A scalable algorithm for solving a class of separable nonconvex MINLPs to arbitrary numerical precision

Sandra Ulrich NGUEVEU

with the collaboration of Julien Codsi, Claudio Contardo, Bernard Gendron

November 30th,2022

http://homepages.laas.fr/sungueve
ngueveu@laas.fr

Outlook

Outlook

We are interested in optimization problems that can be modeled as follows:

$$\min f(x) = \sum_{i=1}^{n} f_i(x_i) \qquad (1)$$

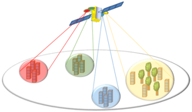
s.t.

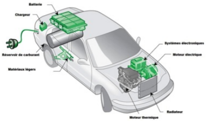
$$Ax \geq b \qquad (2)$$

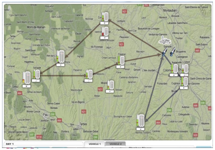
$$x \in X, \qquad X \subset \mathbb{R}^n \times \mathbb{Z}^{n-p} \qquad (3)$$

nonlinear f	MINLP
linear f	MILP

Numerous fields of application







Classical MINLP solution methods

Generic MINLP solution methods / Hybrid algorithms and frameworks

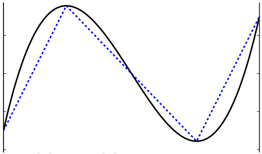
Grossmann 2002 / Bonami *et al.*, 2008

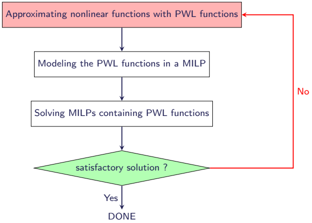
+ global optimality guaranteed if carried out to completion
- restricted to small/medium instances in the absence of specific properties

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

Classical MINLP solution methods

Generic MINLP solution methods / Hybrid algorithms and frameworks

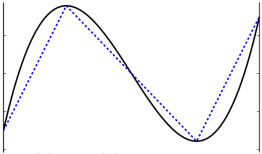
Grossmann 2002 / Bonami *et al.*, 2008

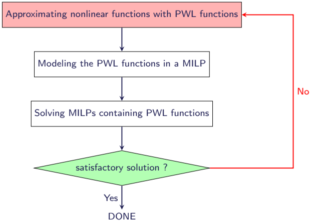
+ global optimality guaranteed if carried out to completion
- restricted to small/medium instances in the absence of specific properties

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

Classical MINLP solution methods

Generic MINLP solution methods / Hybrid algorithms and frameworks

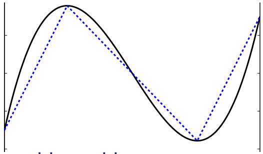
Grossmann 2002 / Bonami *et al.*, 2008

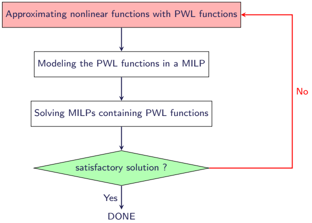
+ global optimality guaranteed if carried out to completion
- restricted to small/medium instances in the absence of specific properties

MILP-based solution methods on similar problems

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

- approximate with piecewise linear functions



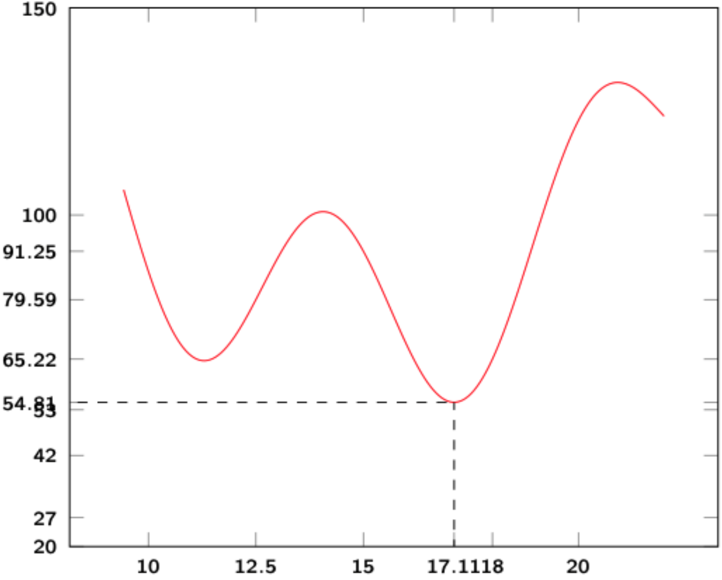


+ (more) tractable problems
- try and error approach: No guarantees on the solution quality or iterative process with an undefined number of iterations
- global optimality cannot be guaranteed

