

A Greedy Heuristic for Optimizing Metro Regenerative Energy Usage

David Fournier, François Fages, Denis Mulard

► **To cite this version:**

David Fournier, François Fages, Denis Mulard. A Greedy Heuristic for Optimizing Metro Regenerative Energy Usage. Railways 2014, Apr 2014, Ajaccio, France. 10.4203/ccp.104.240 . hal-01103710

HAL Id: hal-01103710

<https://hal.archives-ouvertes.fr/hal-01103710>

Submitted on 15 Jan 2015

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.



Abstract

When the regenerative braking energy cannot be stored by the metro producing it, it has to be used instantaneously on the network, otherwise it is lost. In this case, the accelerating and braking trains need be synchronized to fully benefit from the regenerative energy, and a metro timetable is energetically optimized when all the regenerative braking is utilized to power other trains. This synchronization consists in lining up each braking train with an accelerating one in its neighbourhood. Doing so, the latter will benefit from the regenerative energy of the former. We propose a fast greedy heuristic to tackle the problem of minimizing the energy consumption of a metro timetable by modifying solely the dwell times in stations. This heuristic is compared to a state-of-the-art meta heuristic called the covariance matrix adaptation evolution strategy (CMA-ES) and shows similar results with much faster computation time. Finally, we show that a run of the algorithm on a full timetable may reduce its energy consumption by 5.1%.

Keywords: Energy Optimization, Regenerative Braking, Metro Timetabling

1 Introduction

Optimizing the use of the regenerative energy produced by trains when braking has become an essential feature in the process of reducing the energy consumption of metro networks. Greatbanks shows that potential energy savings provided by this technology are about 16.5% [1]. When the installation of super-capacitors or flywheels to store the regenerative energy turns out to be financially irrelevant, a purely software solution to reduce energy costs consists in optimizing the timetable.

However, metro timetables are primarily set to be consistent with quality of service, safety and other constraints like drivers' shifts or weekend periods, required by the

metro company. Energy considerations are classically not taken into account at first [2, 3]. This explains why it is hard to create a brand new timetable only to take into account energy issues. However, it is possible to slightly modify the current off-line timetables in order to minimize any objective function related to the energy consumption of the metro line. Beforehand, a valuation of the resulting energy consumed on every time slot is needed to make a proper optimization.

A timetable compiles a collection of trips – linked to a physical metro – defining the terminal departure time, the dwell times (stopping times in stations) and the speed profiles in interstations. Typically, these parameters are understood as space-time points, e.g. the metro X will reach the place Y at time Z . The literature deals with the modification of these three parameters to minimize a given objective function. Kim et al. [4, 5] modify only the departure times when Nasri et al. [6] modify the dwell times in stations, by choosing long or short dwell times. Other quantities can be modified, like reserve times in [7], but they can be reduced to a combination of the modification of speed profiles and dwell times. Two different objective function are minimized, namely the maximum power peak [8, 9] and the global energy consumption [10]. These functions can be computed with an accurate and time consuming electrical network simulator [6] or with a simple heuristic model [4].

In Section 2 we define the mathematical model of the metro timetable and we propose a classification of the metro timetabling energy optimization problems given in the literature. This classification is based on a triple summarizing three essential features of this class of problems: the objective function to optimize, the modifiable variables and the way to compute the energy consumption of the line. The problems of the literature are sorted based on this classification. Then in Section 3 we present a greedy heuristic to tackle the problem of minimizing the global energy consumption of a metro line by re-scheduling dwell times in stations. This algorithm takes advantage of the regenerative energy produced by braking metros along the line. It aims at synchronizing, for every braking metro, an accelerating one running in its neighbourhood. Every time a dwell time is shifted to better synchronize metros, the algorithm modifies the timetable accordingly and incrementally recomputes the global energy consumption. Eventually, the Section 4 summarizes the results of this heuristic compared with a state-of-the-art meta heuristic on 6 benchmark instances. The greedy heuristic shows competitive results in both computation time and quality of the solutions and is able to re schedule a full timetable to make it save 5.1% energy.

2 Classification of metro timetabling energy optimization problems

A metro line is crossed, along the day, by metros running from terminal to terminal in both directions, performing trips stopping in intermediate stations. The metro timetable compiles the collection of departure and arrival times of every trip at every station, from their departure terminal until their arrival terminal. Let us denote by \mathbf{T}

the set of trips crossing the metro line during a given time horizon, and by \mathbf{S} the set of metro stations on the line. The metro timetable consists of the data of the departure time $dep_{t,s}$ for each trip t and station s , and the arrival time $arr_{t,s}$ for t at s :

$$Timetable = \begin{cases} dep_{t,s} & \forall t \in \mathbf{T}, \forall s \in \mathbf{S} \\ arr_{t,s} & \forall t \in \mathbf{T}, \forall s \in \mathbf{S} \end{cases} \quad (1)$$

The metro timetabling energy optimization problem occurs when a metro company wants to minimize an objective function related to the energy consumption of its metro line by modifying its timetable. To date, there is no formal classification of energy optimization problems in metro timetabling. Xun et al. [11] proposed a classification of approaches focusing mainly on the methods used. In this section, a classification is proposed based on the mathematical definition of the problems, according to three main aspects in which the problems may differ, namely: the objective function, the decision variables and the energy diffusion model.

2.1 The objective function

In terms of energy consumption, two major factors are studied for minimization. The most studied objective function is the maximum power peak. The maximum power peak (PP) is the highest instant power demand of the timetable. A power peak occurs when too many metros are accelerating at the same time. As the metro company is paying fines if too many high power peaks occur during a time interval [12], reducing the height and the frequency of these power peak directly decreases the electricity bill.

