Piecewise linear bounding of univariate nonlinear functions and resulting MILP-based solution methods

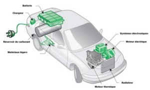
Sandra Ulrich NGUEVEU

Institut National Polytechnique de Toulouse - Laboratoire LAAS-CNRS
ngueveu@laas.fr

OR@DII Seminar - Brescia - 23/10/2019

Outlook

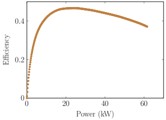
Energy in hybrid electric vehicles

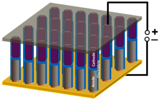


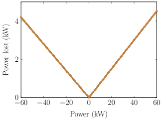
Electric propulsion motor powered by :

- onboard generator : e.g. hydrogen fuel cell (FC)
- reversible source : e.g. supercapacitator (SE)

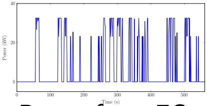
Energy sources characteristics : power limits(kW), efficiency(%), capacity(kWs) ...





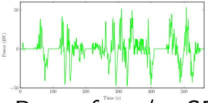


Find at each instant the optimal power split between the energy sources to minimize the total fuel consumption.



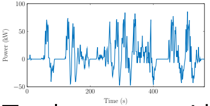
Power from FC

+



Power from/to SE

=



Total power provided

Mathematical model

$$\min \sum_{i=1}^{n} f^{\mathrm{FC}}(x_i) \tag{1}$$

s.t. Power demand satisfaction

$$x_i + y_i - z_i \geq P_i, \qquad \forall i \in \mathbb{I} \tag{2}$$

Final SE energy level higher or equal to initial energy level

$$\sum_{i=1}^{n} f^{\mathrm{SE}^+}(y_i) - f^{\mathrm{SE}^-}(z_i) \leq 0 \tag{3}$$

SE energy level within bounds

$$E_0^{\mathrm{SE}} - E_{\max}^{\mathrm{SE}} \leq \sum_{k=1}^{i} f^{\mathrm{SE}^+}(y_k) - f^{\mathrm{SE}^-}(z_k) \leq E_0^{\mathrm{SE}} - E_{\min}^{\mathrm{SE}}, \qquad \forall i \in \mathbb{I} \tag{4}$$

Variables domains

$$x_i \in \{0\} \cup [P_{\min}^{\mathrm{FC}}, P_{\max}^{\mathrm{FC}}], y_i \in [0, P_{\max}^{\mathrm{SE}}], z_i \in [0, P_{\min}^{\mathrm{SE}}], \qquad \forall i \in \mathbb{I} \tag{5}$$

Water pumping and desalination process

Electrical model

- V_m, I_m : electrical tension, current
- T_m : motor electromag. torque
- Ω : rotation speed
- k_Φ : torque equivalent coefficient
- r : stator resistance

Electric motor equations (inertia neglected) :

$$V_m = rI_m + k_\Phi \Omega \quad (6)$$

$$T_m = \Phi_m I_m \quad (7)$$

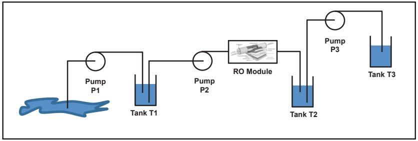
Electrical power needed : $P_e = V_m I_m$.

Pressure drop in the pipe

- $\Delta \mathrm{Pipe}, \rho$: pressure drop,water density
- h : height of water pumping

Static+Dynamic pressure

$$\Delta \mathrm{Pipe} = kq^2 + \rho g h \quad (8)$$



Mechanical-Hydraulic conv.

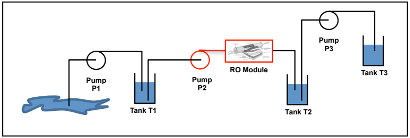
- P_p : output pressure
- q : debit of water
- a, b : non linear girator coefs
- c : hydraulic friction
- p_0 : suction pressure
- $s_p + s_m$: mechanical losses

Static equations of the motor-pump (mechanical inertia neglected) :

$$P_p = (a\Omega + bq)\Omega - (cq^2 + p_0) \quad (9)$$

$$T_m = (a\Omega + bq)q + (s_m + s_p)\Omega \quad (10)$$

Efficiency function of pump 2 + RO module



Subsystem pump 2 + Reverse Osmosis module is modeled with

$$\mathbf{P_e} = r*\mathcal{K}(\mathbf{q},\mathbf{h})+((s_m+s_p)*\Omega(\mathbf{q},\mathbf{h})+(\mathbf{q}+\mathcal{F}(\mathbf{q})/R_{\mathrm{Me}})*\mathcal{M}(\mathbf{q},\mathbf{h}))*\Omega(\mathbf{q},\mathbf{h})$$

where

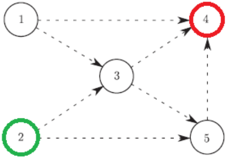
$$\begin{cases} \mathcal{F}(\mathbf{q}) = (R_{\mathrm{Mod}} + R_{\mathrm{Valve}}) * \mathbf{q}^2 \\ \mathcal{G}(\mathbf{q}) = (b * (\mathbf{q} + \mathcal{F}(\mathbf{q})/R_{\mathrm{Me}})) \\ \mathcal{M}(\mathbf{q},\mathbf{h}) = a * \Omega(\mathbf{q},\mathbf{h}) + \mathcal{G}(\mathbf{q}) \\ \Omega(\mathbf{q},\mathbf{h}) = \frac{-\mathcal{G}(\mathbf{q})+\sqrt{\mathcal{G}(\mathbf{q})^2-4a*(-(p_0+\rho g*(\mathbf{h}-l_{\mathrm{out}})+(k+c)*((\mathbf{q}+\mathcal{F}(\mathbf{q})/R_{\mathrm{Me}})^2)+\mathcal{F}(\mathbf{q})))}}{2*a} \\ \mathcal{K}(\mathbf{q},\mathbf{h}) = (((s_m + s_p) * \Omega(\mathbf{q},\mathbf{h}) + (\mathbf{q} + \mathcal{F}(\mathbf{q})/R_{\mathrm{Me}}) * (a * \Omega(\mathbf{q},\mathbf{h}) + \mathcal{G}(\mathbf{q})))/k_\phi)^2 \end{cases}$$

