 4.3. Indeed, given that the coasting from speed $v_{a,2}$ to $v_{b,2}$ brings to an acceleration, it is neces-



(a) Set of solutions



(b) Reference solution and two tradeoffs

Figure 13: Results obtained on instance 1b: (a) Fronts of solutions at times $t=\{0, 30, 60\}$ for inward journey; (b) Examples of speed profiles for inward journey.

sary to brake for exiting section 2 at speed $v_{b,2}$. Then, in section 3, speed $v_{a,3}$ is reached after accelerating as prescribed by the model. In such a case, due to the negative gradient, the model produces a sequence of driving regimes which will need to be smoothed: (Acceleration while coasting, Braking, Acceleration). In the same way, the accelerating effect in coasting occurring in sections 4 and 5 is also smoothed according to the same method. The smoothed speed profile, denoted I2a_1s, is depicted in Figure 14(b).

Inward journey. In the inward way, the gradient of the track is mostly positive except for the last three kilometers. The topography of the track implies

Table 5: Numerical results of two solutions obtained on instance 2: reference solution I2* and one alternative solutions I2_1 and its smoothed counterpart I2_1s.

Solution	Duration [s]	Dev. [%]	Energy [kJ]	Dev. [%]
I2a*	734		10.5194×10^6	
I2a_1	766	+4.35	7.26566×10^6	-30.93
I2a_1s	760	+3.54	6.61866×10^6	-37.08

for the train a greater energy consumption than in the outward journey. This implication can be observed on the energy consumption values of the solutions obtained during the optimization (Fig. 15(a)), which are far larger than those obtained for the outward journey.

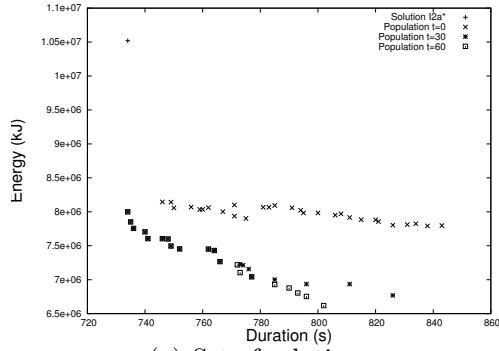
Figure 14(b) represents three speed profiles for the inward journey: the reference solution I2b* and two compromise solutions denoted I2b_1 and I2b_2. The detailed results are presented in Table 6. Contrary to the solutions obtained for the outward journey, a coasting introduced in one of the first four sections results in a slowdown.

If we consider the solution I2b_1, two coastings are respectively introduced in the fourth and the last section. The former is realized while the train moves in a climb. That results in a slowdown until reaching the speed to maintain. The latter coasting results first in a slight slowdown directly followed by an acceleration due to the negative gradient at the end of the track.

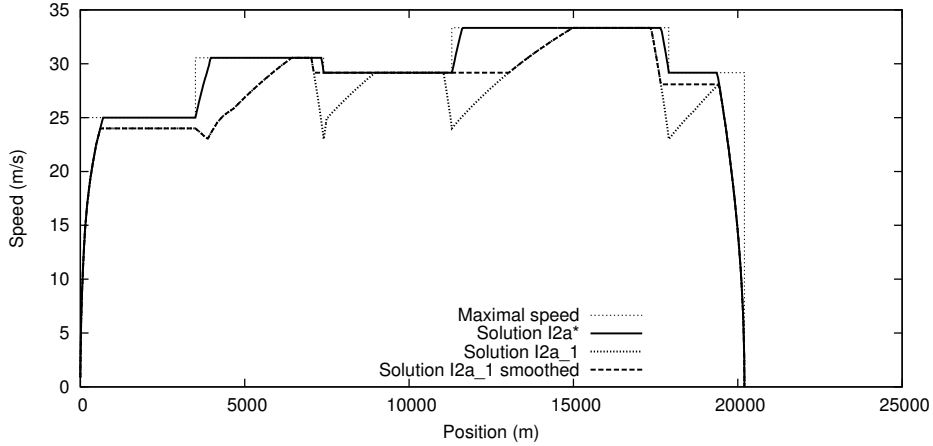
In the speed profile of the solution I2b_2, three coastings are used all along the journey. The first two appear respectively in the second and the fourth section, and result in a slowdown. In the last section, the coasting begins with a slight slowdown before accelerating due to the steep descent.

Table 6: Numerical results of three solutions obtained on instance 2b: reference solution I2b* and two alternative solutions I2b_1 and I2b_2.