Also, research is active in decreasing the global energy consumption (G) of a timetable. The advantage of minimizing this objective function is more obvious as using less electrical energy shall lead to pay less to the electricity provider.

2.2 The decision variables

The optimization of the objective function is done by modifying some variables in the timetable. Namely, three variables can be independently modified, changing the shape of the trips in the timetable and potentially reducing power peaks or the global consumption of the metro line:

- The departure times (D), or the startin time of a metro at its departure terminal. This is done by modifying $dep_{t,0}$ for every trip.
- The dwell times (d), or the stopping times in every station but the first one. The dwell times can be lengthened or shortened by changing the value $dep_{t,s}$ for all trips and stations except the terminal departure one.
- The speed profiles (s). It is common to have several speed profiles for a metro between two stations ; typically a nominal, a full speed and an economical one.

A speed profile modification directly changes the length of an interstation trip, modifying $arr_{t,s}$.

It is of course possible to combine different variables to optimize the objective function. For instance, Albrecht changes in [7] speed profiles and dwell times by using the reserve time of each trip.

2.3 The energy valuation

Both objective functions are computed by evaluating the energy consumption implicated by the metro runs. A metro running is effectively consuming energy, provided by electric sub stations placed along the line. Also, braking metros produce energy that can be used by other metros or that can be lost as heat.

Optimization processes can use an accurate energy valuation of a timetable (J), which require non linear electricity equations, applied on an equivalent electric network of the metro line. An accurate simulator computes exactly what are the transfers of energy between electric sub stations, accelerating metros and braking metros, evaluating the losses induced by Joule effects. This may requires a lot of computation time and is not well suited for optimization algorithms that make many objective function evaluations, like evolutionary algorithms.

That is why articles in the literature deal with more simple energy models (woJ) that allow them to use, for instance, mixed-integer linear programming. These models cannot aim at giving an accurate evaluation of the energy consumption of a metro line but can suffice to guide an optimization algorithm towards good solutions in the search space.

2.4 Problem classification

We propose to classify the energy optimization metro timetabling problems using a triple which indicates the choice done for the three previous features, namely: the objective function, the decision variables and the energy valuation. Table 1 shows the classification of different problems from the literature using these triples.

Triple = {

1. Objective function

- PP - Maximum power peak
- G - Global energy consumption ,

2. Decision variables

- D - Departure times
- d - Dwell times

- s - Speed profiles ,

3. Energy valuation

- J Joule effects consideration
- woJ Simple energy model

}

Problem Class	References
$\{PP, d, J\}$	Chen et al. [8]
$\{PP, d, woJ\}$	Sanso et al. [9]
$\{PP, d-s, woJ\}$	Albrecht et al. [7]
$\{PP, D, woJ\}$	Kim et al. [4, 5]
$\{G, d, J\}$	Chang et al. [13], Fournier et al. [14], Nasri et al. [6]
$\{G, D-d, J\}$	Peña et al. [10]

Table 1: Classification of some metro timetabling energy optimization problems from the literature.

3 Greedy heuristic algorithm for (G, d, J)

The literature tackles the metro timetabling energy optimization problems taking advantage of mixed-integer linear programming techniques [4, 5, 10] or using meta heuristics like genetic algorithms [6, 7, 8]. Both ways are *global* methods and have the advantage to let aside the problem of knowing where to find, in the search space, high quality solutions. These techniques are guided to good solutions by different mechanisms which are valid for any kind of problem.

The drawback is that they do not take advantage of natural paths leading to the best solutions that can appear when looking closely at how these solutions look like. Likewise, using these global techniques may lead to scalability problems [6] when the search space starts growing to the size of industrial problems.

We propose a fast greedy heuristic, scalable to an entire timetable, that minimizes the energy consumption by lengthening or shortening dwell times of metros in stations. It tackles problems of more than 10000 variables and is based on the idea of increasing the synchronization of braking metros with accelerating ones in their neighbourhood.

3.1 Braking and acceleration phases

The metro interstation runs can be divided into three phases. The first one is the acceleration phase, when the metro leaves its departure station and accelerates at full

power. Then, the metro cruises, using a small amount of power to keep its speed constant, or coasts, using its inertia to keep moving without powering. Finally, the metro enters a braking phase to stop at its arrival station.

The braking phase allows the metro to regenerate energy to the third rail or the catenaries. If during the braking phase of a metro, another metro is in its acceleration phase, then a transfer of energy is possible between them. The accelerating metro will demand less power to the electric sub stations and the global energy consumption is thus reduced. Better synchronizing braking and acceleration phases between metros running in the same time on the line is thus a way to optimize the use of regenerative energy, minimizing in the same time the global energy consumption.

\mathbb{B} is the set of braking phases and \mathbb{A} is the set of acceleration phases.

Definition 1. *The braking phase $B_{t,s} \in \mathbb{B}$ is the interval of time where the trip $t \in \mathbf{T}$ is braking, in order to dwell in station $s \in \mathbf{S}$. It is characterized by two parameters: its initial start time $start(B_{t,s})$ and its duration length($B_{t,s}$) ; and by one variable: its cumulated shift $shift(B_{t,s})$.*

Definition 2. *The accelerating phase $A_{t,s} \in \mathbb{A}$ is the interval of time where the trip $t \in \mathbf{T}$ is accelerating, in order to leave the station $s \in \mathbf{S}$. It is characterized by two parameters: its initial start time $start(A_{t,s})$ and its duration length($A_{t,s}$) ; and one variable: its cumulated shift $shift(A_{t,s})$.*

It is worth noticing that every acceleration phase occurs right after a dwell time. Shifting the starting time of an acceleration phase in the timetable is thus equivalent to modify the length of the adjacent dwell time.