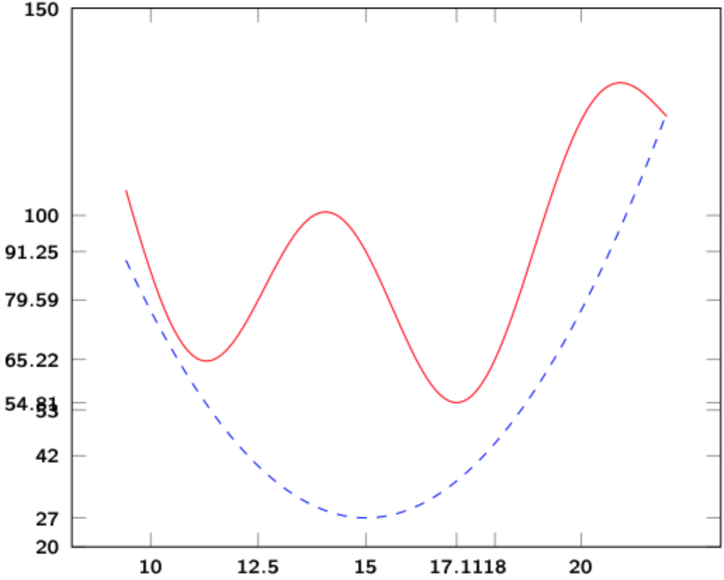
Spatial branch-and-bound

- Spatial branch-and-bound
  1. Maintain global LBs and UBs
  2. Solve a convex relaxation of a problem, compute LBs and UBs along the way
  3. If global optimality is not proven, split the space into two subregions
  4. Tighten the convex relaxation for each subregion separately and get back to 2

