Introduction to Integer Linear Programming, reformulation and decomposition

Sandra Ulrich Ngueveu

CIMPA School 2021: July 6th - 8th, 2021

Université de Toulouse - Toulouse INP - Laboratory LAAS-CNRS
ngueveu@laas.fr

Part 2: using piecewise linear functions to approximate MINLP with MILP

Content

Illustrative case studies

How to obtain a *good* piecewise linear function ? (Focus on 1D)

How to model the piecewise linear function within an MILP

Illustrative case studies

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):

$$V_m = rI_m + k_\Phi \Omega \quad (1)$$

$$T_m = \Phi_m I_m \quad (2)$$

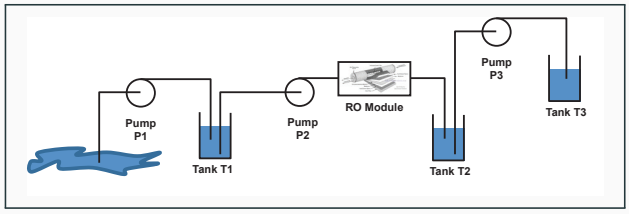
Electrical power needed: $P_e = V_m I_m$.

Pressure drop in the pipe

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

Static+Dynamic pressure

$$\Delta\mathrm{Pipe} = kq^2 + \rho g h \quad (3)$$



Mechanical-Hydraulic conv.

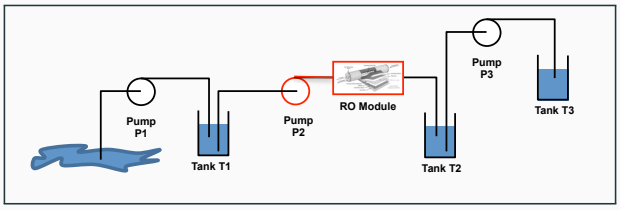
- P_p: output pressure
- q: debit of water
- a, b: non linear girator coefs
- c: hydraulic friction
- p_0: suction pressure
- $s_p + s_m$: mechanical losses

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

$$P_p = (a\Omega + bq)\Omega - (cq^2 + p_0) \quad (4)$$

$$T_m = (a\Omega + bq)q + (s_m + s_p)\Omega \quad (5)$$

Efficiency function of pump 2 + reverse osmosis module



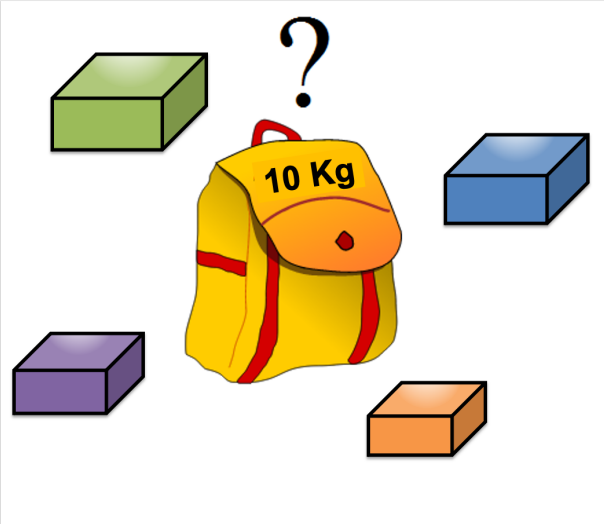
Subsystem pump 2 + Reverse Osmosis module is modeled with

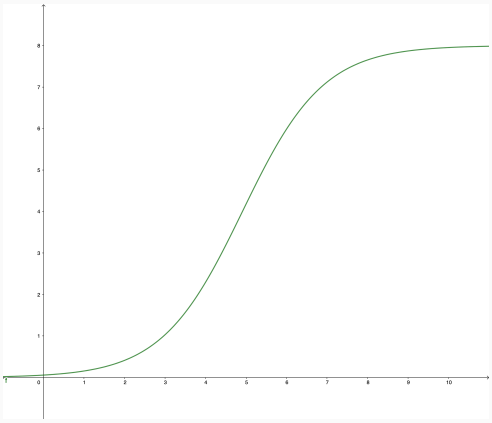
$$\mathsf{P_e} = r*\mathcal{K}(\mathsf{q},\mathsf{h})+((s_m+s_p)*\Omega(\mathsf{q},\mathsf{h})+(\mathsf{q}+\mathcal{F}(\mathsf{q})/R_{\mathrm{Me}})*\mathcal{M}(\mathsf{q},\mathsf{h}))*\Omega(\mathsf{q},\mathsf{h})$$

where

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

Our case study: the nonlinear knapsack problem





example of nonlinear cost function f_1

$$\max f_1(x_1) + f_2(x_2) + f_3(x_3) + f_4(x_4) \tag{6}$$

s.t.