3.2 Tolerances on the optimization

The optimization algorithm aims at minimizing the global energy consumption by the sole modification of the dwell times of an initial timetable. Each dwell time can be modified within an interval defined by a tolerance given by the metro company, typically of a few seconds. Using it, the algorithm is able to reduce or increase the duration of every dwell time, increasing the synchronization between braking and accelerating metros.

Let us denote $\delta_{t,s} \in \mathbb{Z}$ the shift applied to the acceleration phase $A_{t,s}$ in order to optimize the global energy consumption. The minimum and maximum shifts that can be applied to $A_{t,s}$ are denoted $\underline{\delta}_{t,s} \in \mathbb{Z}^-$ and $\overline{\delta}_{t,s} \in \mathbb{N}$. The shifts applied on several dwell times of the same trip cannot reduce or increase the total duration of the trip – from terminal to terminal – by too much. This is due to the fact that headways between trips that are following each other have to be fairly constant. Another reason is the fact that the commercial speed – the time a metro runs its entire trip – is often contractually set. A trip duration cannot be reduced by more than $\underline{\Delta}_t$ and cannot be increased by more than $\overline{\Delta}_t$:

$$\underline{\Delta}_t \leq \sum_{s \in \mathbf{S}} \delta_{t,s} \leq \overline{\Delta}_t \quad t \in \mathbf{T} \quad (2)$$

Shifting an acceleration phase modifies its own cumulative shift, but also the cumulated shifts of all the phases of the same trip occurring after in the timetable. The cumulated shift of a phase of a given trip is thus the sum of the deltas applied on this trip in previous stations:

$$shift(A_{t,s}) = \sum_{u < s} \delta_{t,u} + \delta_{t,s} \quad t \in \mathbf{T}, s \in \mathbf{S} \quad (3)$$

$$shift(B_{t,s}) = \sum_{u < s} \delta_{t,u} \quad t \in \mathbf{T}, s \in \mathbf{S} \quad (4)$$

3.3 Neighbourhood of a braking phase

The algorithm is shifting, for each braking phase, one neighbour acceleration phase. Doing so, the use of the regenerative energy is maximized, reducing the global energy consumption. To facilitate the understanding of the equations, let us denote the starting time of a phase modified by the cumulative shifts

$$start^*(A_{t,s}) = start(A_{t,s}) + shift(A_{t,s}) \quad (5)$$

and the ending time of a phase modified by the cumulative shifts

$$end^*(A_{t,s}) = start(A_{t,s}) + length(A_{t,s}) + shift(A_{t,s}) \quad (6)$$

The neighbourhood of a braking phase $\mathcal{N}(B_{t,s})$ is defined as the set of acceleration phases that can overlap it within the given tolerances. This means that every acceleration phase that may start before the end and finish after the beginning of a given braking phase, belongs to its neighbourhood:

$$\mathcal{N}(B_{w,v}) = \{A_{t,s} \in \mathbb{A} \mid end^*(A_{t,s}) + \overline{\delta}_{t,s} > start^*(B_{w,v}) \quad (7)$$

$$\wedge start^*(A_{t,s}) + \underline{\delta}_{t,s} < end^*(B_{w,v})\} \quad w \in \mathbf{T}, v \in \mathbf{S} \quad (8)$$

3.4 Objective function

The algorithm evaluates the resulting energy consumption of solution timetables by the means of an electrical consumption simulator. Because of non trivial electricity equations, the simulator is only capable to compute the *instant power demand* during a small period of time. It is indeed possible to assume that, if the time duration is small enough, metros running on the line produce or consume a constant power and stay at a constant position.

The timetable is cut in small periods of time, called time slots, and the *instant power demand* is a function of the time slot, the acceleration and braking phases and a distribution matrix which summarizes the potential transfers between metros as a function of their position on the line. The objective function of the timetable is thus the sum, over the time slots τ , $0 \leq \tau \leq \mathcal{T}$, of the instant power demands:

$$f_{obj} = \sum_{\tau=0}^{\tau=\mathcal{T}} f(\mathbb{A}_\tau, \mathbb{B}_\tau, DMat) \quad (9)$$

\mathbb{A}_τ and \mathbb{B}_τ being the *relevant* acceleration and braking phases of the time slot τ .

Definition 3. The relevant braking phases \mathbb{B}_τ and acceleration phases \mathbb{A}_τ of time slot τ are the subset of \mathbb{B} and \mathbb{A} of phases that are occurring at time slot τ :

$$\mathbb{A}_\tau = \{A_{t,s} \in \mathbb{A} \mid \text{start}^*(A_{t,s}) \leq \tau \leq \text{end}^*(A_{t,s})\} \quad (10)$$

$$\mathbb{B}_\tau = \{B_{t,s} \in \mathbb{B} \mid \text{start}^*(B_{t,s}) \leq \tau \leq \text{end}^*(B_{t,s})\} \quad (11)$$

3.4.1 Distribution matrix

The distribution matrix $DMat$ is the matrix of size $|\mathbf{S}| \times |\mathbf{S}|$ which compiles the rate of regenerative energy one metro at a given position on the line is capable to transfer to another metro accelerating somewhere else on the line. This transfer rate is computed thanks to an electrical network simulator.

The simulator embeds the entire electrical topology of the line, like the position of the electric sub stations and the equivalent resistance of the catenaries. It takes into account Joule effects occurring in cables as well as voltage peak limitations of the network. Figure 1 illustrates a network where a metro is braking, arriving in station 2, and is regenerating energy that the metro accelerating from station 5 could benefit from.

