



HAL
open science

Piecewise linear bounding of energy conversion functions and resulting MILP-based solution methods

Sandra Ulrich Ngueveu

► **To cite this version:**

Sandra Ulrich Ngueveu. Piecewise linear bounding of energy conversion functions and resulting MILP-based solution methods. [Research Report] 16358, INP Toulouse; LAAS-CNRS. 2016. hal-01444317v2

HAL Id: hal-01444317

<https://hal.science/hal-01444317v2>

Submitted on 19 Oct 2017 (v2), last revised 30 May 2018 (v3)

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.

Piecewise linear bounding of energy conversion functions and resulting MILP-based solution methods

Sandra Ulrich Ngueveu

LAAS-CNRS, Université de Toulouse, CNRS, INP, F-31400 Toulouse, France
ngueveu@laas.fr

02/10/2017

Technical report LAAS-CNRS n° 16358
version 2

Abstract

Different energy sources can have very different characteristics in terms of power range and energy demand/cost function also known as efficiency function or energy conversion function. Introducing these energy sources characteristics in combinatorial optimization problems such as energy resource allocation problems or energy-consuming activity scheduling problems results into mixed-integer non-linear problems neither convex or concave. Approximations via piecewise linear functions have been proposed in the literature. Non-convex optimization models and heuristics exist to compute optimal breakpoint systems subject to the condition that the piecewise linear continuous approximator (under- and overestimator) never deviates more than a given delta-tolerance from the original continuous separable function over a given finite interval, or to minimize the area between the approximator and the function. We present an alternative solution method based on the upper and lower bounding of energy conversion expressions using non necessarily continuous piecewise linear functions with a relative epsilon-tolerance. Conditions under which such approach yields a pair of mixed integer linear programs with a performance guarantee are analyzed. Models and algorithms to compute the non necessarily continuous piecewise linear functions with absolute and relative tolerances are also presented. Computational evaluations performed on energy optimization problems for hybrid electric vehicles show the efficiency of the method with regards to the state of the art.

Keywords: OR in energy, Combinatorial optimization, Non-linear efficiency functions, , Piecewise linear bounding

1 Introduction

Various optimization problems resulting from the introduction of energy conversion functions into combinatorial optimization problems can be modeled as non linear problems or mixed

integer non linear problems (MINLP). Let us consider for example energy optimization in hybrid electric vehicles. In such vehicles the electrical powertrain system has multiple energy sources that it can gather power from to satisfy the propulsion power requested by the vehicle at each instant. The problem usually consists in finding at each instant the optimal power split between the multiple energy sources to satisfy the power demand of a driver on a predefined road section. The objective is to minimize the total fuel consumption of the vehicle performing a predefined mission, taking into account the characteristics and the limitations of each energy source, such as the energy losses happening during any energy transfer. Problem (P) models such problem for a HEV operating with two energy sources: (1) a Fuel Cell stack (FC) able to produce power from P_{\min}^1 up to P_{\max}^1 at each instant $i \in \{1 \dots I\}$, (2) a storage element (SE) able to retrieve power up to P_{\min}^2 and to produce power up to P_{\max}^2 at each instant $i \in \{1 \dots I\}$. The amount of energy stored in the SE is also called state of charge (SOC). To avoid a premature aging of SE, its state of charge is only allowed to vary between E_{\min} and E_{\max} , typically 25% and 100% of its energy capacity. Problem (P) can be modeled with equations (1)-(5) where x_i^1 , x_i^2 and x_i^3 are the amount of energy produced by FC, produced by SE and retrieved by SE at instant $i \in \{1 \dots I\}$. Problem (P) is a (MI)NLP because of non-linear energy conversion functions f^1 , f^2 and f^3 continuous on intervals $[P_{\min}^1, P_{\max}^1]$, $]0, P_{\max}^1]$ and $]0, P_{\max}^1]$ respectively and verifying $f^1(0) = f^2(0) = f^3(0) = 0$; $f^1(x) = 0, \forall x < P_{\min}^1$, often with a discontinuity at P_{\min}^1 that requires the insertion of binary variables to be modeled.

$$(P) \quad \min \sum_{i=1}^I f^1(x_i^1) \quad // \text{minimize total cost on FC} \quad (1)$$

subject to (s.t.)

$$x_i^1 + x_i^2 - x_i^3 \geq P_i, \quad \forall i \in \{1 \dots I\} \quad // \text{power demand satisfaction} \quad (2)$$

$$\sum_{i=1}^I (f^2(x_i^2) - f^3(x_i^3)) \leq 0 \quad // \text{final SOC} \geq \text{initial SOC} \quad (3)$$

$$E_0 - E_{\max} \leq \sum_{k=1}^i (f^2(x_k^2) - f^3(x_k^3)) \leq E_0 - E_{\min}, \quad \forall i \in \{1 \dots I\} \quad // \text{SOC limits} \quad (4)$$

$$x_i^1 \in \{0\} \cup [P_{\min}^1, P_{\max}^1], x_i^2 \in [0, P_{\max}^2], x_i^3 \in [P_{\min}^2, 0], \quad \forall i \in \{1 \dots I\} \quad // \text{domain definition} \quad (5)$$

It is possible for the resulting (MI)NLP modelling an energy optimization problem to be neither convex or concave even when all energy conversion functions are convex or concave. Therefore, only small instances may be tractable using standard MINLP solvers. Several real-world applications have been addressed using piecewise linear approximations of the non linear functions of the MINLP to obtain a MILP easier to solve (see for example Camponogara et al. (2007), Borghetti et al. (2008), DAmbrosio et al. (2010), Boukouvala et al. (2016)). Such approach presents the main advantage of producing solutions faster than purely MINLP-based approaches if not too many additional binary variables or combinatorial conditions have been introduced in the process, meaning that the number of pieces of the piecewise linear (pwl) functions used should be kept to a minimum. Geißler et al. (2012) explain that the approach suffers from a few drawbacks due to the fact that the nature of the nonlinearities inherent to the problem may be lost and the solutions obtained from the MILP may have no meaning for the MINLP. If the solution obtained is not satisfactory, a new pwl approximation may be performed using a higher number of pieces, to obtain a new MILP to solve. This yields an iterative solution procedure with a number of iterations unknown a priori which translates into

high computing times, either because several iterations needed to be performed, or because an unnecessary large number of pieces were chosen resulting into an unnecessarily large MILP that required a high solution time.

As an alternative we propose a straightforward two-phase solution method. The first phase consists in bounding each non linear term from above and below using a pair of piecewise linear functions satisfying conditions that will be specified in the core of the paper. Contrary to most publications on piecewise linear approximation which focus on the minimization of the approximation error for a given number of pieces or breakpoints that may or may not be equidistant, we aim at minimizing the number of pieces for a given error. The second phase of the solution method proposed consists in solving two MILP obtained from the replacement of the non linear functions with either one of the two piecewise linear functions. This paper focusses on the first phase of the iterative procedure, assuming that the resulting MILPs can be solved efficiently with a MILP solver. If it is not the case, then a specific solution method for the resulting MILP may need to be designed and an example of such case is presented in Ngueveu et al. (2016), which focused on the MILP solution without specifying how to obtain the piecewise linear functions.

2 State of the art

Several publications exists on the application of piecewise linear (pwl) approximation on non linear univariate functions to solve MINLP problems, but the issue addressed in the large majority of them is to minimize the approximation error given a predefined number of breakpoints or pieces. To the best of our knowledge, only three papers focused on the specific problem of minimizing the number of breakpoints for a given precision or bounded approximation error.

Rosen and Pardalos (1986) were the first to propose the computation of breakpoints for a given error tolerance for concave quadratic functions. The pwl interpolators were built using equidistant breakpoints and by concavity their interpolators were underestimators. Rosen and Pardalos identified conditions on the number of breakpoints they required to achieve a given error tolerance. Geißler et al. (2012) showed that certain cases of general MINLPs can be solved by just applying techniques purely from the mixed integer linear programming by approximating the nonlinearities with pwl functions. They proposed to compute a priori errors for pwl approximations or a priori errors for over- and under- pwl estimators. However, they did not focus on the computation of optimal (minimal) breakpoint systems.

Rebennack and Kallrath (2015) propose two exact approaches and two heuristics for the computation of optimal continuous pwl approximators for univariate continuous functions over a compactum $\mathbb{D} = [X_-, X_+]$. Their methodology is also applicable if the function has finitely many discontinuities. Their algorithms handle more general functions than the ones of Rosen and Pardalos (1986). In addition, their breakpoints are distributed freely and shifts from the function are allowed at breakpoints, which were shown to be important degrees of freedom contributing to a significant reduction of the number of breakpoints. The work of Rebennack and Kallrath (2015) differs from Geißler et al. (2012) in the following aspects. Geißler et al. do not target on computing minimal breakpoint systems whereas Rebennack and Kallrath do so. Then the latter solve non-convex NLP problems to global optimality to

obtain the tightests approximators, tightest in the sense of minimizing the largest deviation. In addition, Rebennack and Kallrath consider shift variables at breakpoints, which adds an additional degree of freedom. Since their work is close to ours in some aspects, let us focus on their contributions before highlighting the main differences and contributions of this paper.

Rebennack and Kallrath (2015) show that ensuring that the approximator and the original function do not deviate more than a predefined tolerance δ from each other leads to sets of constraints which have to hold over a continuum, resulting in a semi-infinite programming (SIP) problem denoted OBSC for “Optimal Breakpoint System using a Continuum approach for x ”. The authors show that it is NP-hard to compute a δ -approximator for an arbitrary continuous function and propose an iterative solution procedure based on the evaluation of the continuum conditions only on a discrete set of grid points, resulting into a MINLP model which is a relaxation of the SIP. The feasibility of the resulting solution with regards to OBSC is then evaluated by solving an NLP on each interval corresponding to a line-segment of the pwl approximator, to compute the true maximal deviation between the approximator and the original function. The algorithm stops if the true deviation is less or equal to δ on all line-segments. In this case the solution obtained is optimal for OBSC. Otherwise the grid is refined to obtain a new MINLP to be solved. Two different discretization strategies were proposed, resulting in MINLP models denoted OBSD and OBSI respectively. OBSI considers a uniform discretization of the entire interval $[X_-, X_+]$ whereas OBSD discretizes uniformly the intervals between pairs of consecutive breakpoints. OBSC, OBSD and OBSI are in general too large and difficult to solve to optimality even for small numbers of breakpoints and discretization points, therefore Rebennack and Kallrath (2015) proposed two heuristic methods.