$$h_1(x_1) + h_2(x_2) + h_3(x_3) + h_4(x_4) \leq 10 \tag{7}$$

$$x_1, x_2, x_3, x_4 \geq 0 \tag{8}$$

Classical MINLP solution methods

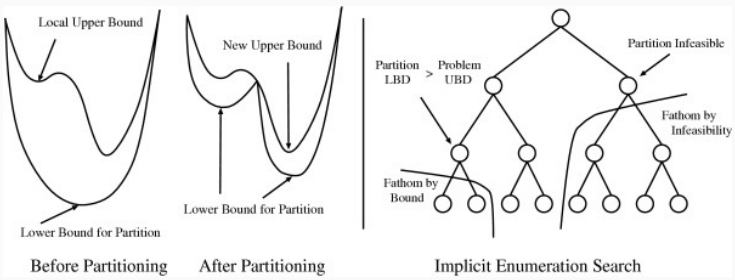
(a) Branch-and-Bound-based methods for MINLP

spatial B&B,α-B&B, interval analysis, Branch-and-reduce

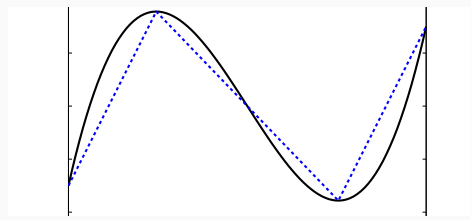
+ global optimality guaranteed if carried out to completion
- only for small/medium instances

(b) Approximate with piecewiselinear function and solve MILP

+ (more) tractable problems
+- can we prove guarantee global optimality ?



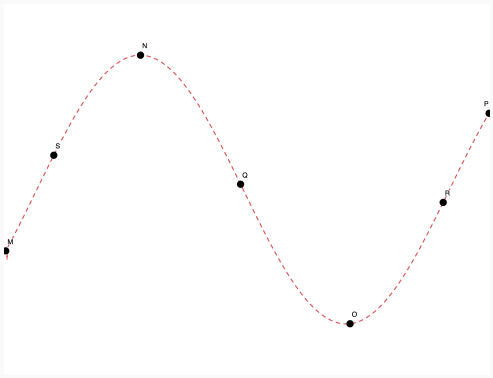
(a)



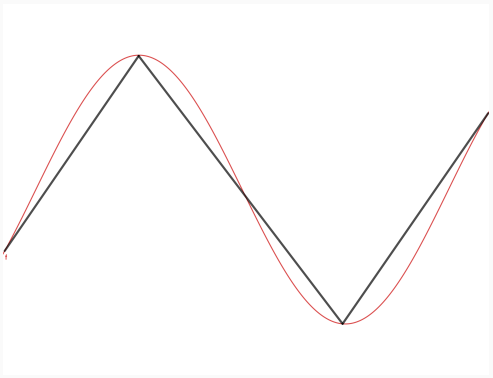
(b)

How to obtain a *good* piecewise linear function ? (Focus on 1D)

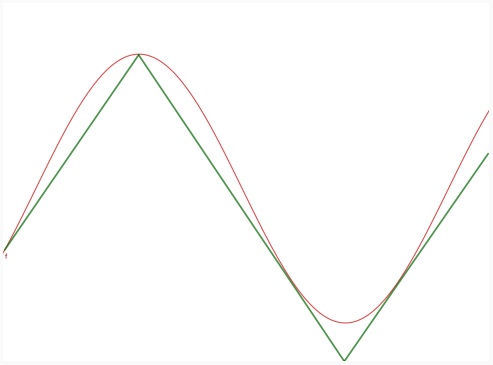
4 ways to (piecewise) linearize a nonlinear function



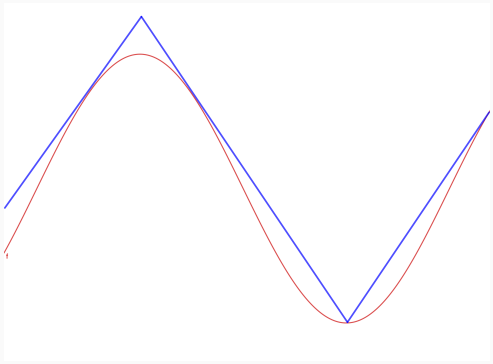
(a) Sampling



(b) Approximation



(c) Lower bounding



(d) Upper bounding

How can we assess the quality of a (piecewise) linearization ?

Let $f(x) : \mathbb{D} \to \mathbb{N}$ be a nonlinear function and let $g(x)$ be a piecewise linear function defined on the same domain. Then $e_a(x) = |f(x) - g(x)|$ expresses the pointwise absolute approximation error at point x while $e_r(x) = \frac{|f(x)-g(x)|}{|f(x)|}$ is the pointwise relative error.