The distribution rate $DMat_{2,5}$ is computed by calculating the resulting energy consumption of the metro line ratio between the situation of a metro accelerating alone in station 5 and the same metro benefiting of the regenerative energy of the metro braking in station 2. If we denote the power demand of the network when the regenerative braking is used P_{reg} , the distribution rate is equal to:

$$DMat_{2,5} = \frac{P^+ - P_{reg}}{P^-} \quad (12)$$

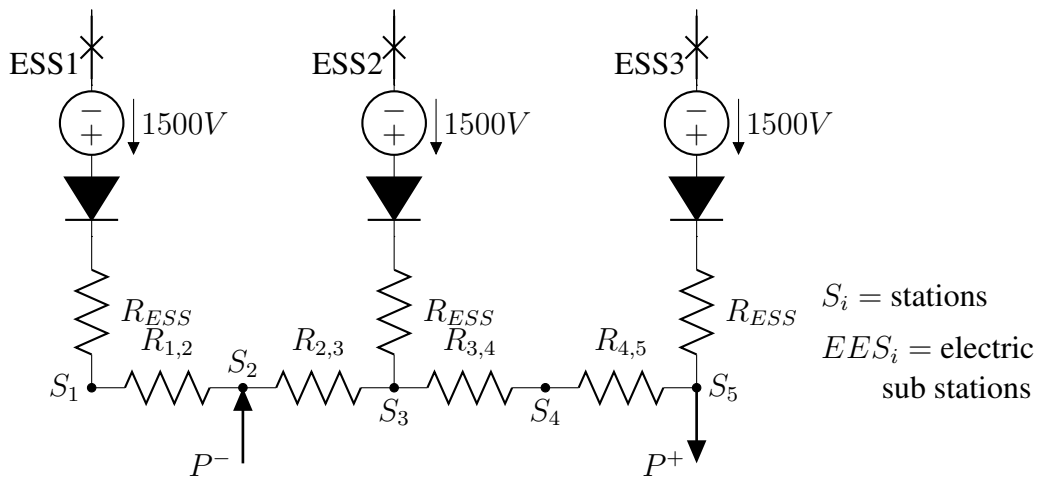


Figure 1: Equivalent electrical network of the metro line used to compute the distribution rate from station 2 to station 5.

3.4.2 Computation of the instant power demand

The computation of the instant power demand $f(\mathbb{A}_\tau, \mathbb{B}_\tau, DMat)$ is based on a heuristic that aims at maximizing the energy transfers between braking and accelerating metros. A braking metro has the capacity to transfer its energy to another metro, accelerating at the same time on the line. The heuristic leads this metro to transfer its regenerative energy to the metro which has the best transfer rate, in order to minimize the losses on the line.

Let us denote $power(A_{t,s})$ the power demanded by the trip t accelerating from station s and $power(B_{t,s})$ the power supplied by the trip t braking to station s . These powers are deduced thanks to the energy profiles of each trip provided by the metro company. For a given time slot τ , the *metros power demand* represents the cumulated power demanded by accelerating trains on the line:

$$\text{metros power demand} = \sum_{A_{t,s} \in \mathbb{A}_\tau} power(A_{t,s}) \quad (13)$$

Likewise, the *energy saving* at time slot τ is the powers supplied by braking metros, attenuated by a factor determined in the distribution matrix.

In an electrical network, Joule effects are naturally minimized. The transfers of regenerative energy are also following this rule. So, each braking metro t transfers its regenerative energy to an accelerating metro w on the line such that the distribution factor $DMat_{t,w}$ is maximized. Eventually, the instant power demand is equal to the metros power demand minus the energy saving. The following pseudo-code describes in detail the computation of the instant power demand at a time slot τ .

Algorithm 1 Computation of the instant power demand at a given time slot τ

Require: $\mathbb{B}_\tau, \mathbb{A}_\tau, DMat$

```
1: metros power demand =  $\sum_{A_{t,s} \in \mathbb{A}_\tau} power(A_{t,s})$ 
2: energy saving = 0
3: while  $\mathbb{B}_\tau \neq \emptyset$  do
4:   Choose randomly  $B_{t,s} \in \mathbb{B}_\tau$ 
5:    $p_{brake} \leftarrow power(B_{t,s})$ 
6:   while  $p_{brake} > 0$  do
7:     if  $\mathbb{A}_\tau \neq \emptyset$  then
8:       Choose  $A_{v,w} \in \mathbb{A}_\tau$  such that  $DMat_{s,w}$  is maximized
9:        $p_{accel} \leftarrow power(A_{v,w})$ 
10:      if  $p_{brake} \cdot DMat_{s,w} > p_{accel}$  then
11:         $p_{brake} \leftarrow p_{brake} - p_{accel} \cdot DMat_{s,w}$ 
12:         $\mathbb{A}_\tau \leftarrow \mathbb{A}_\tau \setminus \{A_{v,w}\}$ 
13:        energy saving  $\leftarrow$  energy saving +  $p_{accel}$ 
14:      else
15:         $p_{accel} \leftarrow p_{accel} - p_{brake} \cdot DMat_{s,w}$ 
16:         $p_{brake} \leftarrow 0$ 
17:        energy saving  $\leftarrow$  energy saving +  $p_{brake} \cdot DMat_{s,w}$ 
18:       $\mathbb{B}_\tau \leftarrow \mathbb{B}_\tau \setminus B_{t,s}$ 
19: return instant power demand = metros power demand – energy saving
```

