Multi-objective Evolutionary Algorithm for Speed Tuning Optimization with Energy Saving in Railway: Application and Case Study

Rémy Chevrier, Grégory Marlière, Bogdan Vulturescu, Joaquin Rodriguez

Multi-objective Evolutionary Algorithm for Speed Tuning Optimization with Energy Saving in Railway: Application and Case Study

[1]Rémy Chevrier, [1]Grégory Marlière, [2]Bogdan Vulturescu, [1]Joaquín Rodriguez
[1]Université Lille Nord de France F59000 Lille
INRETS ESTAS, 20 rue Élisée Reclus – F59650 Villeneuve d'Ascq, France
Email: {remy.chevrier, joaquin.rodriguez, gregory.marliere}@inrets.fr
[2]INRETS LTN, 25 Allée des Marronniers – F78000 Versailles, France
Email: bogdan.vulturescu@inrets.fr

Abstract
In this paper an approach is proposed for optimization of speed profiles in railway integrating energy saving. This approach deals with a multi-objective problem involving three criteria: reduction of the travel duration, reduction of the delays and minimization of the energy consumption. Based on a state of the art evolutionary algorithm, the proposed approach searches for diversified solutions in a continuous search space. These solutions are evaluated and compared according to the Pareto approach in such a way that the proposed solutions are different and incomparable, in order to help the decision makers. After having reminded railway dynamics elements, the approach is detailed as well as the evolutionary algorithm and its problem-related components. Two case studies (the Gonesse junction and the line Saint-Étienne – Rive de Giers in France) and results are then provided and analyzed.

Keywords
Multi-objective, Evolutionary algorithm, Energy saving, Railway management, Optimization

1 Introduction

For recent years, the concern due to pollution and global warming led to develop more eco-aware transportation systems. Energy spent in the railway management is used to move the trains and if speed tuning is well suited, the energy consumption will be lower than in other cases. Indeed, since the acceleration phases consume a huge quantity of energy, it is necessary to tune the speeds according to the distances, the slopes and the timetable in order to avoid unnecessary sequence of brake/acceleration phases. Moreover, given that the trains have a big inertia, it is judicious to use this physical property by stopping the engine and letting the train advance just thanks to the initial force [15, 12].

The railway management involves multiple objectives which are often antagonist such as the travel duration and the energy consumption. Although the travel duration had often preference of the decision makers, for recent years the criterion of energy consumption is considered from an equivalent point of view. Indeed, global energy saving has become the new challenge of the transportation systems including railway.

Works have been led for analytically computing speed tuning (ST) solutions according to several levels of delay tolerance [2, 17, 1]. But to our knowledge, there are few multi-objective approaches yet. These are based on Differential Evolution [19] for mass transit system [4] or evolutionary algorithms hybridized or not [9]. Nevertheless, to our knowledge few approaches propose to optimize the energy consumption which becomes the new challenge of the decade.

Thus, in this paper we deal with a multi-objective optimization of ST with energy saving. These concurrently optimized criteria are: the minimization of the travel duration; the minimization of the energy consumption and the punctuality (minimization of the delays, which is quite different to reducing the travel duration). According to the timetable, we can work out the delays occurring at stations and use them in a multi-criteria search. Whatever the criteria under consideration, the main goal consists in designing ST solutions diversified enough to help decision makers to choose the solution the most adapted to their needs. In order to solve the problem, we propose an approach based on a Pareto evolutionary algorithm which extends a previous work [5].

After defining the problem we propose to solve in Section 2, we present our model of speed tuning in Section 3. The evolutionary computation principles are presented in Section 4 by explaining also the Indicator-Based Evolutionary Algorithm we use to compute ST solutions. Experimental results based on real data are then provided and discussed in Section 5. These examples are proposed to assess the algorithm with two objectives on the Gonesse junction (France) and with three objectives on the line Saint-Étienne – Rive de Giers (France). Finally Section 6 concludes the paper.

2 Problem overview

The main goal consists in designing the most suited speed profile over space. The space corresponds to a sequence of intervals I (block sections) in which the speeds can be changed. Figure 1 represents the decomposition of a one-section journey in four steps. A maximum speed v_{max} limits the train speed. According to this limit and the train parameters the speed can be defined in each step. The first step (A) corresponds to the train acceleration when the speed grows from 0 to v_{max} (if the train can reach v_{max}). Before dealing with the cruising and coasting phases it is necessary to compute the braking phase (B) to be sure that the needed braking distance will not exceed the remaining distance before the end. The cruising phase (Cr) corresponds to the speed maintaining, that is, a null acceleration when the traction effort equals the resistance to the train advance. The coasting phase (Co), depicted by the dashed lines on Figure 1, is engaged when the engine is stopped and the train moves thanks to its inertia. During this phase, no energy is consumed and hence in order to reduce the energy consumption it is interesting to vary the instant (or position) from which the engine is stopped and the coasting phase is started (see points 1, 2, 3 in Fig. 1). The sooner the coasting phase starts the greater the economy but the later the train will arrive. Thus the goal of the problem solving is to determine a good tradeoff between energy consumption, running time and delay occured.

3 Speed tuning model

Naturally, the four-steps model explained above cannot be applied everywhere and the shape of speed profile depends on the entrance speed v_0 (position 0) and the exit speed v_X (po-

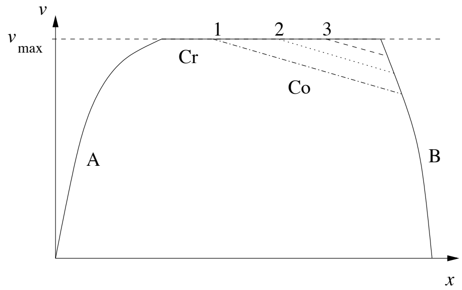


Figure 1: Speed tuning over space in four steps: acceleration (A), cruising (Cr), coasting (Co) and Braking (B)

sition X). Between these two positions 0 and X it is necessary to determine the speeds according to a chosen policy. In this way, we introduce two intermediate speeds v_1 and v_2 which help us to build the speed profile.