The two classical criteria to evaluate PWL functions, are:

- the maximal pointwise error(relative or absolute):

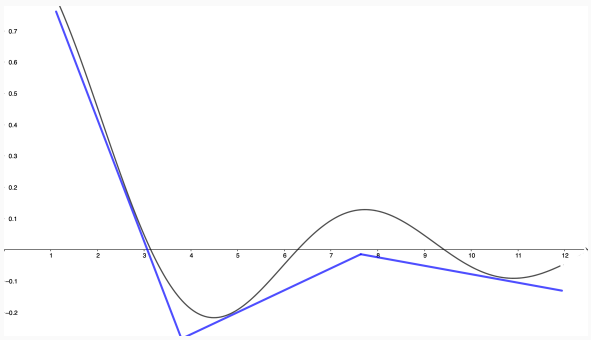
$$\max_{x \in \mathbb{D}} e_a(x) \quad \text{or} \quad \max_{x \in \mathbb{D}} e_r(x)$$

- or the integral of the pointwise error:

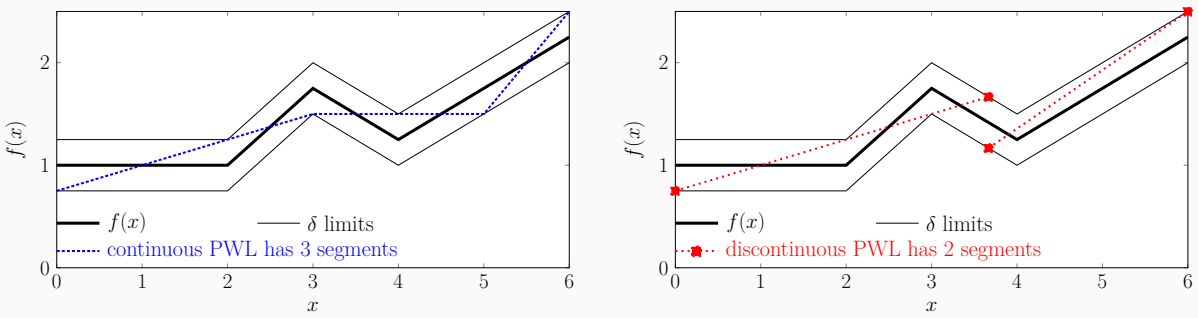
$$\int_{\mathbb{D}} e_a(x)dx \quad \text{or} \quad \int_{\mathbb{D}} e_r(x)dx$$

Consequential choices

Equidistant breakpoints or not ? Interpolation or not ?

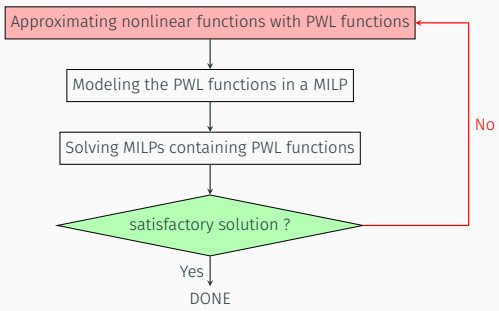


Continuous PWL or discontinuous PWL ?

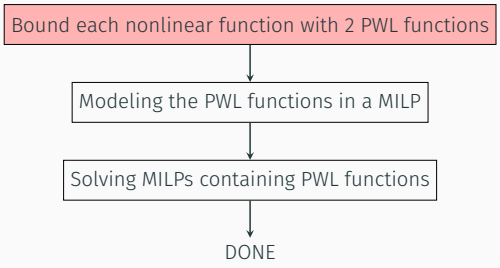


Challenges

A naïve construction of the PWL function can lead to a solution method (“PWL building +MILP resolution”) that becomes a try and error approach: no guarantees on the solution quality or iterative process with a number of iterations not defined a priori.



$\Rightarrow$?

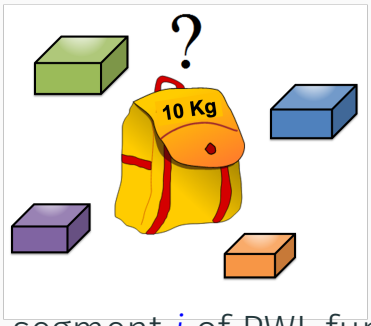


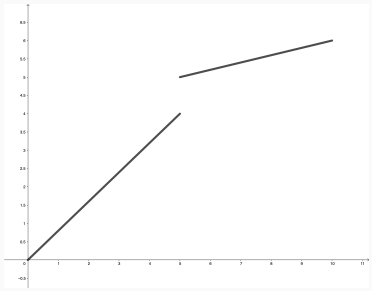
Other difficulties:

- semi-infinite programming
- smaller precision target $\rightarrow$ more segments

How to model the piecewise linear function within an MILP

A simpler nonlinear knapsack problem to work with





Each segment i of PWL function g_A is defined by the left extremity (x_i^L, y_i^L), the right extremity (x_i^R, y_i^R), the slope a_i, the y-intercept b_i.