3.5 Greedy heuristic optimization algorithm

The idea of the greedy heuristic is to pair every braking phase of the timetable with an acceleration phase in its neighbourhood. This is done by shifting each acceleration phase to best synchronize it with its neighbour braking phase and by recomputing the objective function to check for improvement. The shift that should be applied on an acceleration phase is computed to bring its starting time closest to the braking phase's starting time, fulfilling the constraints on shifting tolerances as follows:

if $start^*(B_{v,w}) > start^*(A_{t,s})$

$$\delta_{t,s} = \min\left(start^*(B_{v,w}) - start^*(A_{t,s}), \overline{\delta}_{t,s}, \overline{\Delta}_t - shift(A_{t,s})\right) \quad (14)$$

if $start^*(B_{v,w}) < start^*(A_{t,s})$

$$\delta_{t,s} = \max\left(start^*(B_{v,w}) - start^*(A_{t,s}), \underline{\delta}_{t,s}, \underline{\Delta}_t - shift(A_{t,s})\right) \quad (15)$$

A new synchronization only shifts braking and acceleration phases that occur *after* in the timetable as shown in equations (3) and (4). The algorithm sweeps the timetable chronologically by synchronizing first the braking phases occurring at the beginning of the timetable. Doing so, when a braking phase has been synchronized with its best

acceleration interval, it cannot be undone due to another synchronization as this one will inevitably be done later than the first one.

The algorithm is greedy because once an acceleration phase has been shifted, it is removed from the pool of phases and cannot be shifted any more for another braking phase. From an industrial point of view, the greedy algorithm has the advantage of giving the same output given the same input, contrary to genetic algorithms. It is also monotonic and only minimizes the objective function. If, for a given braking phase, no acceleration phase synchronization can improve the objective function, then the algorithm does not shift any acceleration phase. It is a useful feature for real-time optimization, as the user is certain not to worsen its current solution. It is then possible to run automatically the greedy algorithm on a real-time instance and let it enhance the current solution as it gets better results.

Algorithm 2 Synchronization of acceleration phases with braking phases

Require: $\mathbb{A}, \mathbb{B}, DMat$

- 1: $\mathbb{A}^{INIT} \leftarrow \mathbb{A}, \mathbb{B}^{INIT} \leftarrow \mathbb{B}$
 - 2: Sort $B_{t,s} \in \mathbb{B}$ by $start^*(B_{t,s})$ in ascending order
 - 3: **for all** $B_{t,s} \in \mathbb{B}$ **do**
 - 4: Compute initial objective function f_{obj}^{INIT}
 - 5: Initialize best objective function $f_{obj}^{BEST} = f_{obj}^{INIT}$
 - 6: Initialize best shift $\delta^{BEST} = 0$
 - 7: Compute set of neighbours acceleration phases $\mathcal{N}(B_{t,s})$
 - 8: **for all** $A_{v,w} \in \mathcal{N}(B_{t,s})$ **do**
 - 9: Compute the shift $\delta_{v,w}$ to apply to $(A_{v,w})$
 - 10: Incremental objective function $f_{obj}^{INC} = f(f_{obj}^{INIT}, \mathbb{A}^{INIT}, \mathbb{B}^{INIT}, \delta_{v,w})$
 - 11: **if** $f_{obj}^{INC} < f_{obj}^{BEST}$ **then**
 - 12: $f_{obj}^{BEST} \leftarrow f_{obj}^{INC}$
 - 13: $\delta^{BEST} \leftarrow \delta_{v,w}$
 - 14: Apply δ^{BEST} on the timetable
 - 15: **finish**
-

3.5.1 Incremental computation of the objective function

The equation (9) shows that the instant power demand of the time slot τ is a function of the relevant braking and acceleration phases and of the distribution matrix. As the latter is constant given an electrical topology, only the relevant phases are modified within the optimization process.

During the optimization process, only one acceleration phase is shifted at a time before re-evaluating the objective function. As only one dwell time is modified, many time slots comprise the exact same relevant braking and acceleration phases between two objective function computations, keeping the instant power demand constant on these. The equation

$$f_{obj}^{INC} = f(f_{obj}, \mathbb{A}, \mathbb{B}, \delta_{t,s}) \quad (16)$$

shows that the *incremental* objective function f_{obj}^{INC} is a function of the initial objective function f_{obj} , the braking and acceleration phases, and a shift $\delta_{t,s}$ applied on $A_{t,s} \in \mathbb{A}$ such that:

$$shift(A_{t,w}) \leftarrow shift(A_{t,w}) + \delta_{t,s} \quad t \in \mathbf{T}, \forall w \geq s \in \mathbf{S} \quad (17)$$

$$shift(B_{t,w}) \leftarrow shift(B_{t,w}) + \delta_{t,s} \quad t \in \mathbf{T}, \forall w > s \in \mathbf{S} \quad (18)$$

The evaluation of f_{obj}^{INC} consists in recomputing the sole instant power demands on time slots where the relevant phases are modified by the shift. It avoids recomputing known values, increasing by an order of magnitude the computation time of the algorithm.

3.5.2 Restarts

The algorithm sweeps the timetable by synchronizing the braking phases taken in chronological order. Since the sweep does not go backwards, once a braking phase has been synchronized with its best acceleration neighbour, it can not be modified anymore. Once the sweep is done however, the output optimized timetable can be utilized as an input timetable and it is possible to re-run the optimization process, based on this new solution.

The optimization algorithm can be thus done once, or with restarts until convergence of the objective function. The algorithm with restarts stops when a sweep does not improve the objective function anymore.