The heuristics methods were based on the successive computation of the breakpoints, from X_+ to X_- , maximizing at each iteration the length of the interval corresponding to the projection of the pwl approximation on the x -axis. This meant solving at each iteration a problem denoted BSB that can be expressed as follows: given the breakpoint x_i ending the i^{th} piece (which corresponds by continuity to the beginning of the $i + 1^{\text{th}}$ piece), compute the next breakpoint x_{i+1} (end of the $i + 1^{\text{th}}$ piece) so as to maximize x_{i+1} while ensuring a deviation of at most δ between the original function and the linear approximation on interval $[x_i, x_{i+1}]$. Rebennack and Kallrath provided a counter-example showing that maximizing the length of the intervals do not necessarily lead to an optimal breakpoint system i.e., to a δ -approximator with the least number of breakpoints. Indeed, it might be beneficial, in certain cases, to consider intervals between two breakpoints which are not of maximal length, contrary to what is stated for example in Frenzen et al. (2010). The counter-example is illustrated on Figure 1. It shows for a function $f(x)$ defined with equation (6) on $[X_-, X_+] = [0, 5]$ that the authors proposed, and $\delta = 0.25$, that the unique optimal 0.25-approximator uses three breakpoints whereas maximizing the interval length successively from X_- to X_+ obtains a 0.25-approximator using four breakpoints. The Forward Heuristic with Moving Breakpoints (FHMB) solved each BSB problem to optimality with the iterative “grid discretization + NLP solution” approach. Its main limitation was the necessity to solve many NLPs. The α -Forward Heuristic with Backward Iterations (FHBI) solved each BSB problem heuristically by trying different decreasing values for x_{i+1} with a predefined step parameter α . FHBI solves less NLPs than FHBM and therefore requires less computing time, but FHBI obtains better solutions. Using any of the heuristics, it is possible to obtain breakpoint systems satisfying the required δ -tolerance, and more so, an upper bound on the minimal number of breakpoints.

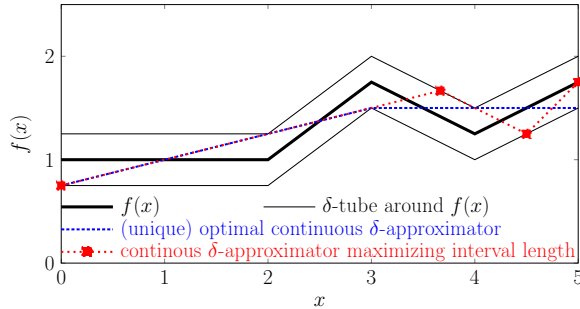


Figure 1: Maximizing the length of the intervals successively is not optimal, in general — $f(x)$ — δ -tube around $f(x)$ - - (unique) optimal δ -approximator ... δ -approximator maximizing interval length successively (source: Rebnack and Kallrath (2015), page 628)

$$f(x) = \begin{cases} 1 & \text{if } x \in [0, 2[\\ 0.75x - 0.5 & \text{if } x \in [2, 3[\\ -0.5x + 3.25 & \text{if } x \in [3, 4[\\ 0.5x - 0.75 & \text{if } x \in [4, 5] \end{cases} \quad (6)$$

It is worth mentioning that there exists publications on piecewise linear approximation with a minimum number of pieces given a predefined bound on the absolute error in the fields of data reduction, pattern recognition or classification and ECG waveform preprocessing (Tomek (1974a), Tomek (1974b), Gritzali and Papakonstantinou (1983)). The main difference with our problem is that those publications consider as an input a discrete set of points. Their objective is the find the piecewise linear function with a minimum number of pieces such that the error for each of the sample point is less than the allowed value δ . Even in cases where an analytical expression of a continuous function was available, the function was sampled and the approximation was performed on the set of sample points. Therefore, the algorithms proposed in these research fields do not ensure the respect of the predefined approximation error on the entire interval $[X-, X+]$ and are therefore not applicable to our problem. Finally, the field of piecewise linear approximation of planar curves could be mentioned (Dunham (1986), Papakonstantinou et al. (1994)), with applications related to shape analysis or pattern classification, since a non linear function $f(x)$ could be represented as a parametric curve $x(t) = t, y(t) = f(t)$. However, the algorithms proposed in that research field are not applicable to our problems for two main reasons: (i) input curves considered are discrete or digitised, not continuous ones as ours, and (ii) the error between a point of the original curve and the approximating planar curve is the Euclidean distance of the point to the nearest point on the piecewise linear curve, which does not correspond to the approximation error between original function and the piecewise linear function.

In the view of the state-of-the-art, the contributions of the paper are the following: (1) instead of using continuous pwl δ -approximators, we propose to approximate general univariate continuous functions with non-necessarily continuous piecewise linear δ -approximators, adding an additional degree of freedom to obtain a breakpoint system of equal or less breakpoints, (2) we prove that when discontinuity is allowed, maximizing the interval length produces an opti-

mal pwl δ -approximator, leading to an exact solution procedure based on the iterative solution of adapted BSB problems, (3) we introduce relative ϵ -tolerance and show the benefit of using it instead of the absolute δ -tolerance, (4) we show that when using relative ϵ -tolerance, it is not possible to shift up (down) an optimal approximator to derive an optimal over- (under-) estimator, contrary to what can be done for when using absolute δ -tolerance, (5) models and algorithms to compute the discontinuous pwl under- and over-estimator with absolute or relative tolerance are presented, (6) for solving MINLP involving non linear univariate energy conversion functions, we propose a solution method based on the upper and lower bounding of energy conversion expressions using discontinuous piecewise linear functions with a relative ϵ -tolerance, and the solution of a pair of mixed integer linear programs, (7) we prove that such approach yields a performance guarantee when the non linearity is restricted to the objective-function and finally (8) computational results on energy optimization problems for hybrid electric vehicles illustrate the efficiency of the method in comparison to state-of-the-art methods including solution procedures based on approximations with absolute δ -tolerance.

3 Non-necessarily continuous pwl δ -approximation of continuous non-linear functions

Let $f : \mathbb{D} = [X_-, X_+] \rightarrow \mathbb{R}$ be a function on the compact interval $\mathbb{D} \subset \mathbb{R}$. A function $g : \mathbb{D} = [X_-, X_+] \rightarrow \mathbb{R}$ is a pwl function with $n_g \in \mathbb{N}$ line-segments if $\exists a \in \mathbb{R}^{n_g}, b \in \mathbb{R}^{n_g}, x^{\min} \in [X_-, X_+]^{n_g}$ and if $\forall i \in [1..n_g], \exists x_i^{\max} \in]x_i^{\min}, X_+]$, such that equations (7)-(10) are verified. Such pwl function is said to be defined by $G = \bigcup_{i=1}^{n_g} ([a_i, b_i], [x_i^{\min}, x_i^{\max}])$ and the two end-points x_i^{\min} and x_i^{\max} of each line-segment i are called breakpoints.

$$g(x) = a_i x + b_i, \quad \forall i \in [1..n_g], \forall x \in [x_i^{\min}, x_i^{\max}] \quad (7)$$

$$x_i^{\max} = x_{i+1}^{\min}, \quad \forall i \in [1..n_g - 1] \quad (8)$$

$$x_1^{\min} = X_- \quad (9)$$

$$x_{n_g}^{\max} = X_+ \quad (10)$$

A pwl function g is:

- continuous iff it verifies all equations (11)
- or discontinuous if it does not verify all equations (11), i.e $\exists j \in [1, n_g - 1]$ such that $(a_i x_i^{\max} + b_i) \neq (a_{i+1} x_{i+1}^{\min} + b_{i+1})$
- or non-necessarily continuous if the satisfaction of equations (11) is neither required or forbidden

$$a_i x_i^{\max} + b_i = a_{i+1} x_{i+1}^{\min} + b_{i+1}, \forall i \in [1, n_g - 1] \quad (11)$$

Definition 3.1 (δ -approximator): A pwl function $g : \mathbb{D} = [X_-, X_+] \rightarrow \mathbb{R}$ is called δ -approximator of a function $f : \mathbb{D} = [X_-, X_+] \rightarrow \mathbb{R}$ with $\delta \in \mathbb{R}^+$, iff inequation (12) is verified.

$$\max_{x \in \mathbb{D}} |l(x) - f(x)| \leq \delta \quad (12)$$

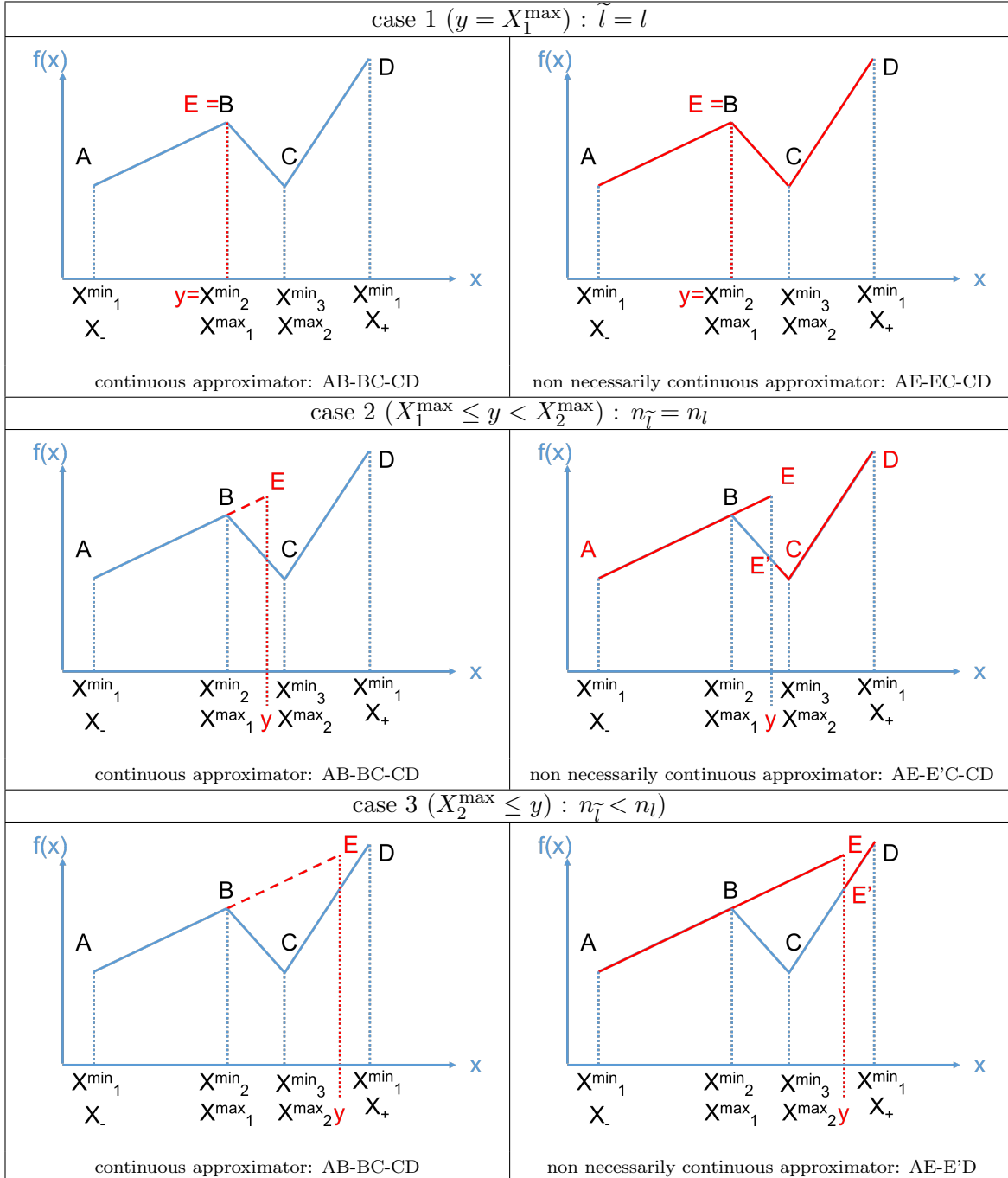


Figure 2: Maximization of the first interval (projection on \mathbb{D} of the first line-segment)

Proposition 3.2 Any optimal pwl continuous δ -approximator with n^* line-segments can be converted into a pwl non-necessarily continuous δ -approximator with $n \leq n^*$ line-segments where the projection of the first line-segment on interval \mathbb{D} is of maximal length.