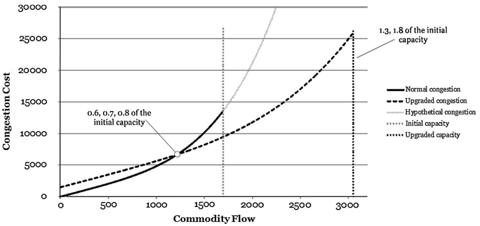
Multicommodity network design problem with congestion

Minimize Total cost = design cost + routing cost + capacity augmentation cost + congestion cost

s.t.

- Flow conservation
- Maximum capacity (with/without upgrade)
- all commodities get to destination





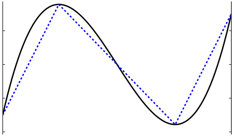
Paraskevopoulos, Sinan Gürel and Bektas, 2016

Classical MINLP solution methods

MILP-based solution methods on similar problems

Camponogara *et al.* 2011 ; Borghetti *et al.*, 2008

- approximate with piecewise linear functions



+ (more) tractable problems
- try and error approach : No guarantees on the solution quality or iterative process with an undefined number of iterations
- global optimality cannot be guaranteed

Generic MINLP solution methods / Hybrid algorithms and frameworks

Grossmann 2002 / Bonami *et al.*, 2008

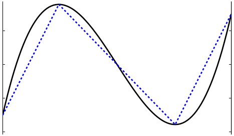
+ global optimality guaranteed if carried out to completion
- only for small/medium instances

Classical MINLP solution methods

MILP-based solution methods on similar problems

Camponogara *et al.* 2011 ; Borghetti *et al.*, 2008

- approximate with piecewise linear functions



+ (more) tractable problems
- try and error approach : No guarantees on the solution quality or iterative process with an undefined number of iterations
- global optimality cannot be guaranteed

Generic MINLP solution methods / Hybrid algorithms and frameworks

Grossmann 2002 / Bonami *et al.*, 2008

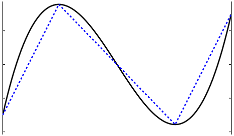
+ global optimality guaranteed if carried out to completion
- only for small/medium instances

Classical MINLP solution methods

MILP-based solution methods on similar problems

Camponogara *et al.* 2011 ; Borghetti *et al.*, 2008

- approximate with piecewise linear functions



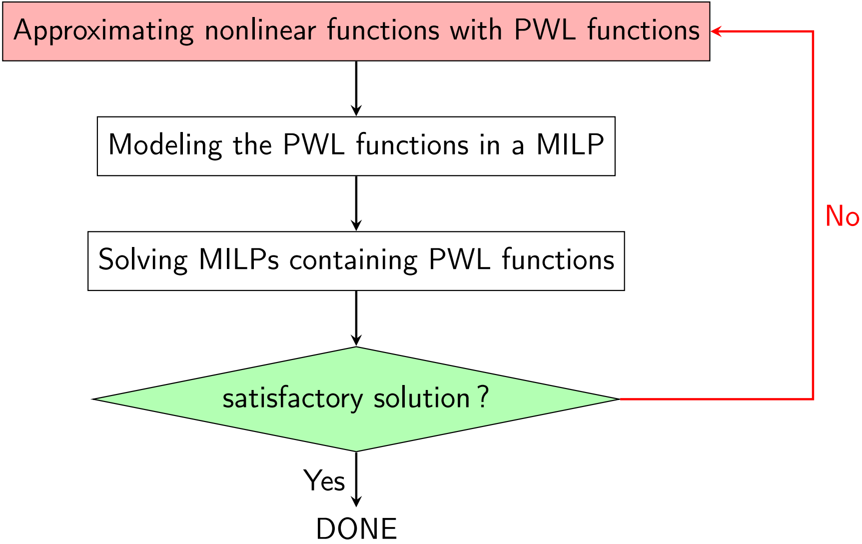
+ (more) tractable problems
- try and error approach : No guarantees on the solution quality or iterative process with an undefined number of iterations
- global optimality cannot be guaranteed

Generic MINLP solution methods / Hybrid algorithms and frameworks

Grossmann 2002 / Bonami *et al.*, 2008

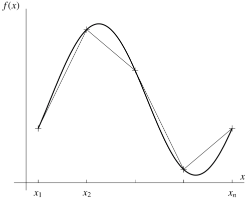
+ global optimality guaranteed if carried out to completion
- only for small/medium instances

MILP-based solution methods for MINLPs



Approximating nonlinear functions with PWL functions

Sampling

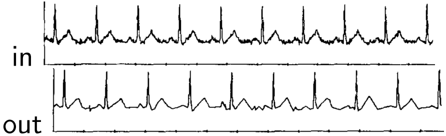


Given a number of breakpoints on the curve, find the PWL functions which minimizes an error metric.