Solution	Duration [s]	Dev. [%]	Energy [kJ]	Dev. [%]
I2b*	731		24.5206×10^6	
I2b_1	747	+2.18	19.5501×10^6	-20.27
I2b_2	786	+7.52	19.1889×10^6	-21.74



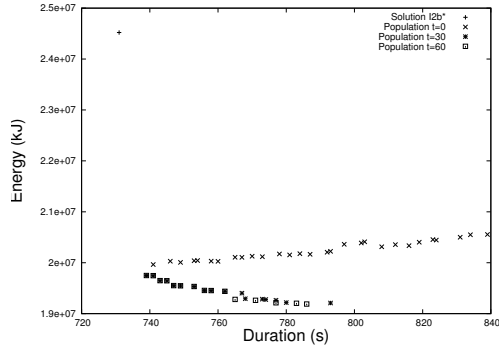
(a) Set of solutions



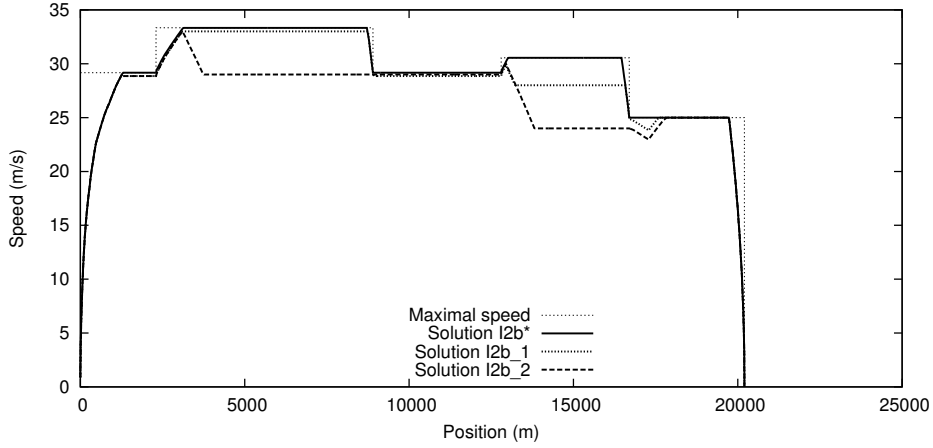
(b) Reference solution and two tradeoffs

Figure 14: Results obtained on instance 2b: (a) Fronts of solutions at times $t=\{0, 30, 60\}$ for inward journey; (b) Examples of speed profiles for inward journey.

The results reported in Tables 5 and 6 show the duration increases and energy savings occurring while using energy-free driving regimes such as coasting or cruising with partial braking. As for line 1, the approach presents a good capacity of producing a set of distinct solutions in a short computation time. Its relevance in decision-aid is shown in so far as it could help train-practitioners to build timetables.



(a) Set of solutions



(b) Reference solution and two tradeoffs

Figure 15: Results obtained on instance 2b: (a) Fronts of solutions at times $t=\{0, 30, 60\}$ for inward journey; (b) Examples of speed profiles for inward journey.

7. Conclusion and perspectives

In this paper, we consider a particular step of the train planning. This step precedes the timetabling and concerns the calculation of running times. This calculation is directly related to the construction of speed profiles, which indicate the speed that the train must hold at each position. In this work, as power recovery is not integrated, this problem is addressed with an approach that uses conventional driving regimes following the Maximum Principle: acceleration at full power, cruising, coasting and maximum braking.

Since timetabling process uses running times increased of a short time

as supplement to prevent disturbances, the approach proposed here consists in defining a running time directly adapted to the needs of planning and optimizing also the energy consumption. Hence, one major contribution of this work is to provide to the practitioners the capability of choosing the solution the most adapted to their needs directly among a set of compromise solutions.

In order to build a set of speed profiles adapted to a track, a specific model has been developed. This model uses two target-speeds per section as decision variables, and builds the speed profile within the section under consideration. The optimization is performed by an evolutionary multi-objective algorithm.

The optimization model is independent of the train dynamics model that we implemented. Indeed, due to the controversial aspects of certain approximations, anyone can use a different model of train dynamics without changing neither the method of speed profiling nor the optimization model. That is clearly an advantage for further comparisons of train dynamics models in the future.

For illustrating the relevance of the approach, two lines have been proposed and studied in both outward and inward directions, that is, four instances in all. They have been tackled with a multi-objective optimization algorithm, which has produced a set of solutions in a short time (60 seconds). Moreover, as observed in the different experiments, satisfying results have been obtained from around 30 seconds of computation.

In the future, we will integrate the calculation of alternative running times in a planning context while building timetables. This will imply the consideration of whole trips and not only point-to-point journeys and of several trains at once, for optimizing their synchronization. It will make the use of regenerative braking relevant for transferring recovered energy from one train to another. This integration would allow the train-practitioners to consider the energy spent per timetable produced with regards to the service (dwell and transfer times, connections).

References

- [1] T. Albrecht, *Railway Timetable & Traffic – Analysis, Modelling, Simulation*, Eurail press, 2008, Ch. Energy-Efficient Train Operation, pp. 83–105.
- [2] R. Liu, I. Golovitcher, Energy-efficient train control, *Transportation Research part(A)* 37 (2003) 917–932.