A train path is composed of n sections. Therefore, for a section S, we have a set of five speeds: $v^S_{max}, v^S_0, v^S_1, v^S_2, v^S_X$. When the train starts its journey, speed v^1_0 is null for section 1 ($v^1_0 = 0$), while in the arrival section speed $v^n_X = 0$. When the train leaves a section S and enters in the following section ($S+1$), the exit speed of section S equals the entrance speed of section $S+1$: $v^S_X = v^{S+1}_0$ in such a way that: $v^S_X \leq \min(v^S_{max}, v^{S+1}_{max})$. Speeds v_1, v_2 to be determined are limited by the maximum speed of the section S: $v^S_1 \leq v^S_{max}$ and $v^S_2 \leq v^S_{max}$.

Taking these elements into account, we generalize for a sequence of sections that three speeds have to be determined per section: v_1, v_2 and v_X. These values will be searched by the evolutionary algorithm we propose in Section 4. Speed profile is determined according to a three steps model:

1. the entrance phase tunes speed for accelerating/decelerating from v_0 to v_1;

2. the exit phase is assessed before the intermediate phase for varying speed from v_2 to v_X;

3. the intermediate phase tunes the speed from v_1 to v_2 according to our policy depending on $v_1 > v_2$ or not.

The set of basic definitions useful for the remainder of the paper are summarized in Table 1.

3.1 Objectives

The problem can be represented as a set Φ of $n \leq 3$ objective functions to be minimized. The first function φ_1 represents the minimization of the travel duration whereas the energy consumption reduction is illustrated by function φ_2. Note that this minimization is related to the reduction of the mechanical energy. The amount of durations corresponds to the sum of all durations needed to travel within the sections. The third objective φ_3 function aims at minimizing the delays.

T	a travel duration which corresponds to the sum of intermediate durations (unit [s])
D	the amount of delays occurred at each station (unit [s])
E_m	the mechanical energy necessary to move the train (unit [J])
$P_m(t)$	the mechanical power delivered at instant t (unit [W]);
$F_T(t)$	the traction effort at instant t (unit [N]);
$F_R(t)$	the resistance to the advance at instant t (unit [N]);
$L_R(t)$	the line resistance to the advance at instant t (unit [N]);
$B_R(t)$	the braking effort at instant t (unit [N]);
$v(t)$	the train speed at instant t (unit [m/s]);
v_0	the entrance speed of a train (unit [m/s]);
v_X	the exit speed of a train (unit [m/s]);
$a_b(t)$	the braking at instant t (unit [m/s^2]);
$a(t)$	the acceleration at instant t (unit [m/s^2]);
m	the train mass (unit [kg]);
ρ	the mass correction factor usually set to 1.04.

Table 1: Definition of the main symbols used in the paper

$$\Phi = (\varphi_1, ..., \varphi_n), n \leq 3 \tag{1}$$
$$\varphi_1 = \min T \tag{2}$$
$$\varphi_2 = \min E_m \tag{3}$$
$$\varphi_3 = \min D \tag{4}$$
$$E_m = \int P_m(t)\, dt \tag{5}$$
$$P_m(t) = F_T(t)\, v(t) \tag{6}$$

In function of the needs, we can add or remove objectives. The basic objective consists in minimizing the travel duration, hence the minimal formulation of the problem is $\Phi = (\varphi_1)$ and this formulation will allow to work out reference solutions for fairly evaluating mult-objective solutions. In addition to this basic objective, we can add one or two objectives such that the problem can be formulated as follows: $\Phi = (\varphi_1, \varphi_2)$ for optimizing two objectives or $\Phi = (\varphi_1, \varphi_2, \varphi_3)$ for optimizing three objectives.

3.2 Elements of railway dynamics

The fundamental equation of dynamics states that the relation between the forces, mass and acceleration:

$$F_T(t) - F_R(t) = \rho\, m\, a(t)$$

Note that ρ is a mass correction factor usually set to $\rho = 1.04$ [12, 1]. The train data also depict the traction effort profile which indicates effort $F_T(t)$ according to a speed $v(t)$. The resistance to the train advance $F_R(t)$ corresponds to the sum of the resistances, i.e. the line resistance L_R and the braking efforts if the train brakes:

$$F_R(t) = L_R(t) + B_R(t)$$

Line resistance L_R is defined according to the slope and its angle β:

$$L_R(t) = \rho\, m\, g\ \sin\beta, \text{ where } g = 9.81\text{m/s}^2$$

The braking effort $B_R(t)$ at instant t is depicted by a braking profile related to the train and depending on speed $v(t)$. Thanks to this profile, we can determine the braking $a_b(t)$ as follows:

$$a_b(t) = \frac{B_R(t)}{\rho\, m}$$

Now, with these elements we can determine the acceleration, cruising, coasting and braking phases. We note that only acceleration and cruising phases need energy.

Acceleration

This can be defined as follows:

$$a(t) > 0 \quad \Leftrightarrow \quad F_T(t) > F_R(t) \tag{7}$$

$$a(t) \quad = \quad \frac{F_T(t) - F_R(t)}{\rho m} \tag{8}$$

$F_T(t)$ and $F_R(t)$ are calculated according to speed $v(t)$ as explained before. The acceleration phase from speed v_A to v_B ($v_A < v_B$) is iterated each second (instant i) and updates the acceleration $a(i)$, the speed $v(i)$ and the position $x(i)$ (Algorithm 1).

Algorithm 1: Calculation of an acceleration phase

$v(i) = v_A$;
while $v(i) < v_B$ **do**
  Update position $x(i+1) = 0.5a(i) + v(i) + x(i)$;
  Update acceleration $a(i+1)$ according to $v(i)$;
  Update speed: $v(i+1) = v(i) + a(i)$;
  Update energy: $E_m = E_m + E_m(i)$ with $E_m(i) = F_T(i) \times v(i)$;
  Update duration: $T = T + 1$;
  $i = i + 1$
end

Cruising

The cruising phase maintains the train speed v from position x_0 along a distance d without accelerating:

$$a(t) = 0 \ \Leftrightarrow \ F_T(t) = F_R(t)$$

Algorithm 2 explains the computation of the cruising.

Coasting

During a coasting phase the engine is stopped: $F_T(t) = 0$. If the slope is null or positive $\beta \geq 0.0$, the speed decreases because $a(t) < 0$. Otherwise the slope is negative ($\beta < 0.0$) and the speed will increase. In this latter case, it will be necessary to brake.

The calculation of this phase is very close to an acceleration except for the energy consumption which stays null. The coasting between speed v_A and v_B ($v_A > v_B$) is iterated each second (Algorithm 3) while the crossed distance does not overtake the available distance d for the coasting.