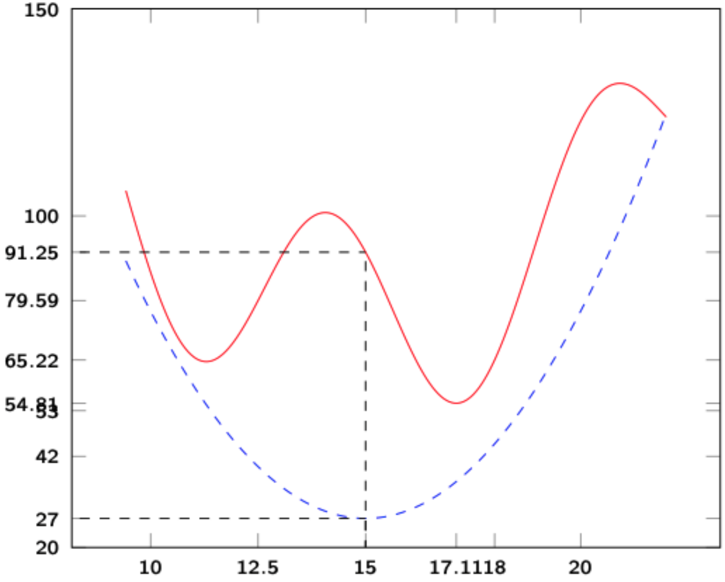
Spatial branch-and-bound



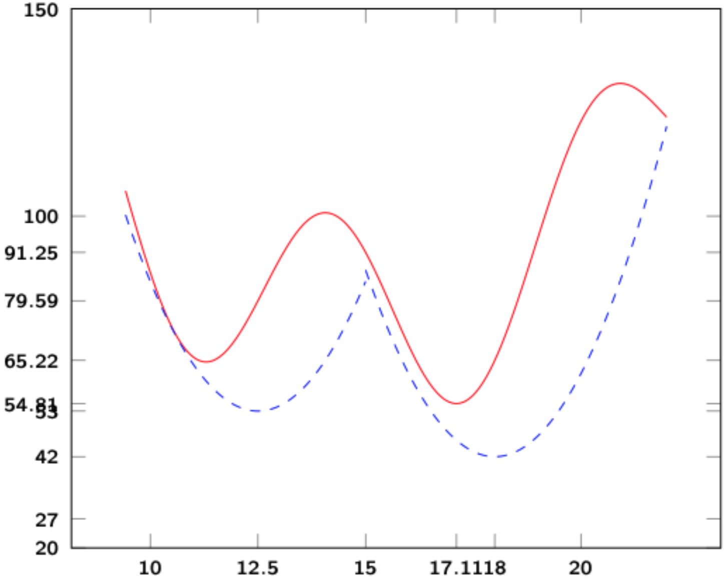
Spatial branch-and-bound



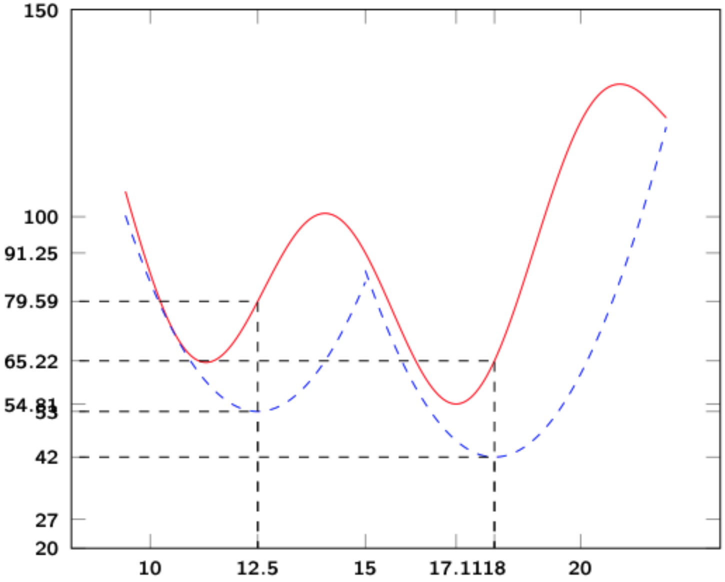
Spatial branch-and-bound



Spatial branch-and-bound

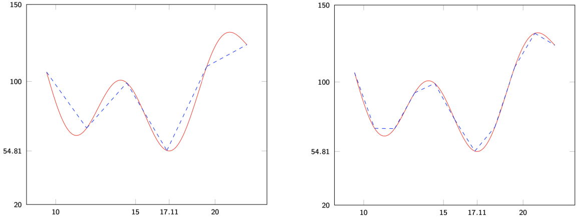


Spatial branch-and-bound



Piecewise linear approximations

- Constructs a continuous piecewise linear function that interpolates the nonlinear function at the breakpoints
- The finer the granularity, the better the approximation

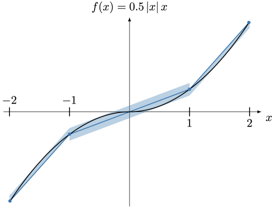


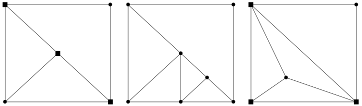
- Numerical guarantees are dependent on the size of the discretization
- Usually tractable only for very rough guarantees on very large problems

Find a global optimal solution of the MINLP by solving a series of MIP relaxations with gradually increasing accuracy.

Consider MINLPs with a linear objective and nonlinear constraints

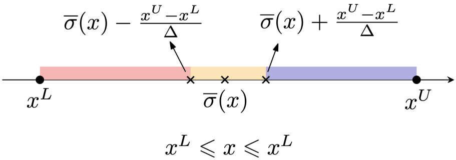
- 2 critical components: (i) how the piecewise linear functions are defined, and (ii) their refinement procedure.

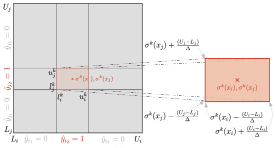




- Proves that the algorithm terminates after a finite number of steps $N_B(\epsilon)$
- Mixed Integer Polyhedron to compute lower bounds
- NLP solvers/oracles compute maximal linearization errors after split

- Split the domain into 3 sections: the size of the middle section depends on a user parameter Δ
- e.g. $\Delta = 8 \Rightarrow \frac{1}{4}$th of the domain





- Provides a proof of convergence when the number of breakpoints increases to infinity
- Open source solver Alpine.jl