Figure 2 shows the minimization of the objective function on a benchmark instance over time, using restarts. It shows that, even if the first run is the one that optimizes most of the timetable, re-runs allow significant improvement, increasing also the robustness of the final solution.

4 Results

4.1 Comparison with the CMA-ES evolutionary algorithm

To validate our optimization algorithm, we shall compare it with a state-of-the-art global search algorithm. We compare our greedy heuristic with a powerful evolutionary algorithm, namely the covariance matrix adaptation evolution strategy (CMA-ES) of N. Hansen [15]. CMA-ES is a meta-heuristic for solving optimization problems given with any arbitrarily complex objective function. It computes at each step a population of solutions, randomly generated through some multi-variate normal distribution. At each step, the distribution is updated by moving the mean in the most promising direction, and by adapting the standard deviation through an estimation of the covariance matrix as illustrated in Figure 3.

Due to the lack of benchmark instances in the literature, we propose a set of 6 instances to compare the different methods. These instances are drawn from real data

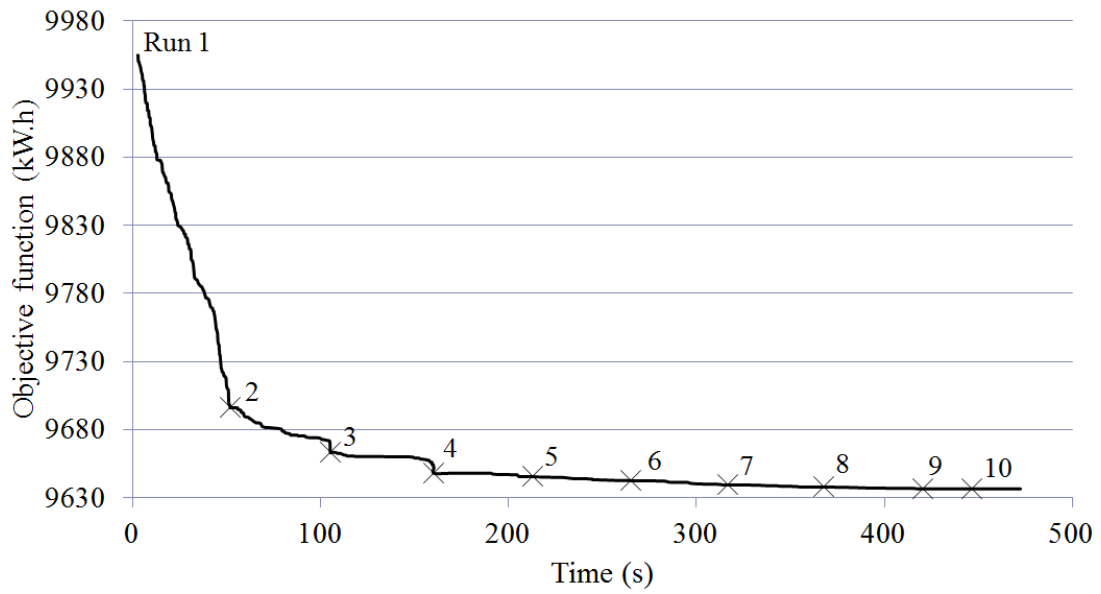


Figure 2: Evolution of the objective function on a benchmark instance during the optimization process with restarts.

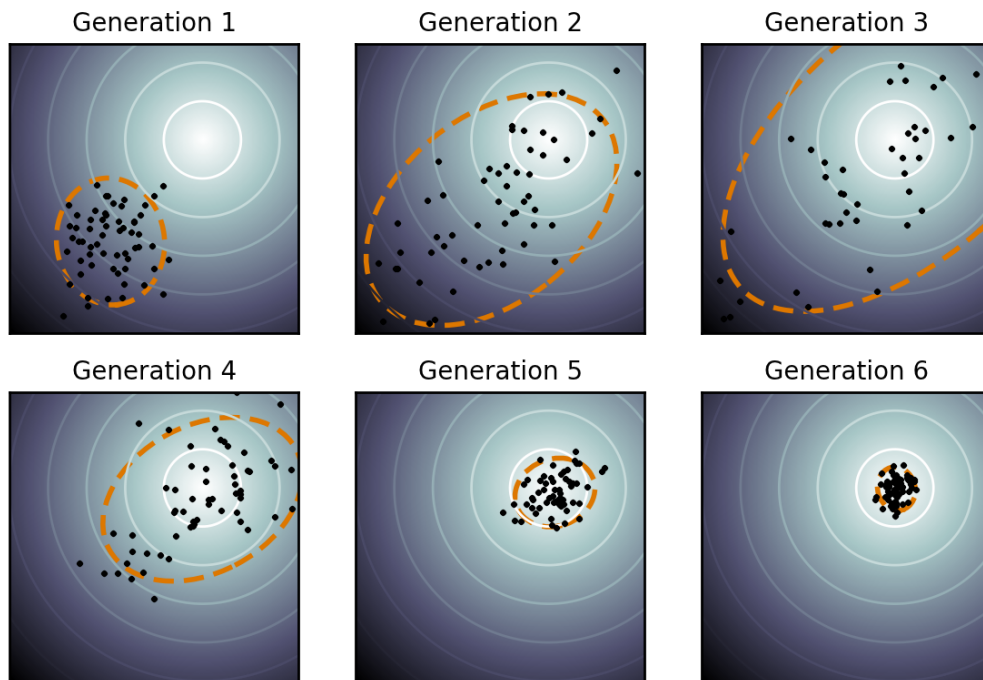


Figure 3: Modification of the multi-variate normal distribution of the population used at each step of the CMA-ES optimization process in a simple quadratic optimization problem (source Wikipedia).