Algorithm 2: Calculation of a cruising phase

while $x(i) < x_0 + d$ **do**
    Update position: $x(i+1) = x(i) + v$;
    Calculate $F_R(v)$ according to v and the gradient β ;
    Set $F_T(v) = F_R(v)$;
    Update energy: $E_m = E_m + F_T(i) \times v$;
    Update duration: $T = T + 1$;
    $i = i + 1$
end

Algorithm 3: Calculation of a coasting phase

$v(i) = v_A$;
while $(x(i) < x_0 + d)$ *and* $(v(i) > v_B)$ **do**
    Update position $x(i+1) = 0.5a(i) + v(i) + x(i)$;
    Update acceleration $a(i+1) = \frac{-F_R(i)}{\rho \times m}$;
    Update speed: $v(i+1) = v(i) + a(i)$;
    Update duration: $T = T + 1$;
    $i = i + 1$
end

Braking
The braking phase combines two resistance forces: the resistance to the train advance and the service braking force. So the calculation is identical as in a coasting phase except for determining the acceleration: $a(i+1) = \frac{-F_R(i)}{\rho \times m} - a_b(i)$ with $a_b(i)$ depending on $v(i)$ and $F_R(i))$ subject to the gradient of the slope.

3.3 Entrance and exit phases

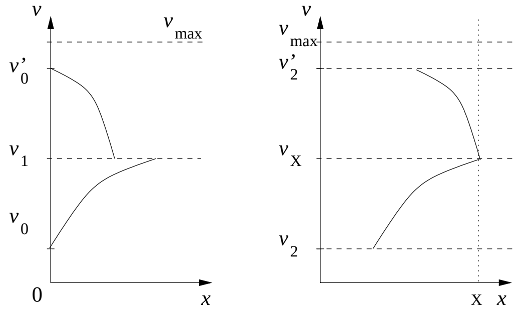
Before evaluating the intermediate phase it is necessary to determine on the one hand the entrance phase and on the other hand the exit phase. For each one, duration, distance and energy are calculated according to v_0 compared with v_1 and v_2 compared with v_X. Figure 2 depicts the possible cases described below.

Entrance
The entrance phase is determined according to v_0 and v_1. Two cases may arise (Fig. 2(a)):

1. if $v_0 < v_1$ then an acceleration occurs, increasing speed from v_0 to v_1;

2. if $v_0 > v_1$ then a braking is done, decreasing speed from v_0 to v_1 (illustrated by v_0' and v_1).

In all these cases, a distance d_0 is needed to vary the speed during T_0. The consumed energy E_0 is null if it is a braking phase, otherwise $E_0 > 0$.



(a) (b)

Figure 2: Scheme of (a) entrance and (b) exit phases

Exit
The exit phase is done in the same way by using v_2 and v_X and two cases may also arise (Fig. 2(b)):

1. if $v_2 < v_X$ then an acceleration occurs, increasing speed from v_2 to v_X;

2. if $v_2 > v_X$ then a braking is done, decreasing speed from v_2 to v_X (illustrated by v_2' and v_X).

A distance d_X and a duration T_X are needed for this phase. The consumed energy E_X is positive ($E_X > 0$) if the phase is an acceleration.

3.4 Intermediate phase

Once the exit phase is computed, the feasibility of the solution must be checked. Indeed the travelled section has an available distance d_S and we must be sure that $d_0 + d_X < d_S$ in so far as an available distance remains to allow varying the speed from v_1 to v_2 during the intermediate phase.

Let $d_I = d_S - d_0 - d_X$ be the available distance to vary the speed from v_1 to v_2. Three cases may arise:

1. if $v_1 > v_2$ and the slope is null or positive, then we try to insert a coasting phase to decrease the speed and to save energy. If it is possible, we insert a cruising phase before the coasting for completing all the distance available (Fig. 3(a), the plain line). The only consumed energy ($E_I > 0$) is due to the cruising phase. When the distance is not enough to do a complete coasting then a braking phase from v_1 to v_2 must be calculated and we search for intersection of coasting and braking phases (Fig. 3(a), the dashed line) and in this case $E_I = 0$;

2. if $v_1 > v_2$ and the slope is negative (it is a descent), then train can accelerate without effort. We compute the acceleration on the distance d_I and the braking from v_{max} to v_2. Then the intersection point of acceleration and braking has to be found and the intermediate phase is achieved (Fig. 3(b));

3. if $v_1 < v_2$ then an acceleration from v_1 to v_2 will be necessary and we insert it halfway through (travelled distance d_A). Cruising phases are added before and after the acceleration, the first at speed v_1 and the second at speed v_2 (Fig. 3(c)). The cruising phases are done on the same distance ($(d_I - d_A)/2$). The consumed energy corresponds to the amount of energy required for each phase.

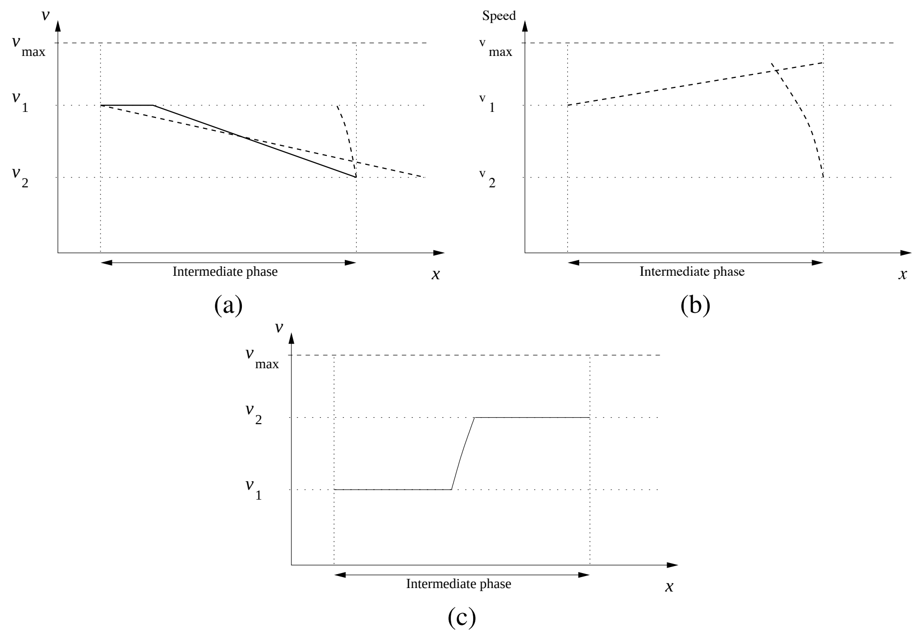


Figure 3: Three cases of intermediate phase: (a) $v_1 > v_2$ and $\beta \geq 0.0$, (b) $v_1 > v_2$ and $\beta < 0.0$, (c) $v_1 < v_2$

Naturally, it is necessary to check whether each phase can be inserted according to the remaining distance or not. If the distances do not allow to insert the chosen phases, the solution is marked as not feasible and penalized during the evaluation in the evolutionary algorithm.