Outlook

Context, motivation and state of the art

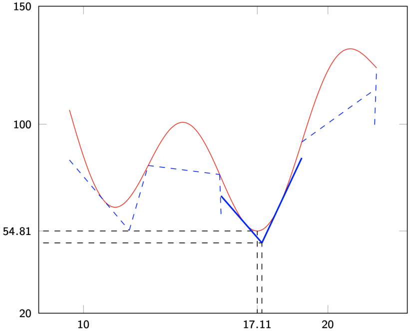
Algorithm proposed

Focus on the piecewise linearization subproblem

Conclusion

Our algorithm

- precision-driven refinement instead of domain-driven
- **not necessarily continuous** piecewise linear **lower bounding** approximations
- We do not require our piecewise linear approximation to interpolate the nonlinear function at breakpoints
- Solve a MIP that provides upper and lower bounds
- If we are unhappy with the result, we only tighten the necessary pieces



Our algorithm

Proposition

If the domain of each variable x_i is bounded within the interval $[l_i, u_i]$ with $l_i \leq u_i$, our algorithm ends in at most

$$N(\epsilon) = \left\lceil \log_2 \left(\frac{\epsilon_0}{\epsilon} \right) \right\rceil \sum_{i=1}^{n} \left\lceil \frac{u_i - l_i}{\delta} \right\rceil \qquad (4)$$

iterations and provides a solution x^* that if far from the optimal by at most ϵ.

Proposition

For every $\epsilon > 0$, $N(\epsilon) \leq O(\log(N_B(\epsilon)))$.

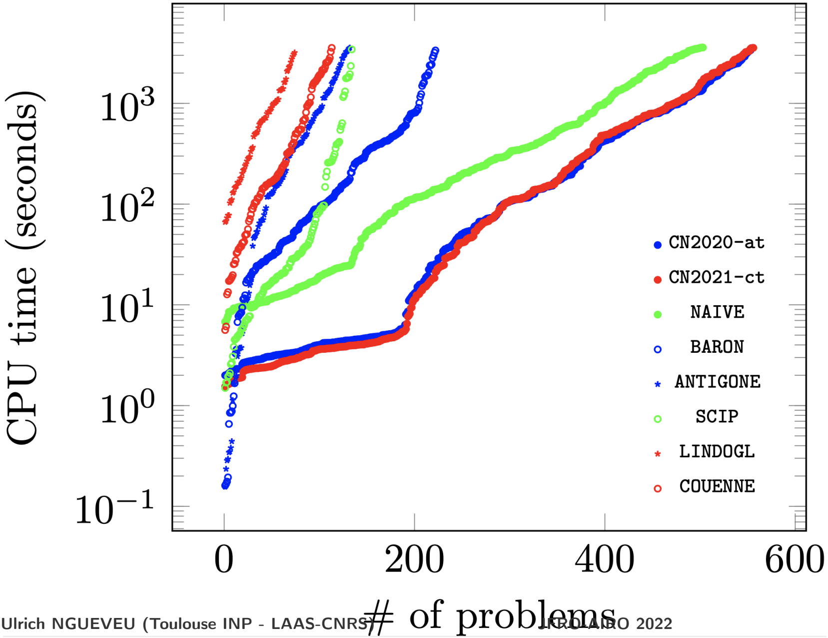
Our algorithm

- Keep the scope of the updates local $\Rightarrow$ tractability is not lost
- Refinement based on a target precision, not an arbitrary number of breakpoints
- Regions of the space that seem unpromising may never be tightened!
- On high dimensions this plays a critical role when going to very small numerical precisions

Applications

- We consider four classes of optimization problems
  - Capacitated facility location with nonlinear warehousing costs
  - Capacitated facility location with nonlinear assignment costs
  - Transportation problem with nonlinear transportation costs
  - Multi-comodity network design problem with nonlinear costs

Computational experiments



Computational experiments

Inst.	Alpine	CN2021	Inst.	Alpine	CN2021
2_1	13.43	0.15	3_1	TLIM	29.33
2_2	ERR	0.85	3_2	TLIM	1.1
2_3	25.84	1.05	3_3	TLIM	6.62
2_4	29.34	1.09	3_4	TLIM	4.45
2_5	TLIM	0.22	3_5	TLIM	4.53
2_6	2.13	0.27	3_6	TLIM	11.08
2_7	ERR	0.16	3_7	TLIM	19.37
2_8	1.84	0.16	3_8	TLIM	11.83
2_9	19.92	0.19	3_9	TLIM	2.29
2_10	ERR	0.93	3_10	TLIM	3.95
Moy.	15.4*	0.5	Moy.	TLIM	8.68