... D'Ambrosio et al 2010 ... D'Ambrosio et al 2015 ...

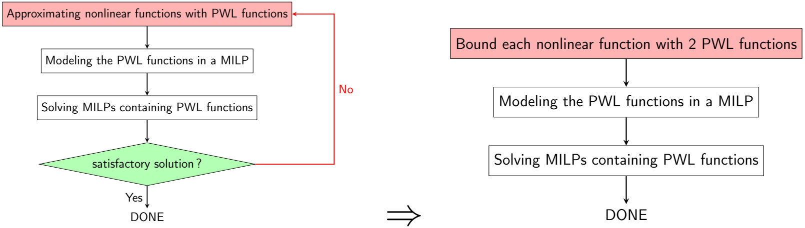
Fitting a discrete data set

Applications : ECG, pattern recognition, data reduction, ...



Bellman and Roth 1969 ... Tomek 1974 ... O'Rourke 1981 ... Toriello and Vielma 2012 ... Rebennack and Krasko 2019

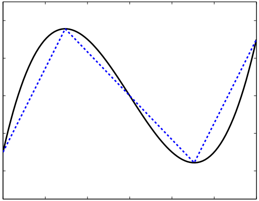
From an iterative to a non-iterative solution method ?



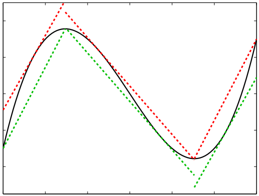
New two-step solution scheme : Ngueveu et al., 2016, 2019

PGMO projects OREM (2014-2016), OPAL(2016-2018)

Step 1 : Piecewise linear bounding of the nonlinear energy transfer/efficiency functions



(c) Linear approximation



(d) Piecewise bounding

Step 2 : Reformulation of the problem into two mixed integer problems (MILP)

- solve with a MILP solver
- or design a dedicated solution method (only one needed)

Resulting PWL approximation/bounding Problem

Given a function $f : \mathbb{R} \to \mathbb{R}$ and a tolerance bound, find a PWL function that :

- Respects the imposed bounded error
- Minimizes the number of pieces of the PWL function

Benefits

- predetermined error criteria $\rightarrow$ non-iterative approach

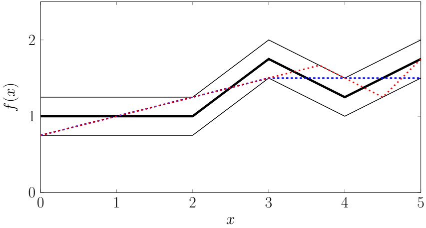
  Bound each nonlinear function with 2 PWL functions
  ↓
  Modeling the PWL functions in a MILP
  ↓
  Solving MILPs containing PWL functions
  ↓
  DONE

- Smallest MILP possible !

Why is the Problem Hard ?

Bounded tolerance constraints → semi-infinite programming (SIP)

Maximizing the length of the projection on x-axis of each segment is not optimal



Approximation of continuous fonctions with PWL functions

Minimize number of linear pieces for a bounded error or distance metric

Rosen and Pardalos, 1986. Global minimization of large-scale constrained concave quadratic problems by separable programming. **Math. Prog**

Frenzen, Sasao and Butler, 2010. On the number of segments needed in a piecewise linear approximation. **J. of Comp. and Applied Math.**

Geibler, Martin, Morsi, Schewe, 2012. Using piecewise linear functions for solving MINLPs. **The IMA Volumes in Mathematics and its applications**

Rebennack and Kallrath, 2015. Continuous piecewise linear δ-approximations for univariate functions : computing minimal breakpoint system. **J. Optim. Theory Appl**

Ngueveu, 2019. Piecewise linear bounding of univariate nonlinear functions and resulting mixed integer linear programming-based solution methods **EJOR**

Rebennack and Krasko, 2019. Piecewise linear function fitting via mixed-integer linear programming. **INFORMS Journal on Computing**

Continuous PWL approximation of univariate functions

Optimal Breakpoint System using a Continuum approach for x [RK2015]

Decision variables

- $x_b \in [X_-, X_+]$: breakpoint value
- $s_b \in [-\delta, +\delta]$: deviation on bpt b
- $\chi_b \in [0,1] = 1$ iff bpt b is used
- $y_b \geq \frac{1}{M} := x_b - x_{b-1}$ if $x_b - x_{b-1} > 0$ and $= |X_-, X_+|$ otherwise
- $\xi^x_{bx} \in [0,1]$: =1 iff $x \in [x_{b-1}, x_b]$
- $l_b(x) \in \mathbb{R} := l(x)$ if $x \in [x_{b-1}, x_b]$

Semi-infinite programming model

Obj = minimize number of active breakpoints
s.t. (c1) Order active breakpoints
(c2) Link x_b and χ_b
(c3) Compute y_b and $l_b(x)$
(c4) Compute ξ^x_{bx}
(c5) Compute $l(x)$
(c6) Ensure the δ-approximation : $|l(x) - f(x)| \leq \delta, \forall x \in D$

Continuous PWL approximation of univariate functions

Solution 1 : Semi-infinite programming models (large and difficult)