Proof Let $l = \bigcup_{i=1}^{n_l} ([a_i, b_i], [x_i^{\min}, x_i^{\max}])$ be an optimal continuous δ -pwl approximator ($n_l = n^*$) of a continuous function $f : \mathbb{D} = [X_-, X_+] \rightarrow \mathbb{R}$. Let $y \in \mathbb{D}$ be the solution value of problem (PY):

$$(PY) \begin{cases} \text{s.t.} & \max y & // \text{maximize interval length} \\ & |a_1 x + b_1 - f(x)| \leq \delta, \quad \forall x \in [x_1^{\min}, y] & // \delta\text{-approximation constraint} \\ & y \in [x_1^{\min}, x_i^{\max}] & // \text{domain definition} \end{cases}$$

Let $q \in [1 \dots n_l]$ be the piece number that verifies $x_q^{\min} \leq y \leq x_q^{\max}$. Breakpoint x_1^{\max} verifies $|a_1 x_1^{\max} + b_1 - f(x)| \leq \delta$, therefore $y \geq x_1^{\max}$. If $y = x_1^{\max}$, then $\tilde{l} = l$ is a pwl continuous δ -approximator of f with $n = n^*$ pieces where the projection of the first line-segment on interval \mathbb{D} is of maximal length. Otherwise $y > x_1^{\max}$ in which case a discontinuous pwl δ -approximator \tilde{l} with $n_{\tilde{l}} \leq n^*$ line-segments where the projection of the first line-segment on interval \mathbb{D} is of maximal length can be defined by $\tilde{L} = \bigcup_{i=1}^{n_{\tilde{l}}} ([\tilde{a}_i, \tilde{b}_i], [\tilde{x}_i^{\min}, \tilde{x}_i^{\max}]) = ([a_1, b_1], [x_1^{\min}, y]) \cup ([a_q, b_q], [y, x_q^{\max}]) \cup \left(\bigcup_{i=q+1}^{n_{\tilde{l}}} ([a_i, b_i], [x_i^{\min}, x_i^{\max}]) \right)$. ■

Theorem 3.3 For any continuous function $f : \mathbb{D} = [X_-, X_+] \rightarrow \mathbb{R}$ and any scalar $\delta \in \mathbb{R}^+$, there exists an optimal non-necessarily continuous pwl δ -approximator g defined by $G = \bigcup_{i=1}^{n_g} ([a_i, b_i], [x_i^{\min}, x_i^{\max}])$ such that each line-segment i has a maximal length projection on the interval $[x_i^{\min}, X_+]$.

Proof There exists a continuous δ -approximator function for any continuous function f on a compactum \mathbb{D} and any scalar $\delta > 0$ (Duistermaat and Kol, 2004). Proposition 3.2 can be applied iteratively on an optimal continuous δ -approximator to obtain an optimal non-necessarily continuous pwl δ -approximator with intervals of maximal length on \mathbb{D} . i.e Applying proposition 3.2 ensures that the first line-segment has a maximal length projection on interval \mathbb{D} . Let $y = x_1^{\max}$ be the resulting breakpoint, then proposition 3.2 can be applied for function $\tilde{f} = f : [x_1^{\max}, X_+]$ to find the next line-segment of maximal length projection. The procedure can be repeated until the end of the compactum \mathbb{D} is reached, i.e $y = X_+$. ■

Figure 2 illustrates the three possible cases related to Proposition 3.2. Figure 3(a) illustrates the implications of theorem 3.3 for the function f defined with equation (6): maximizing the length of intervals leads to a discontinuous pwl 0.25-approximator of two line-segments, which is the optimal number of line-segments for any non necessarily continuous pwl 0.25-approximator of f . In this example optimal continuous and non necessarily continuous pwl approximators have the same number of line-segments. Figure 3(b) illustrates an example where it is not the case: one has more line-segments than the other. In the general case, it can be easily proven that given a continuous function $f : \mathbb{D} = [X_-, X_+] \rightarrow \mathbb{R}$ and given a scalar $\delta \in \mathbb{R}^+$, if n_{nnc} is the number of line-segments of an optimal non necessarily continuous pwl δ -approximator of f and if n_c is the number of line-segments of an optimal continuous pwl δ -approximator of f , then $\lceil \frac{n_c+1}{2} \rceil \leq n_{nnc} \leq n_c$.

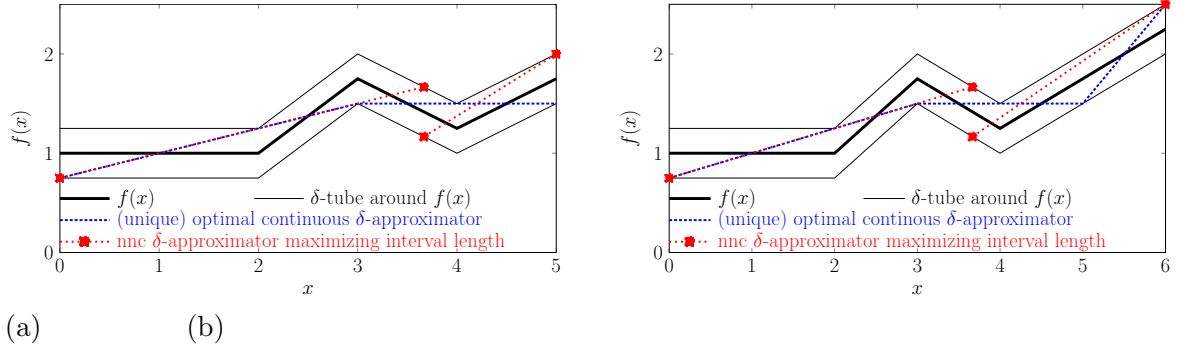


Figure 3: Maximizing the length of the intervals successively is optimal for non necessarily continuous δ -approximators: — $f(x)$ — δ -tube around $f(x)$ - - an optimal continuous δ -approximator ... an optimal non necessarily continuous δ -approximator

The mathematical models and algorithms from Rebennack and Kallrath (2015) for computing continuous pwl δ -approximators are modified to compute non-necessarily continuous pwl δ -approximators by considering two shifts per breakpoint instead of one, namely s_b^- used for the line-segment ending a breakpoint b and s_b^+ used for the line-segment starting at breakpoint b . In particular, thanks to Theorem 3.3, Algorithm 1 resulting from the adaptation of Rebennack and Kallrath’s Forward Heuristic with Moving Breakpoints yields an exact solution method for computing optimal non-necessarily continuous δ -approximators. It’s main limitation is the necessity to solve an SIP. Section 4 presents, among other things, an exact method applicable to convex and concave functions that does not need to solve any SIP or any NLP. The method is shown to provide performance guarantees for functions that are neither convex or concave but that can be decomposed into convex or concave pieces, if the decomposition is part of the input.

Algorithm 1 Compute an optimal non-necessarily continuous pwl approximation

Input: Function f ; domain $\mathbb{D} = [X_-; X_+]$; scalar $\delta > 0$

Output: pwl function g defined by G

- 1: $n_g := 0; y = X_-; G := \emptyset; x^{\text{end}} := X_+$
 - 2: **while** $y < X_+$ **do**
 - 3: $x^{\text{begin}} := y$
 - 4: solve SIP (13)-(15) to obtain a, b and y
 - 5: $G := G \cup ((a, b), (x^{\text{begin}}, y))$
 - 6: $n_g := n_g + 1$
 - 7: **end while**
-

$$\max y \quad // \text{maximize interval length} \quad (13)$$

s.t.

$$|ax + b - f(x)| \leq \delta, \quad \forall x \in [x^{\text{begin}}, y] \quad // \delta\text{-approximation constraint} \quad (14)$$

$$y \in [x^{\text{begin}}, x^{\text{end}}] \quad // \text{domain definition} \quad (15)$$

4 From pwl δ -approximation to pwl δ -bounding

A benefit of computing δ -approximators optimal in terms of number of breakpoints or line-segments is to minimize the number of additional binary variables added to obtain the MILP resulting from the replacement of non linear terms with their pwl approximators. A drawback of applying δ -approximation is that the optimal solution of the resulting MILP can be infeasible with respect to the original MINLP. In this case the solution may not be of interest for the practitioners who formulated the original problem. An alternative is to use over- and under-estimators defined as follows.

Definition 4.1 (δ -underestimator): A pwl function $\underline{l} : \mathbb{D} \rightarrow \mathbb{R}$ is a δ -underestimator of a function $f : \mathbb{D} \rightarrow \mathbb{R}$ with $\delta \in \mathbb{R}^+$ iff it verifies inequation (16).

$$\underline{l}(x) \leq f(x) \leq \underline{l}(x) + \delta, \forall x \in \mathbb{D} \quad (16)$$

Definition 4.2 (δ -overestimator): A pwl function $\bar{l} : \mathbb{D} \rightarrow \mathbb{R}$ is a δ -overestimator of a function $f : \mathbb{D} \rightarrow \mathbb{R}$ with $\delta \in \mathbb{R}^+$ iff it verifies inequation (17).

$$\bar{l}(x) \geq f(x) \geq \bar{l}(x) - \delta, \forall x \in \mathbb{D} \quad (17)$$

In the context of energy optimization for hybrid electric vehicles for example, replacing each non-linear energy loss function with its pwl δ -overestimator yields a $\overline{\text{MILP}}$ solution where a sufficient or excess amount of energy is produced at each time-step. Since excess energy can be dissipated as heat (in a resistance inserted in the braking system or in mechanical brakes present for security reasons) therefore the $\overline{\text{MILP}}$ solution is applicable on a test bench or on the real world hybrid electric vehicle considered. Having ensured the feasibility of the solution obtained, it is of interest to provide an estimate of the quality of the solution with respect to the optimum of the original MINLP. This can be done using a pwl δ -underestimator of the energy loss function and solving the resulting $\underline{\text{MILP}}$ problem to obtain a lower bound of the MINLP. Each non-linear function is therefore bounded from above and below with two pwl functions, yielding two MILPs whose optimal solutions cost verify: $z_{\underline{\text{MILP}}} \leq z_{\text{MINLP}} \leq z_{\overline{\text{MILP}}}$. Table 1 summarizes which estimator should be used for each non-linear term of a MINLP to obtain a $\underline{\text{MILP}}$ or a $\overline{\text{MILP}}$ in the general case.