and represent relevant portions of the timetable, i.e. peak (p) and off-peak (op) parts of size of 15 minutes and one hour. The tolerances on shifts are constant for all 6 instances as follows:

$$\underline{\delta}_{t,s} = -3 \quad \forall t \in \mathbf{T}, \forall s \in \mathbf{S} \quad (19)$$

$$\overline{\delta}_{t,s} = 9 \quad \forall t \in \mathbf{T}, \forall s \in \mathbf{S} \quad (20)$$

$$\underline{\Delta}_t = -30 \quad \forall t \in \mathbf{T} \quad (21)$$

$$\overline{\Delta}_t = 30 \quad \forall t \in \mathbf{T} \quad (22)$$

Table 2 summarizes the energy savings and the computation time on each of the 6 benchmark instances. CMA-ES is compared with the one-run heuristic and with the heuristic restarting until convergence. The heuristic with restarts gives the best results on 4 of 6 instances and is faster than CMA-ES on these. The one-run heuristic gives similar results than CMA-ES in terms of energy savings, gaining one order of magnitude in terms of computation time.

Instance	A	Initial Consumption (kW.h)	Optimized Consumption (kW.h)			
			<i>Computation Time (s)</i>			
			CMA-ES	Heuristic without restart	Heuristic with restarts	MILP
op1 (15 min)	127	2514	2401 (-4.51%) 256	2411 (-4.10%) 7.31	2394 (-4.77%) 45.6	2433 (-3.22%) 0.73
op2 (15 min)	129	2516	2402 (-4.54%) 223	2403 (-4.50%) 7.51	2376 (-5.56%) 38.0	2421 (-3.78%) 1.14
op3 (60 min)	449	9956	9719 (-2.38%) 782	9642 (-3.15%) 52.5	9556 (-4.02%) 648	9555 (-4.03%) 1500 (10.62%)
p1 (15 min)	173	3433	3296 (-4.00%) 503	3293 (-4.08%) 18.3	3262 (-4.98%) 178	3308 (-3.64%) 207
p2 (15 min)	186	3651	3516 (-3.37%) 669	3505 (-4.00%) 25.7	3442 (-5.72%) 291	3496 (-4.25%) 95
p3 (60 min)	670	13067	12696 (-2.84%) 1030	12801 (-2.04%) 168	12713 (-2.71%) 1500	12725 (-2.62%) 1500 (20.44%)

Table 2: Energy saved by three different optimization methods on six benchmark instances.

4.2 Full timetable optimization

The heuristic has been applied to a full timetable used for weekdays revenue service on a major metro line. As the optimized timetable shall be utilized in a real context,

Instance	Initial Consumption (kW.h) <i>Overlap time (s)</i>	Optimized Consumption (kW.h) <i>MILP objective: Overlap time (s)</i>		
		Heuristic without restart	Heuristic with restarts	MILP
op1 (15 min)	2514 12.48	2411 (-4.10%) 105.6	2394 (-4.77%) 101.8	2433 (-3.22%) 318.1*
op2 (15 min)	2516 11.48	2403 (-4.50%) 105.8	2376 (-5.56%) 159.1	2421 (-3.78%) 351.6*
op3 (60 min)	9956 45.97	9642 (-3.15%) 729.5	9556 (-4.02%) 817.9	9555 (-4.03%) 1637 (10.62%)
p1 (15 min)	3433 250.5	3293 (-4.08%) 374.6	3262 (-4.98%) 414.8	3308 (-3.64%) 772.5*
p2 (15 min)	3651 279.1	3505 (-4.00%) 510.7	3442 (-5.72%) 533.8	3496 (-4.25%) 835.8*
p3 (60 min)	13067 1019	12801 (-2.04%) 1423	12713 (-2.71%) 1576	12725 (-2.62%) 3003 (20.44%)

Table 3: For MILP results: * means optimal value ; (X.XX%) is the bounds gap ratio.

the constraints on shifts have been tightened to:

$$\underline{\delta}_{t,s} = -3 \quad \forall t \in \mathbf{T}, \forall s \in \mathbf{S} \quad (23)$$

$$\overline{\delta}_{t,s} = 3 \quad \forall t \in \mathbf{T}, \forall s \in \mathbf{S} \quad (24)$$

$$\underline{\Delta}_t = -15 \quad \forall t \in \mathbf{T} \quad (25)$$

$$\overline{\Delta}_t = 15 \quad \forall t \in \mathbf{T} \quad (26)$$

The metro line comprises 31 stations and is crossed by 694 metros from 6am to 1am. The size of the problem is of 9585 variables to optimize and is intractable by CMA-ES. The use of the incremental computation of the objective function and the algorithm strategy allows the greedy heuristic to find very competitive solutions faster than global methods. The heuristic sweeps the timetable in 20 minutes. The figure 4 compares the energy consumption of the initial timetable and the optimized one. The initial timetable consumes 194.0 MW.h and the optimized timetable consumes 184.0 MW.h, saving 10 MW.h and representing 5.1% of the global energy consumption. It appears that more energy is saved during peak hours. This is due to the fact that, according to the distribution matrix, the energy transfers can be done only between metros that are very close from each other. This case would appear more often during peak hours, when the density of metros on the line is higher. The extrapolation of these savings could save to the metro company 3.65 GW.h of electrical energy per year.