4 Evolutionary Multi-objective Optimization

An evolutionary algorithm (EA) is an iterative process of exploratory search. Our choice is led by a preference to obtain a set of sufficiently diversified solutions in a single run. Indeed, the evolutionary algorithms with Pareto approach are capable to produce well-spread incomparable solutions along the Pareto front. That could be an advantage to help the decision makers in the case of real-life problems [20].

An EA is a nature-inspired metaheuristic gathering a set of solutions (*individuals* or *chromosomes*): a *population*. The latter evolves while recombining pairwise individuals in such a way that new original and improved solutions are produced. A mutation operator allows to diversify the population while randomly modifying solutions. These new solutions are added to a temporary population which will partially or totally constitute the population at the next iteration, depending on the algorithm policy of the population renewal. Algorithm 4 presents the main steps of a general purpose EA.

Algorithm 4: Canonical evolutionary algorithm

Population initialization;
while *Stopping criterion not reached* **do**
  Evaluate each solution in population $\mathcal{P}_i$;
  Select individuals in population $\mathcal{P}_i$ for crossover;
  Cross individuals according to a crossover rate;
  Mutate individuals according to a mutation rate;
  Population $\mathcal{P}_{i+1}$ generation (selection for replacement);
end

4.1 Multi-objective Optimization

A general Multi-objective Optimization Problem (MOP) can be defined by a set of n objective functions $(f_1, f_2, \ldots, f_n)$, a set X of feasible solutions in the decision space, and a set Z of feasible points in the objective space. Without loss of generality, we here assume that each objective function is to be minimized. To each solution $x \in X$ is assigned an objective vector $z \in Z$ on the basis of the vector function $f : X \to Z$ with $z = f(x) = (f_1(x), f_2(x), \ldots, f_n(x))$ as illustrated by Figure 4. An objective vector $z \in Z$ is said to *dominate*[1] another objective vector $z' \in Z$ *iff* $\forall i \in \{1, 2, \ldots, n\}$, $z_i \leq z'_i$ and $\exists j \in \{1, 2, \ldots, n\}$ such as $z_j < z'_j$. An objective vector $z \in Z$ is said to be *non-dominated iff* there does not exist another objective vector $z' \in Z$ such that z' dominates z. A solution $x \in X$ is said to be *efficient* if its mapping in the objective space results in a non-dominated point. The set of all efficient solutions is the *efficient set*, denoted by X_E. The set of all non-dominated vectors is the *Pareto front*, denoted by Z_N. A possible approach in MOP solving is to find the minimal set of efficient solutions, *i.e.* one solution $x \in X_E$ for each non-dominated vector $z \in Z_N$ such as $f(x) = z$. However, generating the entire efficient set is usually infeasible due to the complexity of the underlying problem. Therefore, the overall goal is often to identify a good approximation of it. EAs are commonly used to this end as they are able to find multiple and well-spread non-dominated solutions in a single simulation run [6].

4.2 Indicator Based Evolutionary Multi-objective Algorithm

Although there exists several state-of-the-art multi-objective EAs (NSGA-II [8], SPEA2 [21]), we use Indicator Based Evolutionary Algorithm [22]. Indeed IBEA is a method more modern than the most widely used EA: NSGA-II. This is a good illustration of the new trend dealing with indicator-based search, and started to become popular for recent years. The main idea behind IBEA is to introduce a total order between solutions by means of a binary quality indicator. Its fitness assignment scheme is based on a pairwise comparison of solutions from the current population with regards to an arbitrary indicator I. To each individual x is assigned a fitness value $F(x)$ measuring the 'loss in quality' if x was removed

[1] We will also say that a decision vector $x \in X$ *dominates* a decision vector $x' \in X$ if $f(x)$ dominates $f(x')$.

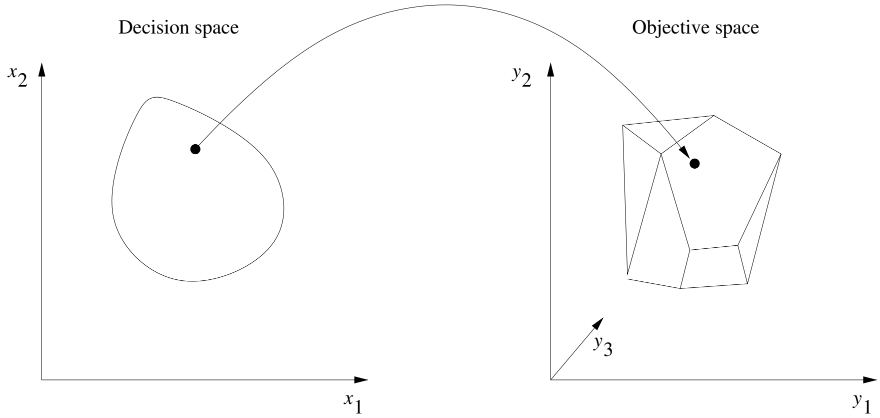


Figure 4: Representation of a solution (x_1, x_2) in the decision space and the corresponding values in the objective space: $(y_1, y_2, y_3) = f(x_1, x_2)$.

from the current population P, *i.e.* $F(x) = \sum_{x' \in P \setminus \{x\}} (-e^{-I(x',x)/\kappa})$, where $\kappa > 0$ is a user-defined scaling factor. Different indicators can be used for such a purpose, and we here choose to use the binary additive ϵ-indicator ($I_{\epsilon+}$) as defined in [22]. $\mathrm{I}_{\epsilon+}(x, x')$ gives the minimum value by which a solution $x \in X$ has to or can be translated in the objective space to weakly dominate another solution $x' \in X$. Selection for reproduction consists of a binary tournament between randomly chosen individuals. Selection for replacement consists of iteratively removing the worst solution from the current population until the required population size is reached; fitness information of the remaining individuals is updated each time there is a deletion.