Origin	non-linear term	$\overline{\text{MILP}}$	$\underline{\text{MILP}}$
Objective-function to minimize	$+f(x)$	$+\bar{l}(x)$	$+\underline{l}(x)$
	$-f(x)$	$-\underline{l}(x)$	$-\bar{l}(x)$
Constraint of type \geq	$+f(x)$	$+\underline{l}(x)$	$+\bar{l}(x)$
	$-f(x)$	$-\bar{l}(x)$	$-\underline{l}(x)$
Constraint of type \leq	$+f(x)$	$+\bar{l}(x)$	$+\underline{l}(x)$
	$-f(x)$	$-\underline{l}(x)$	$-\bar{l}(x)$

Table 1: Estimator in replacement of each non-linear term $f(x) \geq 0$ to obtain $\overline{\text{MILP}}$ or $\underline{\text{MILP}}$

Given an optimal continuous pwl δ -approximator g of a function f , an optimal continuous pwl 2δ -over- (resp. under-) estimator can be obtained from shifting g by δ (resp. $-\delta$), as stated by Rebennack and Kallrath (2015) i.e. $g(x) + \delta$ (resp. $g(x) - \delta$) is a 2δ -over

(resp. under-) estimator of f . This result can be extended to non-necessarily continuous pwl approximators and estimators. In addition, in the case of energy optimization for example, the MINLP can be neither convex or concave even though the individual non-linear energy loss functions or energy demand/cost conversion functions are convex or concave. Yet, if a non-linear continuous function $f : \mathbb{D} \rightarrow \mathbb{R}$ is convex (resp. concave) over \mathbb{D} with a derivative efficiently computable at any point of \mathbb{D} , then taking advantage of equation (18) it is possible to compute an optimal δ -underestimator (resp. δ -overestimator) without solving any SIP or even NLP. Then shifting can be applied to obtain an optimal δ -overestimator (resp. δ -underestimator).

$$f \text{ is convex over } \mathbb{D} \Leftrightarrow f(y) \geq f(x) + f'(x)(y - x), \forall x, y \in \mathbb{D} \quad (18)$$

Let us focus for example on the computation of the optimal δ -underestimator of a non-linear continuous convex function f with a derivative efficiently computable. The reasoning hereafter can be adapted to compute the optimal δ -overestimator of concave functions. Thanks to Theorem 3.3 the objective at each iteration i of the algorithm is to maximize x_i^{\max} for a given x_i^{\min} . Each line-segment i is tangent to f (otherwise it can be replaced with a line-segment of identical slope but tangent to f). Therefore the objective at iteration i is to identify a tangent point $q_i \in [x_i^{\min}, x_i^{\max}]$ that defines the slope $a_i = f'(q_i)$ and the y-intercept $b_i = f(q_i) - f'(q_i)q_i$ of line-segment i so that x_i^{\max} is maximized given x_i^{\min} . Given q_i , the gap in function of y defined as $f(y) - (f'(q_i)y + f'(q_i)q_i)$ increases when y decreases if $y \leq q_i$, or when y increases if $y \geq q_i$. Therefore, for a given q_i , checking whether the δ -approximation constraint is verified at points x_i^{\min} and x_i^{\max} is sufficient to ensure the validity of the constraint on the entire interval $[x_i^{\min}, x_i^{\max}]$. In this context proposition 4.3 can be enunciated.

Proposition 4.3 *Finding the line-segment that maximizes x_i^{\max} for a given x_i^{\min} is equivalent to solving sequentially two separate problems: (1) maximizing q_i for the given x_i^{\min} , and then (2) maximizing x_i^{\max} for the previously computed q_i .*

Proof Let $x_i^{\max}(q_i^{(k)})$ be the maximal value possible of x_i^{\max} for a given $q_i^{(k)}$. The proof of property 4.3 consists in proving that $x_i^{\max}(q_i^{(2)}) \geq x_i^{\max}(q_i^{(1)}) \Rightarrow q_i^{(2)} \geq q_i^{(1)}$. This is done using equation (18) which leads to $0 \leq f(y) - (a_i^{(2)}y + b_i^{(2)}) \leq f(y) - (a_i^{(1)}y + b_i^{(1)}), \forall y \geq q_i^{(2)}$. ■

Algorithm 2 summarizes the resulting procedure that computes an optimal δ -underestimator of a non-linear convex function f with a derivative efficiently computable at any point of $[X_-, X_+]$. Problems (PQ) and (PY) can be solved to optimality with dichotomy search methods as illustrated by algorithms 5 and 6 where parameter α is the number of significant digits for q_i , x_i^{\min} and x_i^{\max} . Notice that at no time is there any need to solve an SIP or NLP. It can be noted that any two consecutive line-segments ($[a_i, b_i], [x_i^{\min}, x_i^{\max}]$) and ($[a_{i+1}, b_{i+1}], [x_{i+1}^{\min}, x_{i+1}^{\max}]$) of the pwl function obtained verify $a_i \neq a_{i+1}$ and $a_i x_i^{\max} + b_i \leq a_{i+1} x_{i+1}^{\min} + b_{i+1}$. Their supporting straight lines intersect on a point C of coordinates (x_C, y_C) such that $y_C = a_i x_C + b_i = a_{i+1} x_C + b_{i+1}$ and $x_i^{\min} \leq x_C \leq x_{i+1}^{\max}$ and $f(x_C) - y_C \leq \delta$. Consequently, the non necessarily continuous pwl δ -under-estimator computed with Algorithm 2 can be converted into a continuous pwl δ -under-estimator with the same number of pieces.

Algorithm 2 can be adapted to compute directly an optimal δ -overestimator of a convex function: a tangent point q_i in this case defines the slope $a_i = f'(q_i)$ and the y-intercept $b_i = f(x_i^{\min}) - f'(q_i)x_i^{\min}$. It can also be adapted for the pwl bounding of concave functions with derivatives efficiently computable. Let us consider the case of continuous functions that are neither concave or convex but that can be decomposed into p pieces, each piece being either concave or convex. If the decomposition is part of the input, then pwl bounding the function can be done by bounding each piece separately using the algorithms previously described, then aggregating the pwl functions obtained. To that end, let $k_i \in \mathbb{D}, \forall i \in \{1..p+1\}$ be a set of points such that (i) on each interval $[k_i, k_{i+1}]$ the function $f(x)$ is either convex or concave, (ii) $k_1 = X_-$, (iii) $k_{p+1} = X_+$, and (iv) $k_i < k_{i+1}, \forall i \in \{1..p\}$. For each interval $[k_i, k_{i+1}]$, let g_i be the optimal pwl over-(resp. under-) estimator of f computed with the algorithms from Section 4. Let g be an over (resp. under) estimator of function f on interval \mathbb{D} , resulting from the union of pwl functions g_i , i.e $g = \cup_{i \in \{1..p\}} g_i$. Let n_{g_i} (resp. n_g) be the number of line-segments of g_i (resp. g). If n^* is the number of line-segments of any optimal pwl over-(resp. over-) estimator of f on interval \mathbb{D} , then $n_g = \sum_{i=1}^p n_{g_i}$ and equation (19) is verified.

$$n^* \leq n_g \leq n^* + p - 1 \quad (19)$$

Algorithm 2 Compute an optimal non-necessarily continuous pwl δ -underestimator

Input: Convex function f ; domain $\mathbb{D} = [X_-; X_+]$; scalar $\delta > 0$

Output: pwl function g defined by G

- 1: $n_g := 0; y = X_-; G := \emptyset$
 - 2: **while** $y < X_+$ **do**
 - 3: $x^{\text{begin}} := y$
 - 4: solve problem (PQ) to obtain q and deduce $a = f'(q), b = f(q) - f'(q)q$

$$\text{(PQ)} \left\{ \begin{array}{l} \max q \quad // \text{maximize tangent point} \\ \text{s.t.} \\ f(x^{\text{begin}}) - (f(q) + f'(q)(x^{\text{begin}} - q)) \leq \delta, \quad // \delta\text{-approximation constraint} \\ q \in [x^{\text{begin}}, X_+] \quad // \text{domain definition} \end{array} \right.$$
 - 5: solve problem (PY) to obtain y

$$\text{(PY)} \left\{ \begin{array}{l} \max y \quad // \text{maximize ending breakpoint} \\ \text{s.t.} \\ f(y) - (ay + b) \leq \delta, \quad // \delta\text{-approximation constraint} \\ y \in [q, X_+] \quad // \text{domain definition} \end{array} \right.$$
 - 6: $G := G \cup ((a, b), (x^{\text{begin}}, y))$
 - 7: $n_g := n_g + 1$
 - 8: **end while**
-

Algorithm 3 Solve problem (PQ)

Input: Convex function f ; $x^{\text{begin}}; X_+; \delta > 0; \alpha \in \mathbb{N}$

Output: tangent point q , slope a , y-intercept b

- 1: $\{q, a, b, 0, 0\} = \text{Dichotomy_}q(f, x^{\text{begin}}, X_+, \delta, \alpha, x^{\text{begin}}, X_+, 0, 0, 0)$ {use Algorithm 5}
-

Algorithm 4 Solve problem (PY)

Input: Convex function $f; X_+; \delta > 0; \alpha \in \mathbb{N}$; tangent point q , slope a , y-intercept b

Output: next breakpoint y

1: $\{q, 0, 0\} = \text{Dichotomy_}y(f, x^{\text{begin}}, X_+, \delta, \alpha, q, X_+, 0, 0, 0)$ {use Algorithm 6}

Algorithm 5 Dichotomy_ $_q$

Input: Convex function $f; x^{\text{begin}}; X_+; \delta > 0; \alpha \in \mathbb{N}; B_{\min}, B_{\max}; q; a; b$

Output: tangent point q , slope a , y-intercept b , lower limit B_{\min} , upper limit B_{\max}

1: **if** $B_{\max} - B_{\min} \leq 10^{-\alpha}$ **then**

2: $q = B_{\min}; a = f'(q); b = f(q) - f'(q)q$

3: **return** q, a, b

4: **else**

5: $\tilde{a} = f'(x^{\text{begin}}); \tilde{b} = f(x^{\text{begin}}) - f'(q_i)q_i$

6: **if** $f(x^{\text{begin}}) - (\tilde{a}x^{\text{begin}} + \tilde{b}) \leq \delta$ **then**

7: $B_{\min} = B_{\max}; B_{\max} = X_+;$

8: **else**

9: $B_{\max} = B_{\min} + 0.5(B_{\max} - B_{\min})$

10: **end if**

11: $\{q, a, b, B_{\min}, B_{\max}\} = \text{Compute_}q(f, x^{\text{begin}}, X_+, \delta, \alpha, B_{\min}, B_{\max}, q, a, b)$

12: **end if**

Algorithm 6 Dichotomy_ $_y$

Input: Convex function $f; X_+; \delta > 0; \alpha \in \mathbb{N}; q; a; b; B_{\min}, B_{\max}$

Output: next breakpoint y , lower limit B_{\min} , upper limit B_{\max}

1: **if** $B_{\max} - B_{\min} \leq 10^{-\alpha}$ **then**

2: $y = B_{\max}$

3: **return** y

4: **else**

5: **if** $f(B_{\max}) - (aB_{\max} + b) \leq \delta$ **then**

6: $B_{\min} = B_{\max}; B_{\max} = X_+;$

7: **else**

8: $B_{\max} = B_{\min} + 0.5(B_{\max} - B_{\min})$

9: **end if**

10: $\{y, B_{\min}, B_{\max}\} = \text{Compute_}y(f, X_+, \delta, \alpha, q, a, b, B_{\min}, B_{\max})$

11: **end if**

5 Drawbacks and limitations of pwl δ -bounding

When applying a “ δ -bounding + MILP solution” procedure, choosing relevant δ values is a challenging issue. The chosen value should be orders of magnitude smaller than the resulting solution cost to provide an acceptable precision level. After solving the MILP, verifying

whether the chosen tolerance value δ was sufficiently small in the view of the resulting solution values is straightforward. If it was not the case then the value of δ is decreased before a new round of pwl bounding then MILP solution. Such iterative procedure is not satisfactory because the number of iterations is unknown a priori and the pwl bounding and/or the MILP solution may require significant computational efforts during each single iteration. An alternative is to identify a target precision Δ of the solution obtained from the MILP with regards to the optimal MINLP solution value and precompute a priori the corresponding δ values in function of the chosen Δ . When non linearity occurs in the constraints, the link between δ and Δ is not obvious. But even when non linearity occurs only in the objective-function, solving the MINLP resulting from the introduction of the non-linear energy conversion functions into a combinatorial optimization problem, with the “ δ -bounding + MILP solution” procedure and a target precision Δ presents significant drawbacks and limitations described in the remainder of this section.