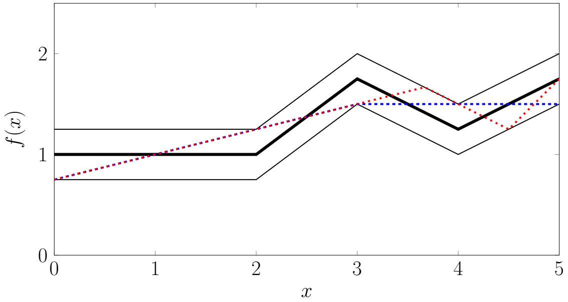
Iterative solution method

- enforce the δ gap constraint on discrete points $|\mathbb{I}|$
  - Rebennack and Kallrath 2015 $\Rightarrow$ solve a MINLP
  - Rebennack and Krasko 2019 $\Rightarrow$ solve a MILP (extension of Toriello and Vielma 2012)
- compute the real maximum error
  - Rebennack and Kallrath 2015, 2019 $\Rightarrow$ solve an NLP

Continuous PWL approximation of univariate functions

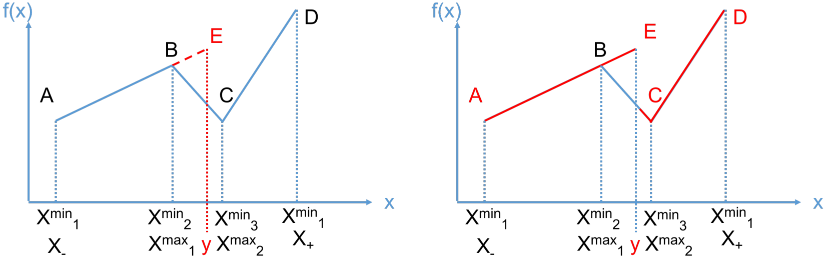
Solution 2 : Greedy heuristics computing x_b given x_{b-1},s_{b-1} and δ

- Max length of interval $[x_{b-1}, x_b]$(projection on x-axis) : not optimal



- solve the NLP problem of max x_b or try discrete values of x_b and s_b

Non necessarily continuous (nnc) PWL approximation



Proposition

Any optimal continuous δ-PWLA with n^* pieces can be converted into a non-necessarily continuous δ-PWLA with $n \leq n^*$ pieces where the projection of the first piece on interval D is of maximal length

Non necessarily continuous (nnc) δ-PWLA

Getting rid of the continuity

Theorem

$\forall$ continuous function $f : \mathbb{D} = [X_-, X_+] \to \mathbb{R}$ and any scalar $\delta \in \mathbb{R}^+$, there exists an optimal nnc δ-PWLA 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 $[x_i^{\min}, X_+]$.

The greedy algorithm becomes optimal

Non necessarily continuous (nnc) δ-PWLA

Getting rid of the continuity

Theorem

$\forall$ continuous function $f : \mathbb{D} = [X_-, X_+] \rightarrow \mathbb{R}$ and any scalar $\delta \in \mathbb{R}^+$, there exists an optimal nnc δ-PWLA 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 $[x_i^{\min}, X_+]$.

The greedy algorithm becomes optimal

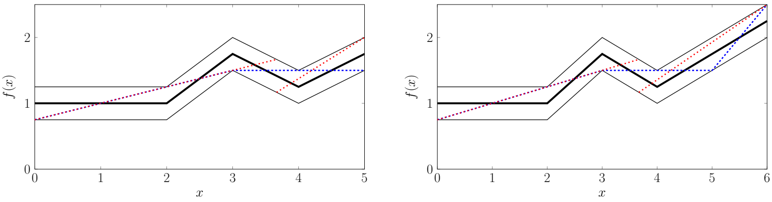
Non necessarily continuous (nnc) δ-PWLA

Getting rid of the continuity

Theorem

$\forall$ continuous function $f : \mathbb{D} = [X_-, X_+] \to \mathbb{R}$ and any scalar $\delta \in \mathbb{R}^+$, there exists an optimal nnc δ-PWLA 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 $[x_i^{\min}, X_+]$.

The greedy algorithm becomes optimal



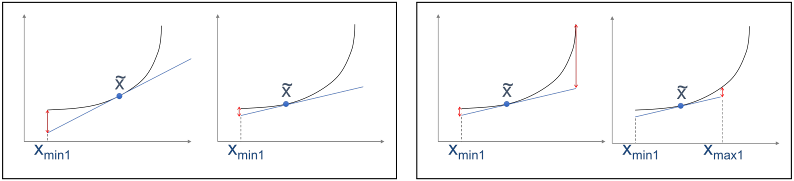
$(n^* + 1)/2 \leq n_g \leq n^*$

Better δ-PWL approximation algorithms

From δ-approx to $\frac{\delta}{2}$-under(-over)estimator : shift by $\frac{\delta}{2}$

Convex or concave function

optimal 2-steps dichotomic search : No NLP or MINLP solved !



step a : max $\tilde{x}$ step b : max $x_{\max_i}$

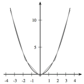
Piecewise convex or concave function with p pieces

approx/bound each convex or concave piece separately

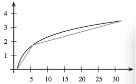
$$n_g^* \leq n_g \leq n_g^* + p$$

Numerical comparisons Computational evaluation on continuous functions Computational evaluation on continuous functions

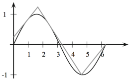
Computational evaluation on continuous functions