4.3 Solution encoding and initialization

A solution is defined by a vector of real values. Each value corresponds to one speed. Given that three speeds are necessary to represent a section, we can state the vector length l equals three times the number of sections (n): $l = 3n$.

Section 0				Section n		
v_1^0	v_2^0	v_X^0	...	v_1^n	v_2^n	v_X^n

Figure 5: Pattern of a solution: a vector of real values for the speeds

Even if the speeds are bounded by maximum entrance, exit and section speeds, three ranges of speeds are defined: slow, middle and high speeds in such a way that one third of the solutions are either slow, middle or high speed tuned. For example, let S_1, S_2 be two sections with respective maximum speeds: 90 km/h and 120 km/h. Three ranges of values are possible: ($[0, 30], [0, 40]$), ($[30, 60], [40, 80]$) and ($[60, 90], [80, 120]$). Such a mechanism is useful to bring a good diversity in the initial population by designing more or less fast solution and inversely more or less energy expensive. Then, the initialization is done by randomly assigning speeds.

4.4 Operators

Evaluation
As we mentioned before, our MOP is composed of two objective functions ($\Phi = (\varphi_1, \varphi_2)$). Objective function φ_1 corresponds to the minimization of the amount of the durations T_i needed for each section i: $\varphi_1 = \min T$ with $T = \sum_{i=1}^{n} T_i$. Objective function φ_2 is in charge of the reduction of the energy consumption: $\varphi_2 = \min E$ with $E = \sum_{i=1}^{n} E_i$. This global consumption equals the amount of energy required for each section i. Each section is evaluated according to the model presented before in such a way that we know pair (T_i, E_i) of each section i.

Recombination
The recombination step is done by means of two operators: crossover and mutation. Since we deal with a continuous problem, we use operators specifically designed for this kind of problem: the Simulated Binary crossover (SBX) [7] and similarly, the mutation is based on a polynomial mutation adapted to search over continuous space.

5 Experimental results

The results presented in this section concern two locations in France. The first example is based on the Gonesse junction (near Paris) and the second represents the line from Saint-Étienne to Rive de Giers with one stop.

5.1 Implementation and Parameter Setting

In order to develop our approach, we use framework ParadisEO in which a lot of metaheuristics are implemented [16]. This tool is a white box in which the different steps of the algorithms have to be defined. In the case of IBEA the user has to implement crossover, mutation and evalutation steps and also the problem-related components.

The population of 50 individuals evolves over $1,000$ generations. Crossover (x_r) and mutation (m_r) rates are respectively set to $x_r = 0.9$ and $m_r = 0.5$.

5.2 Case study 1: Gonesse junction

Description
Here is proposed a real-life case study: the Gonesse junction is crossed by a train ($m = 180,000$kg). Eight sections are used for our example whose results are depicted in Figure 6. The plain line indicates the maximum speeds of the sections. The path has 14,285m of length and eight sections are crossed. Each section is limited by a maximum speed. This case study is interesting because there are two sections with switch points (sections 2 and 5). These sections are very short (resp. 150m and 90m). Furthermore the switch points sections have slow speed limit (60 km/h) whereas the other sections have much higher speed limits (until 200 km/h). So, these sections bring about braking phases to be managed at best. In this example, we assume there is no slope (the gradients are null).

Performance assessment

Here we propose to discuss three solutions obtained by our approach. Let S^* be a solution to the mono-objective problem consisting in just minimizing the travel duration. Let S_1 and S_2 be two solutions to the bi-objective problem and compared with S^*. The results are provided in Table 2 and the deviation compared with S^* is also reported.

Sol.	E [kJ]	Deviation [%]	T [s]	Deviation [%]
S^*	507,618		484.13	
S_1	378,683	-24.5%	508.79	+4.9%
S_2	246,245	-51.5%	555.63	+14.7%

Table 2: Energy consumption and duration of solutions S^*, S_1, S_2

Figure 6 illustrates the speed profile of each solution. It is interesting to note that solutions S_1, S_2 have big coasting phases, that is why these are less energy expensive than S^*. Besides, we can see that S^* has very long acceleration phases which are very energy expensive. The differences of consumption can also be observed in Figures 7(a,b,c) which depict the produced effort and highlight that Solution S^* consumes more energy than the others. If we focus on the energy saving compared with the delay, we can note that with around 5% of time in more, we can save around 25% of energy. Moreover, if we extend the delay to around 15% of time in more, we can save till around more than half of energy. That proves the interest to take some seconds in more to travel for saving a lot of energy.

Furthermore, the algorithm as well as the underlying method have proved their capability to provide a set of diversified and incomparable solutions.

5.3 Case study 2: line Saint-Étienne - Rive de Giers

Description

In this example, a train travels from Saint-Étienne (SE) to Rive de Giers (RG) by stopping at Saint Chamond (SC) during 1 minute. The train used for the example is an AGC[2] which is a light passenger train used for the little regional travels. Data about resistance can be found in [18] and about effort to rim in [14]. A timetable is predefined and depicted on Figure 8. The train starts from SE at 0 and must arrive at SC before $h^1_A = 420$. After stopping during 60 seconds, the train leaves the station at $h^1_D = 480$ for arriving at RG before or at $h^1_A = 960$. The total crossed distance is 20,200m.

Moreover, in order to be more accurate in the description of the problem the slopes are taken into account. The gradients over position are depicted on Figure 9(a) and are defined in n meters for 1,000. Indeed, since $\tan x$ is very close to $\sin x$ for little values of x, it is usual to compute the line resistance by taking $n/1000$ instead of $\sin\beta$ [3]. Therefore, the line resistance can be approximated as follows: $L_R(t) = \rho\, m\, g\, n/1000$.

Contrary to the previous example in which we assume there is no slope, a coasting could not be a slowdown given that the gradients can be negative (fig. 9(a)). Indeed, it would be interesting to manage at best the descents by properly adjusting the speeds in order to avoid to accelerate unnecessarily when the train can increase just thanks to the slope.

[2] *Autorail Grande Capacité* built by Bombardier.