5.1 Data dependence leading to multiple δ values

Let us consider the problem (1)-(5) for a specific HEV, which, for simplicity, is assumed to have an ideal supercapacitor ($f^2(x) = x, f^3(x) = x$). Therefore the non-linearity in the problem comes from the objective-function which is composed of I univariate positive non-linear terms $f^1(x)$. Ensuring a final precision of at least Δ requires to bound f^1 with a tolerance $\delta = \Delta/I$. Even if the same target precision Δ is chosen, different power profiles translate into different δ values. Even two power profiles with the same time horizon I , may require different δ values because different power profiles may translate into solution costs which can differ significantly in order of magnitude, requiring different Δ values, and therefore different δ values.

To summarize, for any new power profile provided, the pwl bounding of the same function f^1 may need to be redone with a value of δ suitable to the new data set, even though the non-linear terms themselves remain unchanged. In addition, since the δ errors on the non linear terms are additive, a longer time horizon means a higher number of univariate terms ($f^1(x_i)$) in equation (1), which means that a smaller tolerance δ may be required. This translates into an increase of the number pieces for the pwl functions, and therefore more binary variables which adds to the complexity of a MILP that was already penalized by the fact that long time horizons meant a large set \mathbb{I} thus more decision variables x_i^1, x_i^2, x_i^3 . These all contribute to a substantial reduction of the size of instances that can be solved efficiently.

5.2 Solution dependence leading to unknown δ values

There exists several cases of MINLP where knowing Δ is not sufficient to infer the corresponding δ values. Let us consider for example problem (CF) modeling a scheduling problem with a single energy source. There are a set of activities \mathcal{A} , each activity a having a release date r_a , a due date d_a , a duration p_a and an instantaneous energy demand b_a . The efficiency function of the energy source used to satisfy the total demand of activities scheduled at each time period is denoted ρ , i.e a cost or energy consumption of $\rho(x)$ produces an amount of usable energy x . Therefore ρ is defined on $[0, \sum_{i \in \mathcal{A}} b_i]$ and verifies $\rho(0) = 0$. The goal is to schedule the tasks so as to minimize the total energy cost. A resulting mathematical model requires binary decision variable x_{at} that is equal to 1 iff activity a is active at instant $t \in \mathcal{T}$.

Continuous variables w_t represent the total energy demand at instant $t \in \mathcal{T}$. In the resulting problem (20)-(24) non-linearity comes from the objective-function which is comprised of $|\mathcal{T}|$ univariate non-linear terms.

$$(CF) \min \sum_{t \in \mathcal{T}} \rho(w_t) \quad // \text{minimize total energy cost} \quad (20)$$

s.t.

$$\sum_{t \in \mathcal{T}} \alpha_{at} x_{at} \geq p_a, \quad \forall a \in \mathcal{A} \quad // \text{execute activity } a \text{ during } p_a \text{ time periods} \quad (21)$$

$$w_t - \sum_{a \in \mathcal{A}} b_a x_{at} = 0, \quad \forall t \in \mathcal{T} \quad // \text{link variables } x_{at} \text{ and } z_t \quad (22)$$

$$x_{at} \in \{0, 1\}, \quad \forall a \in \mathcal{A}, t \in \mathcal{T} \quad // \text{domain definition} \quad (23)$$

$$w_t \in \mathbb{R}^+, \quad \forall t \in \mathcal{T} \quad // \text{domain definition} \quad (24)$$

Let $(CF)_{\Delta}$ refer to the MILP derived from problem (CF) with a target tolerance Δ . Identifying a relevant δ value for obtaining $(CF)_{\Delta}$ is not straightforward. Indeed, in the general case, the duration of the schedule for the optimal solution is not known a priori. The only information available is the time horizon $|\mathcal{T}|$, but this value may be far from the real ending time of the optimal schedule. The “ δ -bounding + MILP solution” methodology requires the computation of a δ -estimator for each single term of the objective-function so as to satisfy the global tolerance value of Δ . Doing so requires each term to be bounded with a precision of $\Delta/|\mathcal{T}|$. Since $|\mathcal{T}|$ may be far from the optimal schedule duration, this can result into unnecessarily small δ values which lead to pwl functions with a higher number of pieces, which translates into more binary variables and MILPs more difficult to solve than necessary given the target tolerance. As an illustration, consider an instance with a time horizon $|\mathcal{T}| = 100$ having an optimal solution with an ending time of 10 (i.e an optimal solution x^* that verifies $\sum_{a \in \mathcal{A}} x_{a,10}^* \geq 1$ and $\sum_{a \in \mathcal{A}} \sum_{t > 10} x_{at}^* = 0$). In this case only 10 terms are active in the objective-function, therefore bounding each term with a tolerance $\delta = \Delta/10$ should have been sufficient to achieve the requested global tolerance Δ . Instead, because the duration is unknown a priori, each term of the objective function has to be bounded with precision $\delta = \Delta/100$. In summary, a pwl bounding tolerance 10 times higher than necessary would be requested, leading to pwl bounding functions with many more line-segments pieces than necessary, leading to a significant increase of the number of binary variables, leading to MILPs more time consuming than necessary to solve given the tolerance target Δ .

6 Using ϵ -relative tolerance

To counter the drawbacks identified in Section 5, we propose to perform the upper and lower bounding of energy conversion expressions using non-necessarily continuous pwl functions with a *relative* ϵ -tolerance.

Definition 6.1 *On a compactum \mathbb{D} , pwl bounding a function $f : \mathbb{D} \rightarrow \mathbb{R}^*$ with a tolerance value $\epsilon \in [0, 100\%]$ consists in identifying two pwl functions $(\overline{f}^{\epsilon}, \underline{f}^{\epsilon})$ that verify equations (25)-(27).*

$$\underline{f}^\epsilon(x) \leq f(x) \leq \overline{f}^\epsilon(x), \quad \forall x \in \mathbb{D} \quad (25)$$

$$|f(x) - \underline{f}^\epsilon(x)| \leq \epsilon|f(x)|, \quad \forall x \in \mathbb{D} \quad (26)$$

$$|\overline{f}^\epsilon(x) - f(x)| \leq \epsilon|f(x)|, \quad \forall x \in \mathbb{D} \quad (27)$$

Definition 6.1 generalizes the one of Ngueveu et al. (2016) by taking into account $\mathbb{D} \subset \mathbb{R}$ instead of $\mathbb{D} \subset \mathbb{R}^+$.

The resulting “ ϵ relative bounding + MILP solution” methodology shares the two advantages of the “ δ absolute bounding + MILP solution” methodology . Firstly, pwl bounding can be applied directly on the univariate non-linear energy conversion function before its insertion into the mathematical model, which may be convex or concave and therefore easier to approximate or bound than the resulting objective or constraint. Secondly, two MILP problems denoted $(\overline{\text{MILP}}^\epsilon)$ and $(\underline{\text{MILP}}^\epsilon)$ can be obtained which provide upper and lower bounds to the original MINLP. But in addition, thanks to the use of relative tolerance, guarantees can be obtained on the quality of resulting bounds under conditions expressed in proposition 6.2. Models and algorithms from Sections 3 and 4 can be adapted to ensure satisfaction of the relative ϵ tolerance constraints (25)-(27) instead of the absolute δ tolerance constraints (16)-(17).

Proposition 6.2 *Let (P) be a (MI)NLP linearly constrained and with an objective-function decomposable into a sum of univariate positive linear or non linear functions, i.e verifying equation (28) where $g_{ki} : \mathbb{D} \rightarrow \mathbb{R}^+, \forall k, \forall i$ are linear or non linear functions. And let \overline{P}^ϵ (resp. \underline{P}^ϵ) be the MILP resulting from the replacement of each non linear term g_{ki} with its over-estimator $\overline{g}_{ki}^\epsilon$ (resp. under-estimator $\underline{g}_{ki}^\epsilon$). Then the solution values of the optimal solutions $z_{(P)}$, $z_{(\underline{P}^\epsilon)}$ and $z_{(\overline{P}^\epsilon)}$ of the corresponding problems verify equations (29).*

$$(P) \min \text{ or } \max \quad z_{(P)} = g(x) = \sum_k \sum_i g_{ki}(x_i) \quad \text{s.t.} \quad Ax \leq B \quad (28)$$

$$(1 - \epsilon)z_{(P)} \leq \max \left\{ z_{(\underline{P}^\epsilon)}, \frac{z_{(\overline{P}^\epsilon)}}{1 + \epsilon} \right\} \leq z_{(P)} \leq \min \left\{ z_{(\overline{P}^\epsilon)}, \frac{z_{(\underline{P}^\epsilon)}}{1 - \epsilon} \right\} \leq (1 + \epsilon)z_{(P)} \quad (29)$$

Proof It results from the combination of equations (25) with $z_{(P)} \leq z_{(\overline{P}^\epsilon)} \leq (1 + \epsilon)z_{(P)}$ and $(1 - \epsilon)z_{(P)} \leq z_{(\underline{P}^\epsilon)} \leq z_{(P)}$ ■

Let $\overline{f}^\epsilon(x)$ (resp. $\underline{f}^\epsilon(x)$) be an optimal ϵ -over- (resp. under-) estimator of f . Contrary to the case of absolute tolerance, if relative tolerance is used then $\overline{f}^\epsilon(x) - \epsilon$ (resp. $\underline{f}^\epsilon(x) + \epsilon$) is not necessarily an under- (resp. over-) estimator of $f(x)$. Therefore is not possible to deduce an optimal ϵ -over- (resp. under-) estimator of f from applying a shift on an optimal ϵ -under- (resp. over-) estimator of f . Each estimator has to be computed separately.

7 Computational results

7.1 Instances, parameter settings and table headings

In order to experimentally evaluate the solution method ϵ PLB+MILP on instances of realistic size, we consider the real-world problem of offline energy optimization for hybrid electric

vehicles. Comparisons to the best known solution methods from the literature are done on adapted instances from the real-world data of Ngueveu et al. (2017). The vehicle characteristics are summarized in table 2. With regards to the mathematical model (1)-(5), f^1 is a given polynomial whereas f^2 and f^3 are linear. Function f^1 is convex on $[1; 7.04029]$ and concave on $[7.04029; 60]$. The six different power demand profiles available ¹ varied in size from $I = 40$ to $I = 1400$.