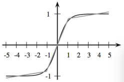
x^2



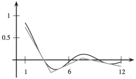
$\ln(x)$



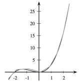
$\sin(x)$



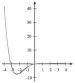
$\tanh(x)$



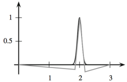
$\frac{\sin(x)}{x}$



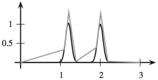
$2x^2 + x^3$



$e^{-x}\sin(x)$



$e^{-100(x-2)^2}$



$1.03e^{-100(x-1.2)^2} + e^{-100(x-2)^2}$

f		δ	continuous approx. [RK2015]		nnc approximation Exact		Heuristic			
	p		n_*	cpu	n_*	cpu	n_*	n_-	n_+	cpu
I	1	0.01	**25**	Hours	**25**	19 s	**25**			30 s
		0.005	**35**	Hours	**35**	4 s	**35**			43 s
II	1	0.01	**9**	Sec	**9**	221 s	**9**			11 s
		0.005	**13**	Sec	**13**	172 s	**13**			16 s
III	2	0.01	**13**	Sec	**13**	57 s		13	14	
		0.005	**17**	Few min	**17**	68 s		17	18	
IV	2	0.01	**9**	Few sec	**9**	161 s		9	10	10 s
		0.005	**13**	Few min	**13**	128 s		13	14	15 s
V	4	0.01	9	Sec	**8**	143 s		7	10	11 s
		0.005	**12**	Few min	**12**	181 s		12	15	15 s

f		δ	continuous approximation [RK2015]				nnc approximation [Ng2019] Exact		Heuristic		
	p		n_*	n_-	n_+	cpu	n_*	cpu	n_-	n_+	cpu
VI	2	0.05	**15**			Few days	**15**	88 s	15	16	17 s
		0.01		15	**34**		**34**	164 s	34	35	36 s
		0.005		15	**47**		**47**	195 s	47	48	59 s
VII	3	0.05		4	**19**		**19**	287 s	19	21	28 s
		0.01		4	**43**		**43**	268 s	43	45	52 s
		0.005		4	**61**		**61**	869 s	61	63	72 s
VIII	3	0.05		4	6		**5**	83 s	4	6	6 s
		0.01		4	**11**		**11**	138 s	10	12	11 s
		0.005		4	**14**		**14**	1466 s	14	16	16 s
IX	5	0.05		7	11		**9**	64 s	7	11	12 s
		0.01		7	**21**		**21**	114 s	19	23	23 s
		0.005		7	28		**27**	784 s	27	31	27 s

f		δ	continuous approximation [RK2019]			nnc approximation [Ng2019] Exact		Heuristic		
	p		n_*	n_-	cpu	n_*	cpu	n_-	n_+	cpu
VI	2	0.05	**15**		107.7 s	**15**	88 s	15	16	17 s
		0.01		31	6819.1 s	**34**	164 s	34	35	36 s
		0.005		40	35787.4 s	**47**	195 s	47	48	59 s
VII	3	0.05	**19**		14514.6 s	**19**	287 s	19	21	28 s
		0.01		34	35411.0 s	**43**	268 s	43	45	52 s
		0.005		35	70313.1 s	**61**	869 s	61	63	72 s
VIII	3	0.05	**5**		5.3 s	**5**	83 s	4	6	6 s
		0.01	**11**		59.6 s	**11**	138 s	10	12	11 s
		0.005	**14**		247.2 s	**14**	1466 s	14	16	16 s
IX	5	0.05	**9**		12 s	**9**	64 s	7	11	12 s
		0.01	**21**		13873.4 s	**21**	114 s	19	23	23 s
		0.005	**27**		42068.4 s	**27**	784 s	27	31	27 s

Weaknesses/Limitations of PWL-based solution methods

- dependent on the instance : e.g. dependent on time horizon
- dependent on the solution : e.g. solution cost
- difficult to choose a relevant value of δ for a new instance or worse a new problem

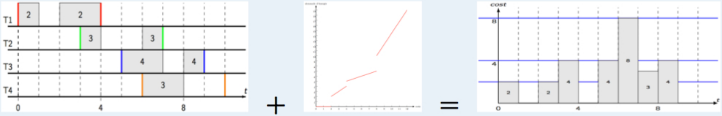
Energy optimization in hybrid electric vehicle

$$\min \sum_{i=1}^{n} f_{\mathrm{FC}}(x_i) \tag{11}$$

s.t.
(c1) Power demand satisfaction
(c2) Final SE energy level higher or equal to initial energy level
(c3) SE energy level within bounds
(c5) Variables domains

Weaknesses/Limitations of PWL-based solution methods

Step 2 : Pre-emptive scheduling problem with non linear efficiency functions



(c) Scheduling + (d) Efficiency = (e) Energy cost

Minimize the total energy cost

$$(\mathrm{CF})\ \min \quad F = \sum_{t \in T} f\Big(\sum_{i \in \mathcal{A}} b_i x_{it}\Big) \tag{12}$$

s.t. Satisfaction of the demand for each activity

$$\sum_{t \in T} a_{it} x_{it} \geq p_i, \qquad \forall i \in \mathcal{A} \tag{13}$$

Validity domain

$$x_{it} \in \{0, 1\}, \qquad \forall i \in \mathcal{A}, t \in T \tag{14}$$

Use relative ϵ-tolerance

Principle

For a function $f : \mathbb{D} \to \mathbb{R}^*$ and a tolerance value $\epsilon \in [0, 1]$, identify two piecewise linear functions $(\overline{f}^{\epsilon}, \underline{f}^{\epsilon})$ that verify :

$$\underline{f}^{\epsilon}(x) \leq f(x) \leq \overline{f}^{\epsilon}(x), \quad \forall x \in \mathbb{D} \tag{15}$$

$$|f(x) - \underline{f}^{\epsilon}(x)| \leq \epsilon |f(x)|, \quad \forall x \in \mathbb{D} \tag{16}$$

$$|\overline{f}^{\epsilon}(x) - f(x)| \leq \epsilon |f(x)|, \quad \forall x \in \mathbb{D} \tag{17}$$

Purpose

- Apply on nonlinear term BEFORE insertion in the model
  - univariate function that can be convex or concave, easier to bound
  - minimize number of pieces ($\neq$ spatial or interval branch-and-bound)
- Two MILP ($\overline{\mathrm{MILP}}$ and $\underline{\mathrm{MILP}}$) are obtained
- Guarantees on the quality of the resulting lower and upper bounds
- Upper and lower bounding PWL functions $\overline{f}^{\epsilon}(x)$ and $\underline{f}^{\epsilon}(x)$ may not share the same breakpoints and number of pieces (no shift !)