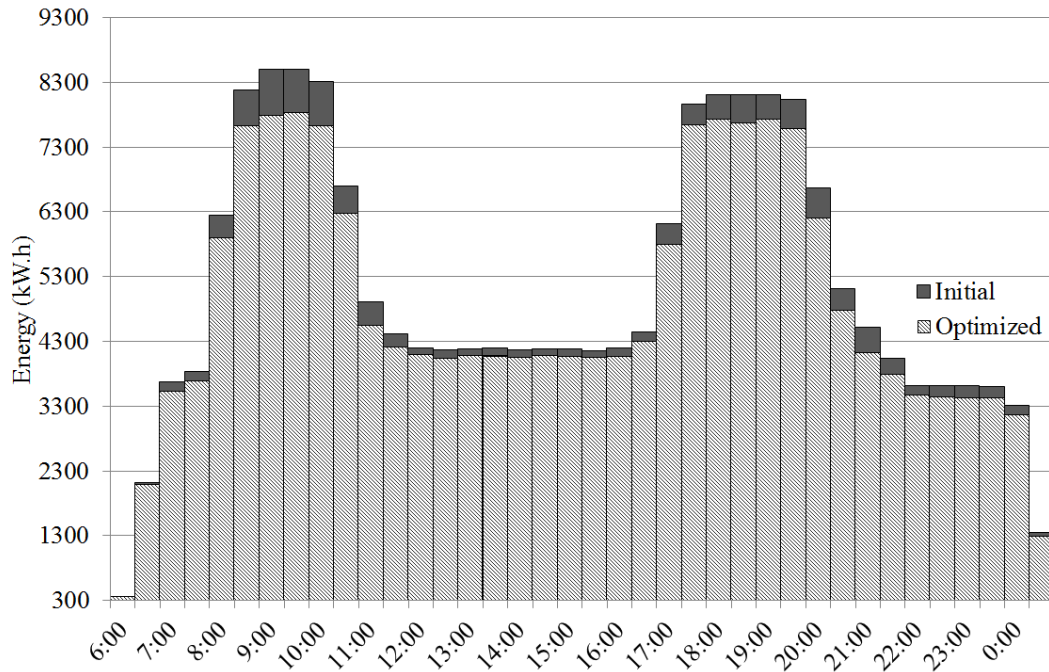


Figure 4: Compared energy consumption by intervals of 30 minutes of a full weekday timetable between the initial solution and the optimized solution.

5 Conclusion

We have proposed a classification of metro timetabling energy optimization problems based on their mathematical definition and have sorted the problems of the literature based on this classification.

We have then proposed a greedy heuristic to tackle optimize the energy consumption of a metro line by shifting the acceleration phases of metros to synchronize them better with braking phases of other metros. This algorithm finds solutions that are competitive with state-of-the-art meta heuristics on 6 benchmark instances, improving the computation time by an order of magnitude.

The results simulated on a full timetable show savings of the total energy consumption by 5.1%. This optimized timetable remains to be validated by running it in real conditions. The fast computation by the heuristics makes it possible to contemplate the re-optimization of a timetable in real-time after minor perturbations.

References

- [1] J. Greatbanks, "Review of the discount for using regenerative braking," *AEA Technology*, 2005.
- [2] E. Bampas, G. Kaouri, M. Lampis, and A. Pagourtzis, "Periodic metro scheduling," *ATMOS 2006*, 2006.

- [3] A. Caprara, M. Fischetti, and P. Toth, “Modeling and solving the train timetabling problem,” *Operations Research*, vol. 50, pp. 851–861, September/October 2002.
- [4] K. M. Kim, S.-M. Oh, and M. Han, “A mathematical approach for reducing the maximum traction energy: The case of korean MRT trains,” *IMECS 2010*, Mar. 2010.
- [5] K. Kim, K. Kim, and M. Han, “A model and approaches for synchronized energy saving in timetabling,” *WCRR 2011*, May 2011.
- [6] A. Nasri, M. F. Moghadam, and H. Mokhtari, “Timetable optimization for maximum usage of regenerative energy of braking in electrical railway systems,” *SPEEDAM 2010*, 2010.
- [7] T. Albrecht, “Reducing power peaks and energy consumption in rail transit systems by simultaneous metro running time control,” *Computers in Railways IX*, 2004.
- [8] J.-F. Chen, R.-L. Lin, and Y.-C. Liu, “Optimization of an MRT train schedule: Reducing maximum traction power by using genetic algorithms,” *IEEE Transactions on power systems*, vol. 20, pp. 1366–1372, Aug. 2005.
- [9] B. Sanso and P. Girard, “Trains scheduling desynchronization and power peak optimization in a subway system,” *IEEE*, 1995.
- [10] M. Peña, A. Fernández, A. P. Cucala, A. Ramos, and R. Pecharromán, “Optimal underground timetable design based on power flow for maximizing the use of regenerative-braking energy,” *Journal of Rail and Rapid Transit*, vol. 226, pp. 397–408, July 2012.
- [11] J. Xun, X. Yang, B. Ning, T. Tang, and W. Wang, “Coordinated train control in a fully automatic operation system for reducing energy consumption,” *Computers in Railways XIII*, pp. 3–13, 2012.
- [12] I. A. of Public Transport, “Reducing energy consumption in underground systems – an important contribution to protecting the environment.” 52nd International Congress of UITP, 1997.
- [13] C. Chang, Y. Phoa, W. Wang, and B. Thia, “Economy/regularity fuzzy-logic control of DC railway systems using event-driven approach,” *IEE Proceedings of Electric Power Applications*, vol. 143, Jan. 1996.
- [14] D. Fournier, D. Mulard, and F. Fages, “Optimisation énergétique de tables horaires de métros: une approche hybride,” in *Actes de Journées Francophones de Programmation par Contraintes JFPC’12*, (Toulouse, France), pp. 127–130, 2012.

- [15] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.