Parameter	Value or expression
P_{\min}^1	1 kW
P_{\max}^1	60 kW
P_{\min}^2	-60 kW
P_{\max}^2	60kW
E_{\min}	400
E_{\max}	1600 kW
E_0	900 kW
$f^1(x)$	$0.0000002x^5 - 0.0000274x^4 + 0.00151450x^3 - 0.02453270x^2$ $+1.92434870x + 5.90568630$
$f^2(x)$	$x/0.92470$
$f^3(x)$	$-0.930x$

Table 2: HEV characteristics

To the best of our knowledge the best known solution method for the problem was proposed by (Gaoua et al., 2013) and is based on the reformulation of the problem as a MILP based on the use of efficiency points from the original data obtained experimentally for the FC. This approach was replicated by (Chauvin et al., 2015) in an iterative procedure based on the discretization of the non linear continuous efficiency function in equidistant efficiency points and solution of the resulting sub-problems with a MILP solver. Applying such method on problem (P) consists in defining a set \mathcal{K}_{FC} of efficiency points k denoted $(\tilde{x}^k, f^1(\tilde{x}^k))$, defining binary variables y_i^k equal to 1 iff efficiency point k is used at time i , replacing x_i with $\sum_{k \in \mathcal{K}_{FC}} y_i^k$ in the constraints and replacing each non linear term $f^1(x_i)$ of the objective-function with the linear term $\sum_{k \in \mathcal{K}_{FC}} f^1(\tilde{x}^k) y_i^k$. The resulting MILP is denoted $(P_{\mathcal{K}})$. The resulting solution method is denote EP+MILP. The only known lower bound for the problem was proposed by Ngueveu et al. (2017), based an assumption of ideal conditions: that the FC efficiency remains at its maximum level when it is used, i.e. each term $f^1(x)$ is replaced with αx where $\alpha = (\max_{y \in [P_{\min}^1, P_{\max}^1]} \frac{f^1(y)}{y})$. The resulting relaxation was a MILP (P_{ideal}) which could then be solved to optimality.

Applying ϵ PLB+MILP on problem (P) consists in identifying, for a given value of ϵ , two pwl functions $\overline{f^{1^\epsilon}}$ and $\underline{f^{1^\epsilon}}$ verifying equations (25)-(27) to replace f^1 . The two resulting MILPs are denoted (\overline{P}^ϵ) and (\underline{P}^ϵ) . Four different ϵ values are tested: 5%, 1%, 0.5% and 0.1%. Different number of pieces \overline{n}^ϵ and \underline{n}^ϵ of pwl functions $\overline{f^{1^\epsilon}}$ and $\underline{f^{1^\epsilon}}$ are obtained, as illustrated in Tables 3 and 4. To ensure a fair comparison between $(P_{\mathcal{K}})$ and (\overline{P}^ϵ) , the same number

¹Power profiles derived from the real-wold test drive cycles (Ngueveu et al. (2017)) and available at http://homepages.laas.fr/sungueve/Data/HEV_I_PowerProfiles.zip.

of binary variables are imposed by setting $|\mathcal{K}_{FC}| = \bar{n}^\epsilon$ in Tables 5 and 6. The solutions costs of (P_{ideal}) and (\underline{P}^ϵ) are compared in Table 7. The new best known upper and lower bounds are summarized in Table 8. Finally, in Table 9 is computed for each instance, each tolerance value ϵ and the resulting upper bound $UB_{\bar{P}^\epsilon}$, the largest absolute tolerance value $\delta = \frac{UB_{\bar{P}^\epsilon} - \underline{f}^\epsilon}{\underline{f}^\epsilon}$ required to obtain a MILP whose solution cost would be equivalent to $UB_{\bar{P}^\epsilon}$. Table 9 then compares the number of pieces of the resulting pwl functions in order to highlight the differences in sizes of the resulting MILPs.

The PLB algorithms (dichotomic search, heuristic) are programmed with MATLAB R2015b 64bits. The parameter α of the dichotomic search is set to 4. The MILPs $(\bar{P}^\epsilon, \underline{P}^\epsilon, P_{\mathcal{K}}, P_{\text{ideal}})$ are solved with CPLEX 12.6.2 in the single computing thread mode on a desktop computer Intel(R) Xeon(R) 2.90 GHz CPU E3-1271 v3 with 16 GB of RAM.

Results are reported with the following headings:

- DS: dichotomic search-based algorithm from Section 3 to compute $\underline{f}^{1^\epsilon}$ or \bar{f}^{1^ϵ}
- $H^{(1)}, H^{(0.5)}, H^{(0.1)}$: heuristic obtained by replacing steps 4 and 5 of Algorithm 2 with a heuristic trying different decreasing values with a step parameter α ($= 1, 0.5$ or 0.1) to compute $\underline{f}^{1^\epsilon}$ or \bar{f}^{1^ϵ} , following the same logic as the α -Forward Heuristic with Backward Iterations from Rebennack and Kallrath (2015)
- \bar{P}^ϵ (resp. \underline{P}^ϵ): MILP to solve, generated by our ϵ PLB+MILP solution method. It provides an upper bound (resp. lower bound) of problem (P)
- $P_{\mathcal{K}}$: MILP to solve, generated by the EP+MILP solution method from Gaoua et al. (2013), Chauvin et al. (2015). It provides an upper bound of problem (P)
- P_{ideal} : MILP to solve, generated by the lower bounding procedure from Ngueveu et al. (2017). It provides a lower bound of problem (P)
- $\underline{n}_{XXX}^\epsilon$ (resp. \bar{n}_{XXX}^ϵ): number of pieces computed by XXX for \underline{f}^1 (resp. \bar{f}^1) given relative tolerance ϵ
- cpu_{XXX} : computing time for XXX (algorithm or MILP solution)
- $\text{rlb}_{XXX}, \text{UB}_{XXX}$ (resp. LB_{XXX}): results from the solution of MILP XXX in terms of best bound, best feasible solution cost that is an upper bound for (P) (resp. best feasible solution cost that is a lower bound for (P))
- $\bar{n}_{\text{DS}}^\delta$: number of pieces computed by DS for \bar{f}^{1^δ} given absolute tolerance δ
- $\Omega(XXX, YYY)$ relative gap between values XXX and YYY computed with equation (30)

$$\Omega(XXX, YYY) = 100 * \frac{XXX - YYY}{YYY} \% \quad (30)$$

The remainder of this section is organised in three subsections. Algorithms to perform the first phase (PLB) of our ϵ PLB+MILP method are analyzed in subsection 7.2. The results of the second phase and comparisons to the state-of-the-art are explored in subsection 7.3. A study of the impact of the tolerance type (relative versus absolute) is done in subsection 7.4.

7.2 Results of the first phase: the piecewise linear bounding of f^1

Tables 3 and 4 report for each value of ϵ the number of pieces and computing time for upper and lower bounding the non linear energy conversion function f^1 . According to equation (19) the difference between $\underline{n}_{DS}^\epsilon$ (resp. \bar{n}_{DS}^ϵ) and the optimum number of pieces of $\underline{f}^{1^\epsilon}$ (resp. \bar{f}^{1^ϵ}) is at most one. Results show that the heuristic can much faster than the dichotomic search, but can also obtain much more pieces if the step value chosen is too large. Finding the right step value a priori is not straightforward and the higher number of pieces would lead to larger MILPs to be solved in the second phase. For example, the two extra pieces for $\epsilon = 0.1\%$ lead to $2I$ additional binary variables, which corresponds to an additional number of binary variables from 80 to 2800 on the given instances. Obviously the computing times of the PLB algorithms reported in tables 3 and 4 would be reduced if all algorithms were implemented in C or C++, but the relative differences would remain the same and the conclusion would not differ.

ϵ	Dichotomic Search		Heuristic with different step values (1, 0.5, 0.1)					
	$\underline{n}_{DS}^\epsilon$	cpu _{DS}	$\underline{n}_{H(1)}^\epsilon$	cpu _{H(1)}}	$\underline{n}_{H(0.5)}^\epsilon$	cpu _{H(0.5)}}	$\underline{n}_{H(0.1)}^\epsilon$	cpu _{H(0.1)}}
5.0 %	4	13.2 s	4	0.95 s	4	1.46 s	4	4.64 s
1.0 %	6	32.18 s	7	1.95 s	6	2.81 s	6	10.38 s
0.5 %	9	41.59 s	10	2.6 s	10	3.84 s	9	15.62 s
0.1 %	17	94.06 s	29	7.9 s	22	9.42 s	19	33.93 s
average	9.0	45.26 s	12.5	3.35 s	10.5	4.38 s	9.5	16.14 s

Table 3: Comparison of pwl ϵ -underestimators from DS, $H^{(1)}$, $H^{(0.5)}$ and $H^{(0.1)}$

ϵ	Dichotomic Search		Heuristic with different step values (1, 0.5, 0.1)					
	\bar{n}_{DS}^ϵ	cpu _{DS}	$\bar{n}_{H(1)}^\epsilon$	cpu _{H(1)}}	$\bar{n}_{H(0.5)}^\epsilon$	cpu _{H(0.5)}}	$\bar{n}_{H(0.1)}^\epsilon$	cpu _{H(0.1)}}
5.0 %	4	10.58 s	4	0.99 s	4	1.3 s	4	4.29 s
1.0 %	6	22.07 s	7	2.01 s	7	2.97 s	6	10.45 s
0.5 %	8	46.81 s	11	3.25 s	10	4.44 s	9	14.99 s
0.1 %	17	73.17 s	29	8.72 s	24	10.62 s	19	33.87 s
average	8.8	38.16 s	12.8	3.74 s	11.3	4.83 s	9.5	15.9 s

Table 4: Comparison of pwl ϵ -overestimators from DS, $H^{(1)}$, $H^{(0.5)}$ and $H^{(0.1)}$

7.3 Results of the second phase: the solution of (\bar{P}^ϵ) and \bar{MILP}

In this subsection we investigate ϵ PLB+MILP in comparison to best known upper and lower bounding procedures from the literature. A time limit of 3600 s was set and the table of result shows “- s” for instances where this time limit was reached. The default stopping criterion of CPLEX with regards to the relative gap was left unchanged (0.01%). The table of results shows “<0.01%” for instances where it was the reason for stopping.

Table 5 compares (\bar{P}^ϵ) and $(P_{\mathcal{K}})$. Each MILP provides four outputs: the best lower bound (rlb), the upper bound (UB), the relative gap between both bounds ($\Omega(\text{UB},\text{rlb})$) and the computing time (cpu). Column UB^r displays the real total cost of the solution from (\bar{P}^ϵ) , obtained by using f^1 instead of \bar{f}^{1^ϵ} to recompute the cost of the solution provided by (\bar{P}^ϵ) . By definition, the recomputed cost can be either equal or up to $\epsilon\%$ less than the original cost. Such recomputation would leave unchanged the cost of $(P_{\mathcal{K}})$'s solutions because the efficiency points are belong to the set $(\tilde{x}^k, f^1(\tilde{x}^k))$, therefore already verify f^1 . The results from Table 5 show that 18 instances of out 24 for $(P_{\mathcal{K}})$ reached the time limit while none from the (\bar{P}^ϵ) did. 19 instances out of 24 for the (\bar{P}^ϵ) and 2 instances for the $(P_{\mathcal{K}})$ stopped because of the relative gap limit. Finally, a systematic improvement of the solution cost is obtained after recomputation.