Numerical comparisons

Comparison on the energy optimization problem for hybrid electric vehicles

- upper bounds
- lower bounds
- absolute vs relative tolerance

Comparison of upper bounds

Instances			ϵ-PWLB+MILP [Ng2019]			Effpts+MILP [GCL2013]	
class	#	$\overline{n}^{\epsilon}$	gap orig	cpu	gap recomp	gap	cpu
(R)	6	6	0.20 %	16 s	0.04 %	1.07 %	3000 s
		56	**0.01** %	58 s	**0.01** %	0.03 %	314 s
(A1)	6	10	0.26 %	31 s	0.18 %	0.62 %	3000 s
		82	**0.01** %	78 s	**0.01** %	0.03 %	488 s
(A2)	6	14	0.52 %	96 s	0.49 %	2.26 %	2436 s
		133	**0.01** %	627 s	**0.01** %	0.07 %	1213 s

Instances			antigone	baron	couenne	lindoglobal
class	#	optCR	gap	gap	gap	gap
(R)	6	1 %	13.86 %	9.64 %	20.97 % (4)	0.01 % (5)
		0.01 %	3.78 %	9.69 %	5.40 % (4)	0.01 % (5)
(A1)	6	1 %	1.06 %	8.62 %	3.44 % (5)	0.37 % (5)
		0.01 %	1.71 %	10.15 %	3.44 % (5)	0.37 % (5)
(A2)	6	1 %	0.96 %	14.74 %	20.11 % (2)	0.00 % (5)
		0.01 %	0.94 %	14.73 %	20.11 % (2)	0.00 % (5)

Comparison of lower bounds

Instances		old lower bound [NCMG2019]		ϵ-PWLB+MILP [Ng2019]		
set	#	ratio	cpu	ϵ	ratio	cpu
(R)	6	97.81 %	2 s	1 %	99.18 %	20 s
				0.01 %	**99.99** %	58 s
(A1)	6	97.36 %	2 s	1 %	99.25 %	29 s
				0.01 %	**99.99** %	216 s
(A2)	6	78.50 %	2 s	1 %	99.50 %	26 s
				0.01 %	**99.99** %	405 s

Instances			antigone	baron	couenne	lindoglobal
class	#	optCR	ratio	ratio	ratio	ratio
(R)	6	1 %	24.66 %	- % (6)	- % (6)	- % (6)
		0.01 %	21.32 %	- % (6)	- % (6)	- % (6)
(A1)	6	1 %	49.41 %	37.89 % (0)	33.03 % (3)	35.99 % (5)
		0.01 %	49.60 %	37.72 % (0)	33.07 % (3)	35.80 % (5)
(A2)	6	1 %	53.21 %	- % (6)	- % (6)	- % (6)
		0.01 %	53.21 %	- % (6)	- % (6)	- % (6)

Comparison of relative vs absolute tolerance

Class (R)	using relative tol			using absolute tol		Gap
instance	ϵ	n^ϵ	UB	δ	n^δ	$\frac{n^\delta - n^\epsilon}{n^\epsilon}$
S_40	1 %	6	454.3	0.1136	14	133.33 %
	0.01 %	56	453.4	0.0011	133	137.50 %
I_561	1 %	6	8756.6	0.1561	12	100.00 %
	0.01 %	56	8741.0	0.0016	110	96.43 %
H_734	1 %	6	18626.0	0.2538	10	66.67 %
	0.01 %	56	18569.0	0.0025	89	58.93 %
U_811	1 %	6	2613.5	0.0322	25	316.67 %
	0.01 %	56	2607.7	0.0003	255	355.36 %
N_1200	1 %	6	23137.7	0.1928	11	83.33 %
	0.01 %	56	23114.9	0.0019	102	82.14 %
E_1400	1 %	6	27088.9	0.1935	11	83.33 %
	0.01 %	56	27065.9	0.0019	102	82.14 %

58% to 355% more linear pieces with absolute vs relative tolerance

PWL bounding with relative ϵ-tolerance

Strength

- efficiency for problems with a nonlinear function per data set
- various fields and domains of application

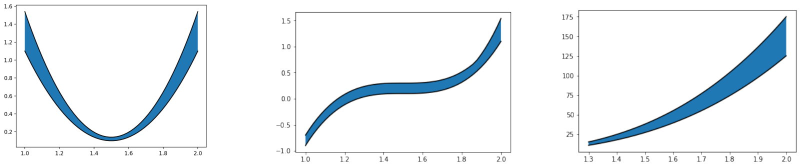
To be improved

- access/ease-of-use for non-technical users
- speed improvement to tackle problems with multiple nonlinear functions

New perspective

Definition (Corridor)

Let $h, l : [a,b] \to \mathbb{R}$ C^1 $h(x) \geq l(x), \forall x \in [a,b]$ and having the same concavity. We call $\mathcal{C} = \{(x,y) | x \in [a,b], l(x) \leq y \leq h(x)\}$ a corridor between h and l .