Table 1: CPU(s) d'Alpine vs our Algorithm on small instances (2x2 et 3x3) of the *transportation problem*

Outlook

Context, motivation and state of the art

Algorithm proposed

Focus on the piecewise linearization subproblem

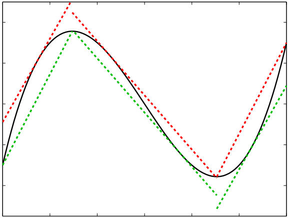
Conclusion

Key subproblem that needs to be solved efficiently

Precision-driven initialization and precision-driven refinement

⇓

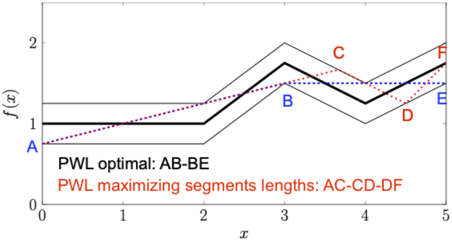
Minimize #pieces under bounded tolerance constraints



Min #pieces s.t. bounded tolerance constraints

Why is the Problem Hard?

- semi-infinite programming
- Maximizing the *domain length* of each segment is not optimal



Few pre-existing studies (Continuous PWL approximation):

- [Rosen and Pardalos, 1986], [Frenzen, Sasao and Butlerc, 2010], [Rebennack and Kallrath 2015], [Rebennack and Krasko 2019], [Kong, Maravelias 2020]

Non necessarily continuous (nnc) PWL

[Ngueveu - LAAS 2016, EJOR 2019] 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 $[x_i^{\min}, X_+]$.

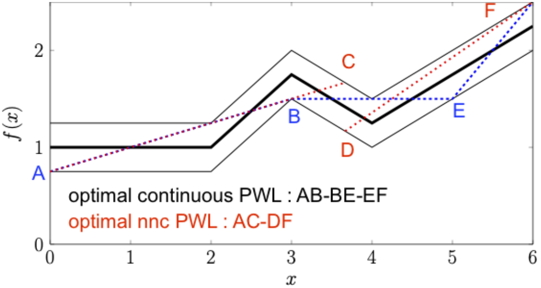
Non necessarily continuous (nnc) PWL

[Ngueveu - LAAS 2016, EJOR 2019] 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 $[x_i^{\min}, X_+]$.

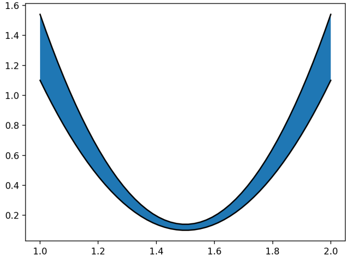
The greedy algorithm becomes optimal

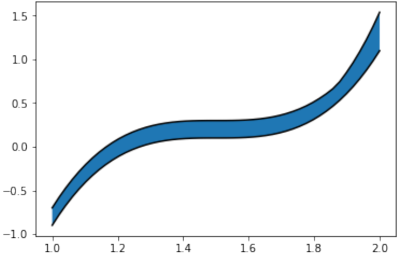


A geometric approach based on corridors

Definition (Corridor)

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





generalizes widely used pointwise error metrics (e.g. absolute or relative)

can be used to approximate, underestimate and overestimate functions

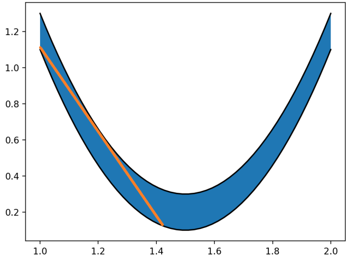
Codsi, Gendron, Ngueveu (2019-2022)

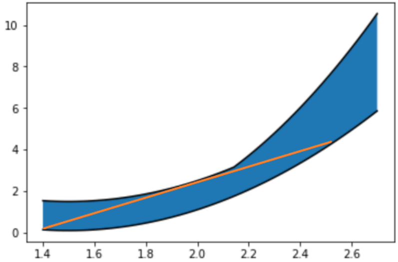
Convex corridor

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*



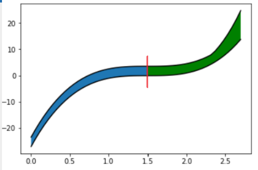


$\Rightarrow$ logarithmic convergence (for each segment)

Codsi, Gendron, Ngueveu (2019-2022)