Instance	ϵ	\bar{n}^ϵ	$\epsilon\text{PLB}+\text{MILP}$				$\text{EP}+\text{MILP}$					
			$\text{rlb}_{\bar{P}^\epsilon}$	$\text{UB}_{\bar{P}^\epsilon}$	$\Omega(\text{UB},\text{rlb})$	$\text{cpu}_{\bar{P}^\epsilon}$	$\text{UB}_{\bar{P}^\epsilon}^r$	\mathcal{K}	$\text{rlb}_{P_{\mathcal{K}}}$	$\text{UB}_{P_{\mathcal{K}}}$	$\Omega(\text{UB},\text{rlb})$	$\text{cpu}_{P_{\mathcal{K}}}$
S_40	5.0 %	4	462.7	462.7	0.00 %	< 1 s	461.6	4	469.7	487.5	3.67 %	- s
	1.0 %	6	454.3	454.3	< 0.01 %	< 1 s	453.4	6	460.2	460.3	0.01 %	1 s
	0.5 %	8	454.8	454.8	< 0.01 %	< 1 s	454.4	8	457.2	457.3	0.01 %	< 1 s
	0.1 %	17	453.6	453.6	< 0.01 %	< 1 s	453.5	17	454.6	454.6	0.01 %	< 1 s
I_561	5.0 %	4	8885.7	8885.7	0.00 %	4 s	8826.8	4	8878.4	8880.2	0.02 %	- s
	1.0 %	6	8756.3	8756.4	< 0.01 %	6 s	8741.9	6	8794.5	8798.3	0.04 %	- s
	0.5 %	8	8765.3	8765.5	< 0.01 %	7 s	8758.7	8	8768.7	8770.6	0.02 %	- s
	0.1 %	17	8742.5	8742.5	0.00 %	8 s	8741.4	17	8748.2	8748.8	0.01 %	- s
U_811	5.0 %	4	2699.7	2699.9	< 0.01 %	27 s	2697.1	4	2635.2	2638.4	0.12 %	- s
	1.0 %	6	2613.5	2613.5	< 0.01 %	13 s	2607.7	6	2633.3	2635.6	0.09 %	- s
	0.5 %	8	2618.3	2618.3	0.00 %	10 s	2615.6	8	2616.4	2618.9	0.09 %	- s
	0.1 %	17	2608.3	2608.5	< 0.01 %	1105 s	2608.2	17	2610.3	2611.4	0.04 %	- s
H_734	5.0 %	4	19185.6	19185.6	0.00 %	9 s	18679.7	4	18705.6	18710.0	0.02 %	- s
	1.0 %	6	18625.3	18626.4	< 0.01 %	13 s	18578.8	6	18767.0	18770.3	0.02 %	- s
	0.5 %	8	18635.1	18635.2	< 0.01 %	12 s	18597.1	8	18628.0	18629.0	0.01 %	434 s
	0.1 %	17	18575.9	18576.3	< 0.01 %	21 s	18571.7	17	18592.1	18593.7	0.01 %	- s
N_1200	5.0 %	4	23892.6	23892.8	< 0.01 %	26 s	23592.3	4	23303.0	23313.1	0.04 %	- s
	1.0 %	6	23136.8	23138.9	< 0.01 %	43 s	23130.4	6	23422.7	23427.6	0.02 %	- s
	0.5 %	8	23172.1	23173.1	< 0.01 %	59 s	23151.7	8	23151.2	23154.4	0.01 %	- s
	0.1 %	17	23119.3	23119.8	< 0.01 %	49 s	23119.3	17	23139.4	23139.9	< 0.01 %	702 s
E_1400	5.0 %	4	27302.0	27303.5	< 0.01 %	64 s	27301.5	4	27506.2	27529.9	0.09 %	- s
	1.0 %	6	27087.5	27087.8	< 0.01 %	68 s	27075.1	6	27248.1	27252.2	0.01 %	- s
	0.5 %	8	27098.3	27098.7	< 0.01 %	61 s	27091.8	8	27130.1	27135.3	0.02 %	- s
	0.1 %	17	27076.3	27076.3	< 0.01 %	76 s	27075.4	17	27084.6	27084.7	< 0.01 %	127 s
average				< 0.01 %	70 s					0.18 %	2580 s	

Table 5: Upper bounds from $\epsilon\text{PLB}+\text{MILP}$ versus $\text{EP}+\text{MILP}$

Table 6 gives a clearer picture of the upper bounding results by comparing relative gaps. The first column shows that on average, recomputing the solution cost of (\bar{P}^ϵ) leads to a improvement of 0.27% of its value. The next column shows the gaps between the upper bounds from (\bar{P}^ϵ) and $(P_{\mathcal{K}})$. For high ϵ values, it is possible for $(P_{\mathcal{K}})$ to be better than (\bar{P}^ϵ) . This can be explained by the fact that efficiency points are chosen on the curve $(x, f^1(x))$ whereas the overestimation \bar{f}^{1^ϵ} can be up to $\epsilon\%$ far from f^1 . However, when ϵ reduces, (\bar{P}^ϵ) becomes better than $(P_{\mathcal{K}})$. On average there is a 0.48% gap in favor of (\bar{P}^ϵ) . The last column compares the recomputed upper bound from (\bar{P}^ϵ) with the lower bound from $(P_{\mathcal{K}})$. The goal is to estimate the minimum predictable gap if $(P_{\mathcal{K}})$ was solved to optimality, for example if the time limit was increased. It shows an average value of 0.30%, meaning that the average gap in terms of solutions costs between $\epsilon\text{PLB}+\text{MILP}$ and $\text{EP}+\text{MILP}$ vary between 0.30% and

0.48% in favor of ϵ PLB+MILP.

Instance	ϵ	\bar{n}^ϵ	ϵ PLB+MILP	ϵ PLB+MILP vs EP+MILP	
			$\Omega(\text{UB}_{\underline{P}^\epsilon}^r, \text{UB}_{\underline{P}^\epsilon})$	$\Omega(\text{UB}_{\underline{P}^\epsilon}^r, \text{UB}_{(P_\kappa)})$	$\Omega(\text{UB}_{\underline{P}^\epsilon}^r, \text{rlb}_{(P_\kappa)})$
S_40	5.0 %	4	-0.24 %	-5.32 %	-1.72 %
	1.0 %	6	-0.20 %	-1.49 %	-1.48 %
	0.5 %	8	-0.07 %	-0.62 %	-0.61 %
	0.1 %	17	-0.02 %	-0.25 %	-0.24 %
I_561	5.0 %	4	-0.66 %	-0.60 %	-0.58 %
	1.0 %	6	-0.16 %	-0.64 %	-0.60 %
	0.5 %	8	-0.08 %	-0.14 %	-0.11 %
	0.1 %	17	-0.01 %	-0.09 %	-0.08 %
U_811	5.0 %	4	-0.10 %	2.23 %	2.35 %
	1.0 %	6	-0.22 %	-1.06 %	-0.97 %
	0.5 %	8	-0.10 %	-0.13 %	-0.03 %
	0.1 %	17	-0.01 %	-0.12 %	-0.08 %
H_734	5.0 %	4	-2.64 %	-0.16 %	-0.14 %
	1.0 %	6	-0.26 %	-1.02 %	-1.00 %
	0.5 %	8	-0.20 %	-0.17 %	-0.17 %
	0.1 %	17	-0.02 %	-0.12 %	-0.11 %
N_1200	5.0 %	4	-1.26 %	1.20 %	1.24 %
	1.0 %	6	-0.04 %	-1.27 %	-1.25 %
	0.5 %	8	-0.09 %	-0.01 %	0.00 %
	0.1 %	17	0.00 %	-0.09 %	-0.09 %
E_1400	5.0 %	4	-0.01 %	-0.83 %	-0.74 %
	1.0 %	6	-0.05 %	-0.65 %	-0.64 %
	0.5 %	8	-0.03 %	-0.16 %	-0.14 %
	0.1 %	17	0.00 %	-0.03 %	-0.03 %
average			-0.27 %	-0.48 %	-0.30 %

Table 6: Relative gaps from solution methods ϵ PLB+MILP and EP+MILP

Table 7 compares (P_{ideal}) and (\underline{P}^ϵ) . For each MILP two outputs are considered: the optimal solution value which provides a lower bound (LB) for (P) and the computing time (cpu). The last column shows the relative gap between the bounds, a positive value meaning that our ϵ PLB+MILP method improved the best known lower bound for the problem. The results obtained show that (\underline{P}^ϵ) require a higher computing time than (P_{ideal}) , which was expected since one has I binary variables whereas the other has $\bar{n}^\epsilon I$ binary variables. Also, for high values of ϵ (\underline{P}^ϵ) can provide worse bounds than (P_{ideal}) , which can be explained by the fact that the underestimation \bar{f}^{1^ϵ} can be up to $\epsilon\%$ far from f^1 . However, for all $\epsilon \leq 1\%$ the (\underline{P}^ϵ) was better than (P_{ideal}) , allowing ϵ PLB+MILP to produce the now best known lower bounds of problem (P) .

Table 8 presents the best known upper and lower bounds resulting from ϵ PLB+MILP. Because of equations (25)-(27) a second lower bound LB2 can be deduced from upper bound

Instance	literature		ϵ PLB+MILP				Gap
	$LB_{P_{ideal}}$	$cpu_{P_{ideal}}$	ϵ	n^ϵ	$LB_{\underline{P}^\epsilon}$	$cpu_{\underline{P}^\epsilon}$	$\Omega(LB_{\underline{P}^\epsilon}, LB_{P_{ideal}})$
S_40	444.1	< 1 s	5.0 %	4	439.0	< 1 s	-1.15 %
			1.0 %	6	449.9	< 1 s	1.30 %
			0.5 %	9	452.5	< 1 s	1.90 %
			0.1 %	17	453.1	< 1 s	2.04 %
I_561	8626.2	< 1 s	5.0 %	4	8433.1	4 s	-2.24 %
			1.0 %	6	8668.4	5 s	0.49 %
			0.5 %	9	8720.5	5 s	1.09 %
			0.1 %	17	8733.8	8 s	1.25 %
U_811	2532.0	2 s	5.0 %	4	2559.6	10 s	1.09 %
			1.0 %	6	2587.6	-	2.20 %
			0.5 %	9	2605.2	10 s	2.89 %
			0.1 %	17	2605.7	2762 s	2.91 %
H_734	18323.7	2s	5.0 %	4	18199.5	8s	-0.68 %
			1.0 %	6	18437.7	10 s	0.62 %
			0.5 %	9	18541.7	12 s	1.19 %
			0.1 %	17	18557.3	17 s	1.27 %
N_1200	22356.2	6 s	5.0 %	4	22663.8	45 s	1.38 %
			1.0 %	6	22905.6	39 s	2.46 %
			0.5 %	9	23056.5	29 s	3.13 %
			0.1 %	17	23096.2	50 s	3.31 %
E_1400	26407.3	9 s	5.0 %	4	25923.4	41 s	-1.83 %
			1.0 %	6	26817.8	65 s	1.55 %
			0.5 %	9	26960.5	52 s	2.09 %
			0.1 %	17	27049.2	120 s	2.43 %
average		3 s			143 s	1.28 %	

Table 7: Lower bounds from ϵ PLB+MILP versus literature

$UB_{\overline{P}^\epsilon}$ using relation $LB2 = \frac{UB_{\overline{P}^\epsilon}}{1+\epsilon}$. Likewise, a second upper bound $UB2$ can be deduced from lower bound $LB_{\underline{P}^\epsilon}$ using relation $UB2 = \frac{LB_{\underline{P}^\epsilon}}{1-\epsilon}$. The best lower (resp. upper) bound from ϵ PLB+MILP is the maximum (resp. minimum) of the two lower (resp. upper) bounds. Results show that the final gap between the two bounds is as expected lower or equal to $\epsilon\%$.