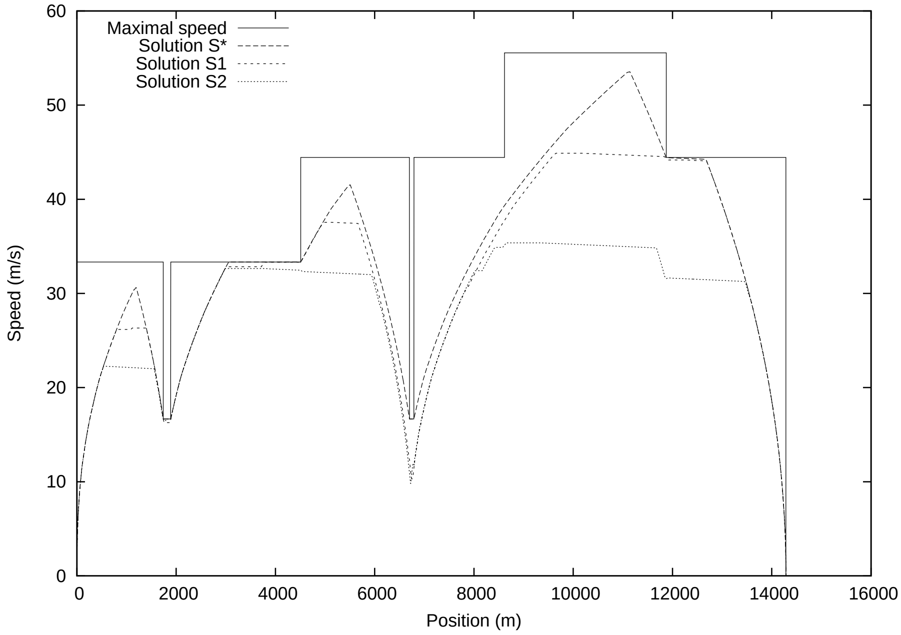


Figure 6: Example of speed tuning on several sections near Gonesse junction

By defining timetable constraints, the delays have to be reduced and this reduction can be seen as a third objective which can be integrated in the MOP. Hence the resolution adopted here can involve till three criteria: MOP= $(\varphi_1, \varphi_2, \varphi_3)$.

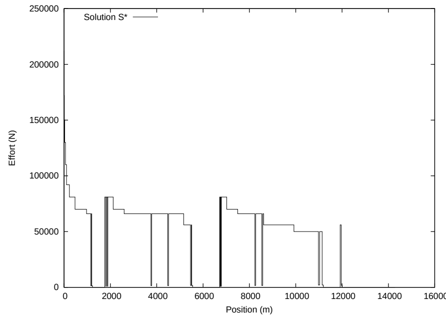
Performance assessment

Due to the different number of objectives, three cases of optimization are considered. In the first case, just the travel duration is minimized ($\Phi = (\varphi_1)$) for obtaining a reference solution as with the previous example. This solution, denoted A*, is the basis of comparison with the other solutions obtained in the other cases with two or three objectives, respectively denoted A_2^i and A_3^i.

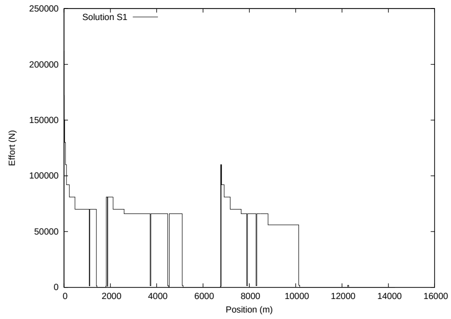
Table 3 presents the detailed results of the solutions proposed in each case. As we could assume, Solution A* proposes the best performance to minimize the travel duration, but also to minimize the delay. However, the energy consumption is the highest observed among the results as we can expect.

When adding at least one objective, the performance about the travel duration is slightly weaker than for A*. Indeed, with just a few seconds in more (less than 2%), the saved energy can reach till 50% of Solution A* consumption. Such a saving can be explained by the track slope which favours high cruising speeds without strong effort. That is why reducing the duration of some seconds needs a far bigger effort.

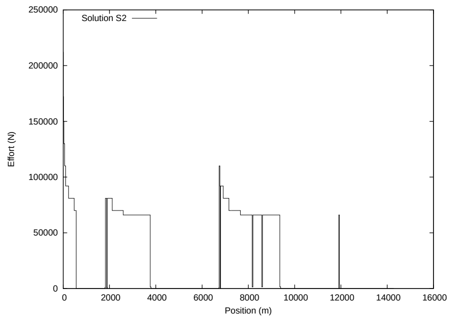
In the third case, when optimizing all the objectives, minimizing duration and delay may appear redundant. However, in this example, the delay occurs at station SC and not after. Indeed, the allowed duration for going from station SE to station RG seems to be a little bit too short, whereas the delay may be recovered between SC and RG. The allowed duration



(a) Effort during Solution S^*



(b) Effort during Solution S_1



(c) Effort during Solution S_2

Figure 7: Train effort according to its position for each solution: S^*, S_1, S_2

between these two latter stations is fully sufficient and such that the train can arrive in advance at the last station. Thus, it is possible to reduce the travel duration while increasing the delay. That is why it is relevant to distinguish travel duration and delay into two separate objectives to be optimized concurrently.

Therefore, the results provided by the last optimization (MOP= $(\varphi_1, \varphi_2, \varphi_3)$) tend to highlight the pressure on the solutions to optimize all the objectives. Indeed, all the solutions have durations less than 1% of time in more and the increase of occurred delays does not exceed 10%. This deviation of the delays is more satisfying than in the case of optimization with two objectives. That shows the relevance to add a third objective to limit the delays. Moreover, by analyzing the results we can observe that the delays occur at station SC (less than one minute) whereas the total travel durations indicate that the train can arrive in advance to its final destination. Even if the timetable constraints are almost respected, the arrival time at SC seems to be a little bit tight while it is slackened at station RG. We can assume the timetable was built for a previous rail engine with different technical data such that the timetable is not completely adapted to the new equipment.