Corridors without constant convexity

Splitting the corridor into sub-corridors convex or concave



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

$$n^* \leq n \leq n^* + \#\text{Sub-corridors} - 1$$

O'Rourke adaptation

based on function sampling and constraints on the line coefficient space

+ Exact
- Not as efficient

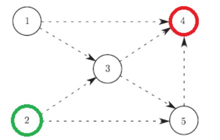
LinA Package `http://homepages.laas.fr/sungueve/LinA.html`

Computational evaluation of our PWL computations

$$f(x) = 2x^2 + x^3$$

δ	continuous		nnc			
	Exact		exact		heuristic	
	[RK2015]	[RK2019]	[Ng2019]	LinA	[Ng2019]	LinA
0.1	Minutes	24 s	115 s	2.9 s	11 s	0.008 s
0.05	Few days	107 s	88 s	3.0 s	17 s	0.01 s
0.005	–	35787 s	195 s	2.8 s	59 s	0.03 s

Multicommodity network design with congestion



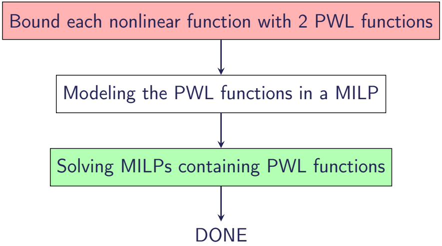
instance	literature	LinA+CPLEX
c36_8_8	21.53 s	14.59 s
c49_8_6	172.25 s	118.17 s
c50_8_6	2609.57 s	2575.08 s

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

Impact of arbitrary ϵ precisions

Strengths of the baseline method using LinA

- one-shot
- computational effort focussed on solving a single MILP



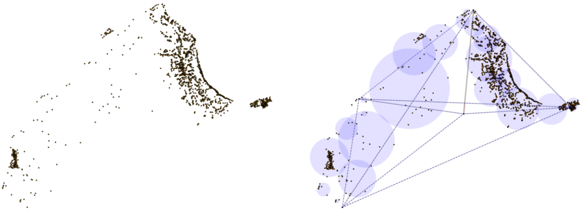
Weaknesses of the baseline method if a very small precision ϵ is requested

- size of the resulting MILP
- internal precisions of LinA or of the MILP solution method

$\Rightarrow$ Usually tractable only for very rough guarantees on very large problems

Decremental / Iterative sampling

Find a small sample whose optimal solution is that of the original problem, and build an optimal solution from enlarging that sample.



Used successfully for variety of problems, mostly related to clustering.

- minimax location-allocation problem: Chen and Handler (1987)
- interdiction problems with fortification: Lozano and Smith (2017)
- minimax diameter clustering problem: Daniel and Contardo (2018)
- p-center problem: Chen and Chen (2009), Contardo, Iori, Kramer (2019)
- *p*-dispersion problem : Contardo (INFORMS 2020)

Can similar ideas be applied to improve the baseline method ?

⇓

Adaptive refinement

⇓

Find a global optimal solution of the MINLP by solving a series of MIP relaxations with gradually increasing accuracy.

⇓

Iterative algorithm: see part 1 of the talk

Can similar ideas be applied to improve the baseline method ?

$\Downarrow$

Adaptive refinement

$\Downarrow$

Find a global optimal solution of the MINLP by solving a series of MIP relaxations with gradually increasing accuracy.

$\Downarrow$

Iterative algorithm: see part 1 of the talk

Outlook

Context, motivation and state of the art

Algorithm proposed

Focus on the piecewise linearization subproblem

Conclusion

Conclusion

In summary

- We present an iterative procedure to solve a class of linearly constrained MINLPs a predefined numerical precision
- Among the keys components, let us mention (i) local refinement, (ii) the ability to compute optimal piecewise linear functions with a bounded error
- Piecewise linearizations are computed using LinA available at `http://homepages.laas.fr/sungueve/LinA.html`

What next ?

- dedicated solution method
- extension to nonlinear functions not linearly separable ?

Useful Tools / Julia Packages

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

- link: `http://homepages.laas.fr/sungueve/LinA.html`
- `https://github.com/LICO-labs`
- 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)

A scalable algorithm for solving a class of separable nonconvex MINLPs to arbitrary numerical precision

Sandra Ulrich NGUEVEU

with the collaboration of Julien Codsi, Claudio Contardo, Bernard Gendron

November 30th,2022

http://homepages.laas.fr/sungueve
ngueveu@laas.fr