7.4 Comparison of relative and absolute tolerance for piecewise linear bounding

This section focuses on the impact of type of tolerance (relative or absolute) used for the first phase of the PLB+MILP solution method. For each pair “instance, epsilon”, given the resulting cost of the feasible solution obtained by \overline{P} , we compute what would have been the value of absolute tolerance δ to be used for pwl bounding with absolute tolerance in order to obtain an equivalent solution cost. It is computed as $\delta = \frac{UB^*\epsilon}{I}$. The value computed is used

Instance	ϵ	Upper bounds		Lower bounds		Gap $\Omega(\text{UB}_{\text{best}}, \text{LB}_{\text{best}})$
		$\text{UB}_{\overline{P}^\epsilon}$	$\text{UB2} = \frac{\text{LB}_{\overline{P}^\epsilon}}{1-\epsilon}$	$\text{LB}_{\overline{P}^\epsilon}$	$\text{LB2} = \frac{\text{UB}_{\overline{P}^\epsilon}}{1+\epsilon}$	
S_40	5.00 %	461.60	462.06	438.96	439.62	5.00 %
	1.00 %	453.43	454.40	449.86	448.94	0.79 %
	0.50 %	454.44	454.76	452.49	452.18	0.43 %
	0.10 %	453.48	453.56	453.10	453.03	0.08 %
I_561	5.00 %	8826.80	8876.91	8433.06	8406.47	4.67 %
	1.00 %	8741.92	8755.92	8668.36	8655.37	0.85 %
	0.50 %	8758.73	8764.32	8720.49	8715.15	0.44 %
	0.10 %	8741.39	8742.53	8733.79	8732.65	0.09 %
U_811	5.00 %	2697.13	2694.28	2559.57	2568.69	4.89 %
	1.00 %	2607.73	2613.75	2587.61	2581.91	0.78 %
	0.50 %	2615.57	2618.29	2605.20	2602.55	0.40 %
	0.10 %	2608.19	2608.32	2605.72	2605.59	0.09 %
H_734	5.00 %	18679.75	19157.36	18199.49	17790.23	2.64 %
	1.00 %	18578.84	18623.98	18437.74	18394.89	0.77 %
	0.50 %	18597.12	18634.88	18541.70	18504.59	0.30 %
	0.10 %	18571.66	18575.87	18557.30	18553.10	0.08 %
N_1200	5.00 %	23592.33	23856.59	22663.76	22468.88	4.10 %
	1.00 %	23130.40	23137.01	22905.64	22901.39	0.98 %
	0.50 %	23151.73	23172.37	23056.50	23036.54	0.41 %
	0.10 %	23119.28	23119.35	23096.23	23096.18	0.10 %
E_1400	5.00 %	27301.52	27287.82	25923.43	26001.44	4.95 %
	1.00 %	27075.06	27088.65	26817.76	26806.99	0.96 %
	0.50 %	27091.81	27095.98	26960.50	26957.02	0.49 %
	0.10 %	27075.40	27076.30	27049.23	27048.36	0.10 %
average	1.65 %					1.43 %

Table 8: Best bounds from $\epsilon\text{PLB}+\text{MILP}$, with $\text{UB}_{\text{best}} = \min(\text{UB}_{\overline{P}^\epsilon}, \text{UB2})$ and $\text{LB}_{\text{best}} = \max(\text{LB}_{\overline{P}^\epsilon}, \text{LB2})$

to perform the pwl over-estimator $\overline{f^1}^\delta$ and the resulting number of pieces \overline{n}^δ is presented. Table 9 reports the results obtained.

Results show that the δ value varies from on instance to another: for example for $\epsilon = 1\%$, the δ values vary between 0.032 and 0.254; consequently, the number of pieces vary, here between 10 and 25. Overall, on average the absolute δ tolerance led to more than twice the number of pieces than the relative ϵ tolerance. This can be expected to translate into larger MILPs with a higher number of binary variables and therefore more difficult to solve in the second phase of $\text{PLB}+\text{MILP}$ applied on problem (P) if absolute tolerance was used, only to reach solutions of similar quality as with relative tolerance.

Instance	I	using relative tolerance ϵ				using absolute tolerance δ			Gap $\Omega(\bar{n}^\delta, \bar{n}^\epsilon)$
		ϵ	\bar{n}^ϵ	cpu _{DS} $^\epsilon$	UB _{MILP} $^\epsilon$	$\delta = \frac{UB^* \epsilon}{I}$	\bar{n}^δ	cpu _{DS} $^\delta$	
S_40	40	5.00 %	4	13.20 s	462.72	0.578	7	21.27 s	75.0 %
		1.00 %	6	32.18 s	454.33	0.114	14	61.47 s	133.3 %
		0.50 %	8	41.59 s	454.77	0.057	19	81.45 s	137.5 %
		0.10 %	17	94.06 s	453.55	0.011	41	205.47 s	141.2 %
I_561	561	5.00 %	4	13.20 s	8885.73	0.792	6	28.28 s	50.0 %
		1.00 %	6	32.18 s	8756.36	0.156	12	55.67 s	100.0 %
		0.50 %	8	41.59 s	8765.48	0.078	16	80.90 s	100.0 %
		0.10 %	17	94.06 s	8742.46	0.016	35	155.06 s	105.9 %
U_811	811	5.00 %	4	13.20 s	2699.92	0.166	12	49.16 s	200.0 %
		1.00 %	6	32.18 s	2613.50	0.032	25	125.12 s	316.7 %
		0.50 %	8	41.59 s	2618.26	0.016	35	161.68 s	337.5 %
		0.10 %	17	94.06 s	2608.51	0.003	83	374.64 s	388.2 %
H_734	734	5.00 %	4	13.20 s	19185.64	1.307	5	21.48 s	25.0 %
		1.00 %	6	32.18 s	18626.42	0.254	10	46.31 s	66.7 %
		0.50 %	8	41.59 s	18635.21	0.127	13	65.06 s	62.5 %
		0.10 %	17	94.06 s	18576.26	0.025	28	126.18 s	64.7 %
N_1200	1200	5.00 %	4	13.20 s	23892.82	0.996	6	24.41 s	50.0 %
		1.00 %	6	32.18 s	23138.94	0.193	11	42.43 s	83.3 %
		0.50 %	8	41.59 s	23173.09	0.097	15	67.63 s	87.5 %
		0.10 %	17	94.06 s	23119.80	0.019	32	155.21 s	88.2 %
E_1400	1400	5.00 %	4	13.20 s	27303.47	0.975	6	19.57 s	50.0 %
		1.00 %	6	32.18 s	27087.75	0.193	11	57.73 s	83.3 %
		0.50 %	8	41.59 s	27098.73	0.097	15	66.59 s	87.5 %
		0.10 %	17	94.06 s	27076.32	0.019	32	146.72 s	88.2 %
average		1.65 %	8.75	45.26 s		0.263	20.38	93.31 s	121.8 %

Table 9: Comparison of relative pwl bounding and absolute pwl bounding

8 Conclusion

The paper presents a two-phase solution method for combinatorial optimization problems involving non linear energy conversion functions. The first phase consists in bounding the non linear univariate functions from above and below with two piecewise linear non-necessarily continuous functions with a relative ϵ tolerance. The second phase consists in solving the two mixed integer linear programs obtained when replacing the non linear terms with their piecewise linear overestimators or underestimators. Models and algorithms to perform the piecewise linear bounding are presented. Computational results on an energy optimization problem for hybrid electric vehicles show the efficiency of the ϵ PLB+MILP solution method.

9 Acknowledgements

This research benefited from the support of the FMJH Program PGM0, from the support of EDF-Thales-Orange and from the support of the PEPS program from the *Cellule Energie* of the CNRS.

References

- Borghetti, A., D'Ambrosio, C., Lodi, A., and Martello, S. (2008). An milp approach for short-term hydro scheduling and unit commitment with head-dependent reservoir. *Power Systems, IEEE Transactions on*, 23(3):1115–1124.
- Boukouvala, F., Misener, R., and Floudas, C. A. (2016). Global optimization advances in mixed-integer nonlinear programming, minlp, and constrained derivative-free optimization, cdf. *European Journal of Operational Research*, 252(3):701 – 727.
- Camponogara, E., de Castro, M., and Plucenio, A. (2007). Compressor scheduling in oil fields: A piecewise-linear formulation. In *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on*, pages 436–441.
- Chauvin, A., Hijazi, A., Bideaux, E., and Sari, A. (2015). Combinatorial approach for sizing and optimal energy management of hev including durability constraints. In *24th International Symposium on Industrial Electronics (ISIE)*, pages 1236–1241. IEEE.
- DAmbrosio, C., Lodi, A., and Martello, S. (2010). Piecewise linear approximation of functions of two variables in milp models. *Oper. Res. Lett.*, 38(1):39–46.
- Dunham, J. G. (1986). Optimum uniform piecewise linear approximation of planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):67–75.
- Frenzen, C., Sasao, T., and Butler, J. T. (2010). On the number of segments needed in a piecewise linear approximation. *Journal of Computational and Applied Mathematics*, 234(2):437 – 446.
- Gaoua, Y., Caux, S., and Lopez, P. (2013). Energy management for an electric vehicle based on combinatorial modeling. In *Industrial Engineering and Systems Management (IESM), Proceedings of 2013 International Conference on*, pages 1–6. IEEE.
- Geißler, B., Martin, A., Morsi, A., and Schewe, L. (2012). *Using Piecewise Linear Functions for Solving MINLPs*, pages 287–314. Springer New York, New York, NY.
- Gritzali, F. and Papakonstantinou, G. (1983). A fast piecewise linear approximation algorithm. *Signal Processing*, 5(3):221 – 227.
- Ngueveu, S. U., Artigues, C., and Lopez, P. (2016). Scheduling under a non-reversible energy source: An application of piecewise linear bounding of non-linear demand/cost functions. *Discrete Applied Mathematics*, 208:98–113.

- Ngueveu, S. U., Caux, S., Messine, F., and Guemri, M. (2017). Heuristics and lower bound for energy management in hybrid-electric vehicles. *4OR - A Quarterly Journal of Operations Research*, doi:10.1007/s10288-017-0343-5.
- Papakonstantinou, G., Tsanakas, P., and Manis, G. (1994). Parallel approaches to piecewise linear approximation. *Signal Processing*, 37(3):415 – 423.
- Rebennack, S. and Kallrath, J. (2015). Continuous piecewise linear delta-approximations for univariate functions: Computing minimal breakpoint systems. *J. Optim. Theory Appl.*, 167(2):617–643.
- Rosen, J. B. and Pardalos, P. M. (1986). Global minimization of large-scale constrained concave quadratic problems by separable programming. *Mathematical Programming*, 34(2):163–174.
- Tomek, I. (1974a). Piecewise-linear approximation with a bound on absolute error. *Computers and Biomedical Research*, 7(1):64 – 70.
- Tomek, I. (1974b). Two algorithms for piecewise-linear continuous approximation of functions of one variable. *IEEE Trans. Comput.*, 23(4):445–448.