Figures 10 and 11 illustrate the distribution of the solutions obtained in three cases (one, two or three objectives). Although the diversity is weak and the fronts are not spread, it is interesting to note that the combination of three objectives allows to find good tradeoff solutions. This fact is very relevant from an operational point of view whenever the decision makers search for a good compromise in order to satisfy both their own needs and the

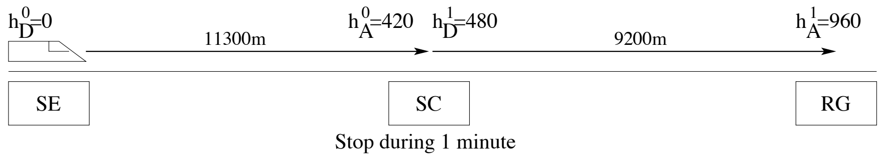


Figure 8: Description of the line Saint Étienne (SE) – Rive de Giers (RG) with one stop at Saint Chamond (SC) during one minute.

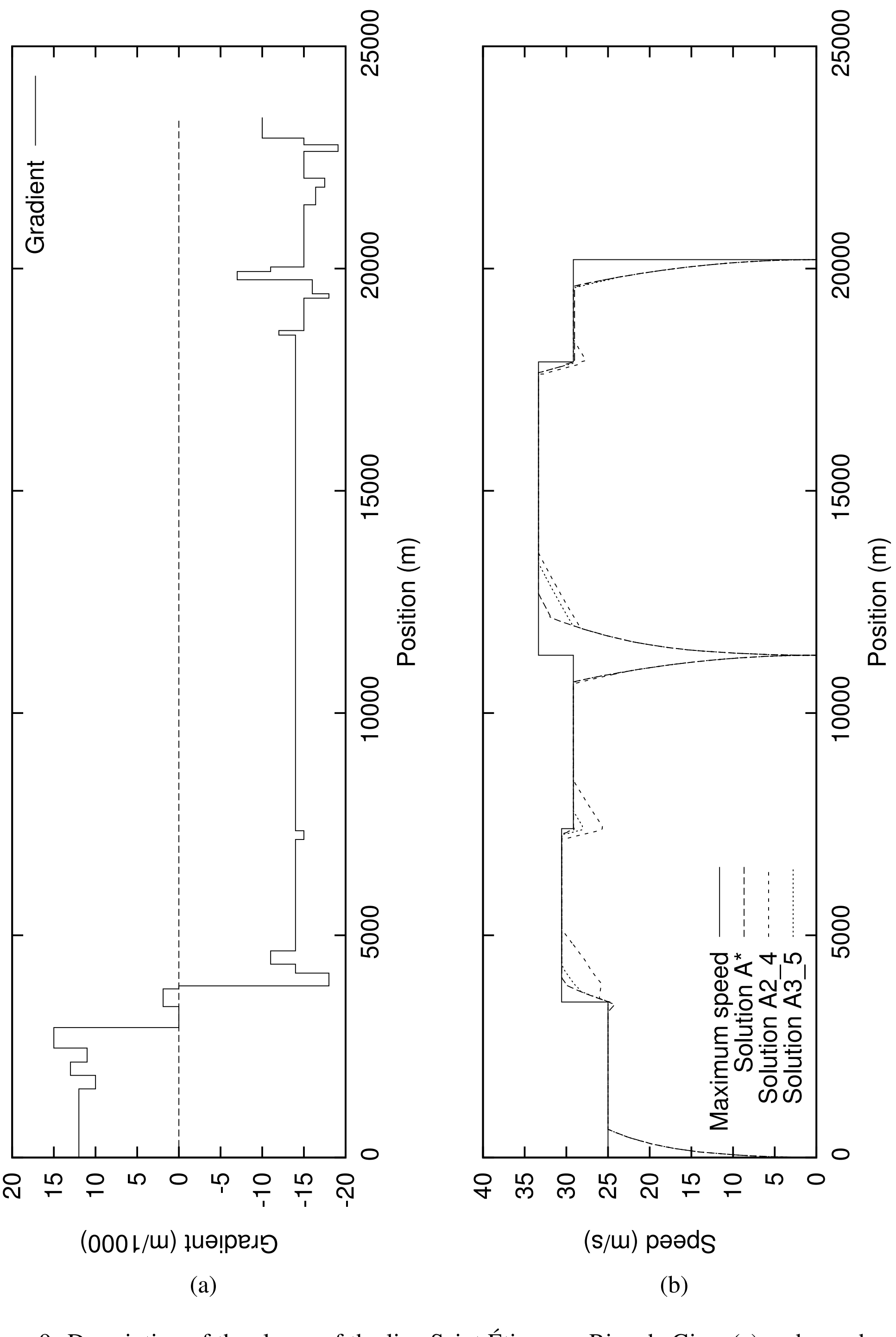
Sol.	E [kJ]	Deviation [%]	T [s]	Deviation [%]	D [s]	Deviation [%]
One objective, φ_1						
A*	216031		763		23	
Two objectives, MOP= (φ_1, φ_2)						
A_2^1	125604	-41.8	766	+0.3	25	+8.7
A_2^2	121051	-43.9	767	+0.5	25	+8.7
A_2^3	117970	-45.3	768	+0.65	25	+8.7
A_2^4	104456	-51.6	777	+1.8	32	+39.1
A_2^5	101486	-53.0	792	+3.8	47	+104.3
Three objectives, MOP= $(\varphi_1, \varphi_2, \varphi_3)$						
A_3^1	182732	-15.4	764	+0.13	23	+0.0
A_3^2	131592	-39.0	765	+0.26	24	+4.3
A_3^3	130094	-39.7	766	+0.3	24	+4.3
A_3^4	123959	-42.6	767	+0.5	24	+4.3
A_3^5	119518	-44.6	769	+0.78	24	+4.3
A_3^6	117975	-45.3	768	+0.65	25	+8.7

Table 3: Energy consumption, travel duration and delay of solutions obtained while optimizing one, two and three objectives during the travel on the line Saint Étienne – Rive de Giers

customers.

6 Conclusion and perspectives

In this paper we have dealt with a problem of speed tuning in railway management. The solving goal is to optimize both concurrent objectives: on the one hand by minimizing the travel duration and the delays, and on the other hand by reducing the energy consumption. To this end we have presented an evolutionary multi-objective approach for tuning speeds in order to minimize the durations while saving energy. This algorithm is based on IBEA and uses specific operators well-known in the literature for searching solutions in continuous space. The speed tuning is achieved by our method of speed profile building which introduces two intermediate speeds between the entrance and exit speeds of a section. This method paired with IBEA for searching in a continuous space brings its efficiency to the light. Furthermore, we have shown the interest of adding a third criterion about the reduction of the delays which is quite different to the reduction of the travel durations, even if



(a) (b)

Figure 9: Description of the slopes of the line Saint Étienne – Rive de Giers (a) and speed profiles obtained by optimization with one (A*), two (A_4^2) and three (A_5^3) objectives.