E.g.: $(x_1^L = 0, y_1^L = 0)$, $(x_1^R = 5, y_1^R = 4)$,$a_1 = 0.8$ and $b_1 = 0$

$$\max g_A(x_A) + 40x_B + 12x_C + 25x_D \tag{9}$$

s.t.

$$x_A + x_B + x_C + x_D \leq 10 \tag{10}$$

$$x_A, x_B, x_C, x_D \geq 0 \tag{11}$$

How to formulate the PWL term $g_A(x_A)$?

PWL modelling

New decision variables

- β_i: equal to 1 if segment i is active, 0 otherwise
- α_i: equal to the x_A if $x_A \in$ segment i

Replace $g(x_A)$ and x_A using equations (12)-(13).

$$g(x_A) = \sum_{i \in \mathcal{I}} (a_i \alpha_i + b_i \beta_i) \tag{12}$$

$$x_A = \sum_{i \in \mathcal{I}} (\alpha_i) \tag{13}$$

Complementary constraints:

$$x_i^L \beta_i \leq \alpha_i \leq x_i^R \beta_i \tag{14}$$

$$\sum_{i \in \mathcal{I}} \beta_i = 1 \tag{15}$$

PWL modelling $\Rightarrow |I|$ binary and $|I|$ continuous variables.

Possible PWL models

Several models exists, especially for the continuous case. The choice of the model must be carefully considered.

- CC (Convex Combination)
- Inc (Incremental)
- DCC (Disaggregated Convex Combination)
- Log (Logarithmic Convex Combination)
- DLog (Disaggregated Logarithmic Convex Combination)
- Multiple Choice
- SOS2 (Specially Ordered Sets of Type 2)

Example of a logarithmic reformulation

Key idea: use $log_2(I)$ binary variables δ_i to encode I intervals

i	$\delta_3\delta_2\delta_1$
1	000
2	001
3	010
4	011
5	100
6	101
7	110
8	111

$U_1 = \{2,4,6,8\}$
$U_2 = \{3,4,7,8\}$
$U_3 = \{5,6,7,8\}$
$V_1 = \{1,3,5,7\}$
$V_2 = \{1,2,5,6\}$
$V_3 = \{1,2,3,4\}$

Decision variables:

- δ_j: binary variable for encoding
- $\lambda_i, \theta_i \leq 1$: chosen weight for x_i^L, x_i^R

Replace $g(x_A)$ and x_A using :

$$x_A = \sum_{i \in \mathcal{I}} \left(x_i^L \lambda_i + x_i^R \theta_i \right) \text{ and } g(x_A) = \sum_{i \in \mathcal{I}} \left(y_i^L \lambda_i + y_i^R \theta_i \right)$$

$$\sum_{i \in \mathcal{I}} (\lambda_i + \theta_i) = 1$$

Complementary constraints

$$\sum_{i \in U_j} (\lambda_i + \theta_i) \leq \delta_j, \text{ and } \sum_{i \in V_j} (\lambda_i + \theta_i) \leq (1 - \delta_j), \quad \forall j \in \{1 \dots \log_2(|I|)\} \tag{16}$$

where U_j (resp. V_j) set of intervals that can not be active if $\delta_j = 0$ (resp. $1 - \delta_j = 0$)

PWL modelling $\Rightarrow$ $\log_2(|I|)$ binary and $2|I|$ continuous variables.

Useful Tools / Julia Packages

LinA: Computing a PWL approximation, over-/under-estimators with minimum number of linear segments

- link: http://homepages.laas.fr/sungueve/LinA.html
- input : a univariate continuous nonlinear function
- output : a PWL function with minimum number of pieces

PiecewiseLinearOpt: Modeling efficiently a given continuous PWL function in MILP

- link: https://github.com/joehuchette/PiecewiseLinearOpt.jl
- input : a continuous PWL (or sampled nonlinear) function
- output : variables and constraints to insert in a MILP

In summary

- PWL functions can be used to approximate non-linearities present in various optimization problems.
- When carefully constructed and combined, PWL functions can lead to MILP which can provide tight bounds and guarantees on the quality of the solutions obtained.
- Such careful constructions are based on concepts of over- and under-estimation of the original nonlinear functions together with the ability to compute the approximation errors a priori.
- The efficiency of the “PWL approximation + MILP” solution method leverages significant advances in MILP solvers for reasonable-sized instances

Next lesson introduces a decomposition method for larger instances