- [3] M. Lüthi, Improving the efficiency of heavily used railway networks through integrated real-time rescheduling, Ph.D. thesis, ETH Zurich (2009).
- [4] T. Albrecht, S. Oettich, Computers in Railways VIII, WIT Press, Southampton, 2002, Ch. A new integrated approach to dynamic schedule synchronization and energy saving train control, pp. 847–856.
- [5] A. D’Ariano, T. Albrecht, Running time re-optimization during real-time perturbations, in: Computers in Railways X, Prague, Czech Republic, 2006, pp. 531–540.
- [6] Y. Wang, B. De Schutter, B. Ning, N. Groot, T. J. van den Boom, Optimal trajectory planning for trains using mixed integer linear programming, in: ITSC’2011, 14th Int. IEEE Conf. on Intelligent Transportation Systems, Washington, DC, USA, 2011, pp. 1598–1603.
- [7] H. Ko, T. Koseki, M. Miyatake, Application of dynamic programming to optimization of running profile of a train, in: Computers in Railways IX, 2004, pp. 103–112.
- [8] M. Miyatake, H. Ko, Optimization of train speed profile for minimum energy consumption, IEEJ Transactions on Electrical and Electronic Engineering 5 (2010) 263–269.
- [9] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization (11) (1997) 341 – 359.
- [10] C. Chang, D. Xu, H. Quek, Pareto-optimal set based multiobjective tuning of fuzzy automatic train operation for mass transit system, IEE Proceedings–Electric Power Applications 146 (5) (1999) 577–583.
- [11] R. Chevrier, An evolutionary multi-objective approach for speed tuning optimization with energy saving in railway management, in: ITSC’2010, 13th Int. IEEE Conf. on Intelligent Transportation Systems, Madeira (Portugal), 2010, pp. 279–284.
- [12] R. Chevrier, G. Marlière, B. Vulturescu, J. Rodriguez, Multi-objective evolutionary algorithm for speed tuning optimization with energy saving in railway: Application and case study, in: RailRome 2011, the 4th

International Seminar on Railway Operations Research, Rome, Italy, 2011.

- [13] L. Kroon, R. Dekker, M. Vromans, Cyclic railway timetabling: A stochastic optimization approach, in: F. Geraets, L. Kroon, A. Schoebel, D. Wagner, C. Zaroliagis (Eds.), *Algorithmic Methods for Railway Optimization*, Vol. 4359 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2007, pp. 41–66.
- [14] A. D’Ariano, D. Pacciarelli, M. Pranzo, Assessment of flexible timetables in real-time traffic management of a railway bottleneck, *Transportation Research part(C)* 16 (2008) 232–245.
- [15] T. Huisman, R. J. Boucherie, Running times on railway sections with heterogeneous train traffic, *Transportation Research part(B)* 35 (2001) 271–292.
- [16] R. Goverde, Punctuality of railway operations and timetable stability analysis, Ph.D. thesis, TU Delft (2005).
- [17] K. Deb, *Multi-objective Optimization using Evolutionary Algorithms*, Wiley, 2001, 517 p.
- [18] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: *Parallel Problem Solving from Nature - PPSN VIII*, Vol. 3242/2004, Springer, Berlin/Heidelberg, 2004, pp. 832–842.
- [19] J. Pachl, *Railway Operation and Control*, VTD Rail Publishing, 2004, 239 p.
- [20] O. Brünger, E. Dahlhaus, *Railway Timetable & Traffic – Analysis, Modelling, Simulation*, Eurail press, 2008, Ch. Running Time Estimation, pp. 58–82.
- [21] P. Howlett, P. J. Pudney, *Energy-efficient train control*, Springer, 1995, 306 p.
- [22] D. Lancien, M. Fontaine, Calculs de marches de train économisant l’énergie de traction – le programme MARECO, *Revue Générale des Chemins de Fer* (1981) 679–692.

- [23] S. Iwnicki (Ed.), *Handbook of Railway Vehicle Dynamics*, Taylor and Francis Group, 2006, 535 p.
- [24] W. Davis Jr, Tractive resistance of electric locomotives and cars, *General Electric Review* 29 (1926) 685–708.
- [25] E. G. Talbi, *Metaheuristics: from design to implementation*, Wiley, 2009, 624 p.
- [26] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Transactions on Evolutionary Computation* 7 (2) (2003) 117–132.
- [27] K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, *Complex Systems* 9 (1995) 115–148.
- [28] A. Liefoghe, L. Jourdan, E.-G. Talbi, A software framework based on a conceptual unified model for evolutionary multiobjective optimization: ParadisEO-MOEO, *European Journal of Operational Research* 209 (2) (2011) 104–112.