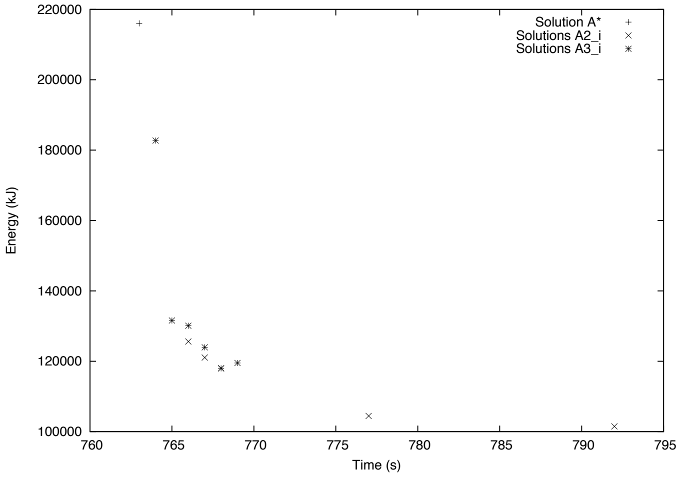


Figure 10: Distribution of the obtained solutions (Time / Energy)

there is a little redundancy.

However, the lack of diversity in the proposed solutions leads up to consider metaheuristics more adapted to the search of solutions in continuous space such as CMA-ES (Evolution Strategies with Covariance Matrix Adaptation [10]) and its multiobjective extension [11]. Moreover, in order to integrate this approach in train dispatching model it will be necessary to mix different kinds of variables. That is why our future works will focus on the development of mixed-variables model capable to both deal with continuous variables for defining speed profiles and discrete variables for scheduling the trains according to the timetable. Besides, in order to improve the robustness of the method and its efficiency, the future developments will take uncertainty [13] into account.

References

[1] Thomas Albrecht. *Railway Timetable & Traffic – Analysis, Modelling, Simulation*, chapter Energy-Efficient Operation Train Operation, pages 83–105. Eurail press, 2008.

[2] Thomas Albrecht and S. Oettich. *Computers in Railways VIII*, chapter A new integrated approach to dynamic schedule synchronization and energy saving train control, pages 847–856. WIT Press, Southampton, 2002.

[3] Olaf Brünger and Elias Dahlhaus. *Railway Timetable & Traffic – Analysis, Modelling, Simulation*, chapter Running Time Estimation, pages 58–82. Eurail press, 2008.

[4] C.S. 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, september 1999.

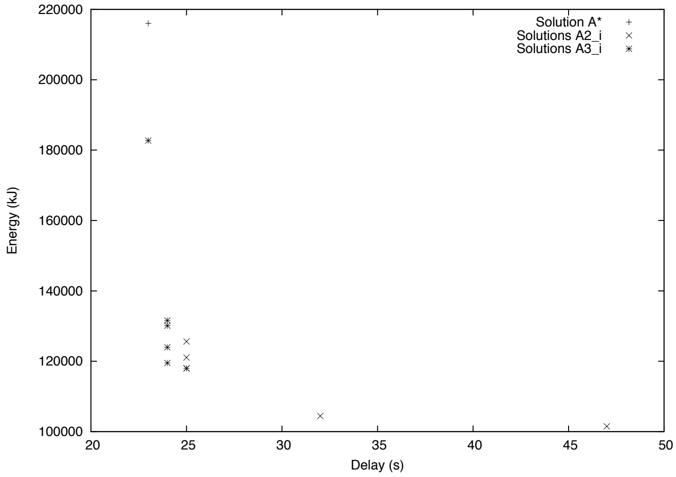


Figure 11: Distribution of the obtained solutions (Delay/Energy)

[5] Rémy 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*, pages 279–284, Madeira (Portugal), 2010.

[6] Kalyanmoy Deb. *Multi-objective Optimization using Evolutionary Algorithms*. Wiley, 2001. 517 p.

[7] Kalyanmoy Deb and Ram Bhusan Agrawal. Simulated binary crossover for continuous search space. Technical Report IITK/ME/SMD-94027, Indian Institute of Technology, Kanpur, 1995.

[8] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Parallel Problem Solving from Nature PPSN VI*, volume 1917/2000, pages 849–858, Berlin/Heidelberg, 2000. Springer.

[9] P.J. Fleming and R.C. Purshouse. Evolutionary algorithms in control systems engineering: A survey. *Control Engineering Practice*, 2002.

[10] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[11] Christian Igel, Nikolaus Hansen, and Stefan Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

[12] Simon Iwnicki, editor. *Handbook of Railway Vehicle Dynamics*. Taylor and Francis Group, 2006. 535 p.

[13] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments - a survey. *IEEE Transactions on Evolutionary Computation*, 9:303–317, 2005.

[14] Roger Kaller and Jean-Marc Allenbach. *Traction Électrique*, volume 2. Presses Polytechniques et Universitaires Romandes, 199. 2nd edition, 432 p.

[15] Daniel Lancien and Michèle Fontaine. Calculs de marches de train économisant l'énergie de traction – le programme MARECO. *Revue Générale des Chemins de Fer*, pages 679–692, november 1981.

[16] A. Liefooghe, L. Jourdan, and E.-G. Talbi. A unified model for evolutionary multiobjective optimization and its implementation in a general purpose software framework: ParadisEO-MOEO. Research Report RR-6906, INRIA, 2009.

[17] Rongfang Liu and Iakov Golovitcher. Energy-efficient train control. *Transportation Research part(A)*, 37:917–932, 2003.

[18] P. Lukaszewicz. Running resistance results and analysis of full-scale tests with passenger and freight trains in sweden. *Proc. IMechE, Part F: Journal of Rail and Rapid Transit*, 221 F2:183 – 192, 2007.

[19] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, (11):341 – 359, 1997.

[20] El Ghazali Talbi. *Metaheuristics: from design to implementation*. Wiley, 2009. 624 p.

[21] E. Zitzler, M. Laumanns, and L.Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.

[22] Eckart Zitzler and Simon Künzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242/2004, pages 832–842, Berlin/Heidelberg, 2004. Springer.