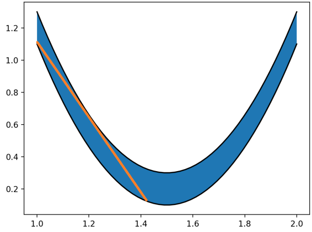
Codsi, Gendron, Ngueveu (2019)

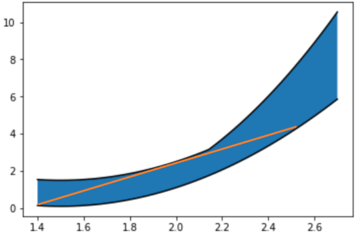
Convex corridor

Theorem (Convex corridor segment characterization)

On convex corridor $\mathcal{C}$ there exists an optimal linear segment such that

- *Both ends lie on the lower curve*
- *it is tangent to the upper curve*



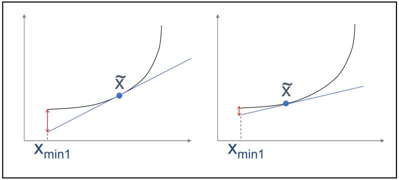


Convex corridor algorithm

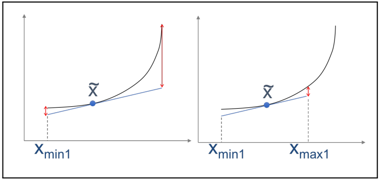
generalisation of the algorithm of Ngueveu 2019

- Use dichotomy to find the tangent point
- find the resulting segment
- repeat 1 and 2 for the next segment

optimal 2-steps dichotomic search : No NLP or MINLP solved !



step a : max $\tilde{x}$



step b : max x_{max_i}

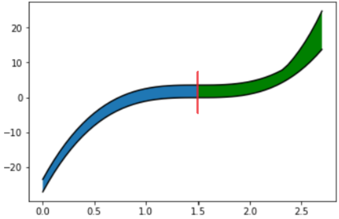
Strength of the algorithm

- logarithmic convergence (for each segment)
- Works on concave corridors too
- not limited to absolute error
- can be used to approximate, underestimate and overestimate functions

Corridors Without Constant Convexity

Splitting the corridor

+ parallelizable
+ Efficient
- Heuristic Not necessarily optimal but the error is tightly bounded



$$n^* \leq n \leq n^* + \#\text{Sub-corridors}$$

O'Rourke adaptation

based on function sampling and constraints on the line coefficient space

+ Exact
- Not as efficient

function	absolute error	Continuous		ncc			
		time					
		exact		exact		Heuristic	
		[RK15]	[RK19]	[Ngu19]	New	[Ngu19]	New
VI	0.1	Min	24.4 s	115 s	2.9 s	11 s	0.8 s
	0.05	Few days	107.7 s	88 s	3.0 s	17 s	0.8 s
	0.01	*	*	164 s	3.5 s	36 s	0.8 s
	0.005	*	*	195 s	2.8 s	59 s	0.8 s
VII	0.1	*	311.1 s	226 s	4.0 s	17 s	0.7 s
	0.05	*	14 514.6 s	287 s	3.9 s	28 s	0.8 s
	0.01	*	*	268 s	4.9 s	52 s	0.9 s
	0.005	*	*	869 s	5.0 s	72 s	0.9 s
VIII	0.1	sec	1.7 s	74 s	4.6 s	4 s	0.4 s
	0.05	*	5.3 s	83 s	5.4 s	6 s	0.4 s
	0.01	*	59.6 s	138 s	4.7 s	11 s	0.4 s
	0.005	*	247.2 s	1466 s	4.5 s	16 s	0.4 s
IX	0.1	Few days	1.9 s	77 s	8.3 s	8 s	0.8 s
	0.05	*	12 s	64 s	8.1 s	12 s	0.8 s
	0.01	*	13 873.4 s	114 s	8.1 s	23 s	0.8 s
	0.005	*	42 068.4 s	784 s	8.1 s	27 s	0.7 s

Results

Network design with congestion

- from a multivariate objective-function ... $\min \sum_{(i,j)\in A} O_{ij} y_{ij} + \sum_{(i,j)\in A} \sum_{p\in P} D_{ij}^p x_{ij}^p + \sum_{i\in N} E_i z_i + \sum_{i\in N} g_i(\sum_{j\in N} \sum_{p\in P} x_{ij}^p, z_i)$
- ... to a univariate objective-function $\min \sum_{(i,j)\in A} O_{ij} y_{ij} + \sum_{(i,j)\in A} \sum_{p\in P} D_{ij}^p x_{ij}^p + \sum_{i\in N} f_i(v_i)$

instance	litterature	new
c35_0.3_0.6	6,615 s	6,92 s
c36_0.8_0.8	21,528 s	14,59 s
c49_0.8_0.6	172,255 s	118,17 s
c50_0.8_0.6	2609,568 s	2575,08 s

- as good as advanced state of the art solution methods
- no consideration on the problem structure
- easy to implement

Implementation

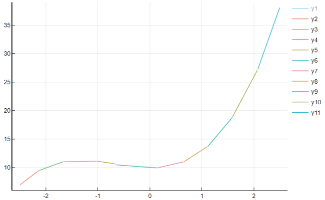
An approximation package named JULIA

Implementing both the exact and heuristic methods and include many classical error metrics !

- Julia based

$2x^2 + x^3 + 10$

```
linearApprox(:(2x^2+x^3+10),-2.5, 2.5, relative = true, epsilon = 1.0)
```



With an error guarantee !

Conclusion

Done

- PWLA+MILP efficient for solving certain classes of MINLPs
  - Non-necessarily continuous piecewise linear functions
  - Relative $\epsilon-$ tolerance
  - Bounding instead of approximation
- 2 similar MILPs to solve
- Various applications

What next ?

- Coming soon (next few days) : opensource PWLB toolbox
- Extension to non-separable functions
- Other classes of problems ? (stochastic programming ?)

Bound each nonlinear function with 2 PWL functions

↓

Modeling the PWL functions in a MILP

↓

Solving MILPs containing PWL functions

↓

DONE

Bibliography I

Bellman, R. and Roth, R. (1969)
Curve fitting by segmented straight liness.
Journal of the American Statistical Association, 64 :1079 - 1084.

Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, L, Margot, F., Sawaya, N. and Wächter, A., (2008)
An algorithmic framework for convex mixed integer nonlinear programs.
Discrete Optimization, 5(2) : 186 - 204. In Memory of George B. Dantzig.

Borghetti, A. C. D'Ambrosio, A. Lodi, and S. Martello (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.

Burlacu, R., Geibler, B., and Schewe, L. (2019)
Solving mixed-integer nonlinear programmes using adaptively refined mixed-integer linear programmes.
Optimization Methods and Software, DOI : 10.1080/10556788.2018.1556661.

Camponogara, E., M.P. de Castro, and A. Plucenio (2007).
Compressor scheduling in oil fields : A piecewise-linear formulation.
In *IEEE int. conf. on Automation Sci. and Engineering. CASE 2007*, p. 436 - 441.

Bibliography II

Codsi, J., Gendron, B. and Ngueveu S.U. (2019).
Non necessarily continuous piecewise linear approximation with a performance guarantee : application to mixed integer optimization.
*LAAS report n*19145 - HAL.*

D'Ambrosio, C., Lodi, A. and Martello, S. (2010).
Piecewise linear approximation of functions of two variables in MILP models.
Operations Research Letters, 38 :39 - 46.

D'Ambrosio, C., Lodi, A. Wiese, S. and Bragalli, C. (2015).
Mathematical programming techniques in water network optimization.
European Journal of Operational Research, 243(3) :774 - 788.

Floudas C.A. and C.E. Gounaris (2009).
A review of recent advances in global optimization.
Journal of Global Optimization, 45 : 3 - 389.

Frenzen, C.L., Sasao, T. and Butler J.T. (2010).
On the number of segments needed in a piecewise linear approximation.
J. of Comp. and Applied Math., 45 : 3 - 389.

Gaoua, Y., Caux, S. and Lopez P. (2013).
Energy management for an electric vehicle based on combinatorial modeling.
In *2013 IEEE Int. Conf. on Ind. Eng. and Systems Management (IESM)*, p. 1 - 6.

Bibliography III

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.

Grossmann I.E. (2002).
Review of nonlinear mixed-integer and disjunctive programming techniques.
Optimization and Engineering, 3 :227 - 252.

Huchette, J. and Vielma, J.P. (2018).
Nonconvex piecewise linear functions : Advanced formulations and simple modeling tools
arXiv :1708.00050v2.

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 Quaterly Journal of Operations Research, 15(4) :407 - 430.

Bibliography IV

Ngueveu, S. U. (2019).
Piecewise linear bounding of non linear univariate functions and mixed integer linear programming-based solution methods.
European Journal of Operational Research, 1058-1071.

O'Rourke, J. (1981)
An on-line algorithm for fitting straight lines between data ranges
in *Communications of the ACM*, 24(9) :574 - 578.

Paraskevopoulos, D. C. and Gürel, S. and Bektaş, T., (2016)
The congested multicommodity network design problem
Transportation Res. Part E : Logistics and Transportation Review, 85 :166 - 187.

Polisetty P.K. and E.P. Gatzke (2006).
A decomposition-based MINLP solution method using piecewise linear relaxations.
Technical report, University of South Carolina.

Rebennack, S. and Kallrath, J. (2014).
Continuous Piecewise Linear Delta-Approximations for Bivariate and Multivariate Functions.
J. Optim. Theory Appl., 167(1) :102 - 117.

Bibliography V

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.

Rebennack, S. and Krasko, V. (2019).
Piecewise linear function fitting via mixed-integer linear programming.
INFORMS Journal on Computing, to appear.

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. (1974).
Piecewise-linear approximation with a bound on absolute error.
Computers and Biomedical Research, 7(1) :64 - 70.

Toriello, A. and Vielma J.P. (2012).
Fitting piecewise linear continuous functions.
European Journal of Operational Research, 219 :86 - 95.

Useful Tools / Julia Packages

LINA : Computing a PWL approximation, over-/under-estimators with minimum # linear segments

- link : `http://homepages.laas.fr/sungueve/LINA.html`
- input : a univariate continuous nonlinear function
- output : a nnc PWL function with minimum number of pieces
- related reference : Codsi, Gendreau, Ngueveu (2019-HAL)

PiecewiseLinearOpt : Modeling efficiently a given continuous PWL function in MILP

- `https://github.com/joehuchette/PiecewiseLinearOpt.jl`
- input : a continuous PWL function (or sampled nonlinear fct)
- output : variables and constraints to insert in a MILP
- related reference : Huchette and Vielma (2018-arXiv)