Blackbox optimization: Algorithms and applications

Sébastien Le Digabel

GERAD
GROUPE D'ÉTUDES ET DE RECHERCHE EN ANALYSE DES DÉCISIONS

POLYTECHNIQUE MONTRÉAL

2026-05-26

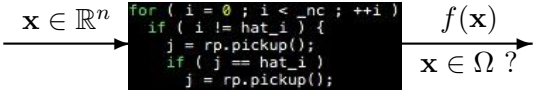
Presentation outline

Blackbox / Derivative-Free Optimization

We consider

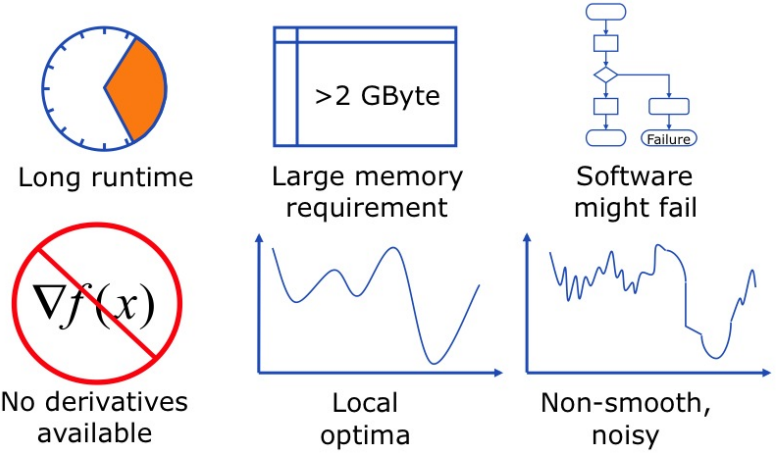
$$\min_{\mathbf{x}\in\Omega} \quad f(\mathbf{x})$$

where the evaluations of f and the functions defining Ω are the result of a computer simulation (a blackbox)



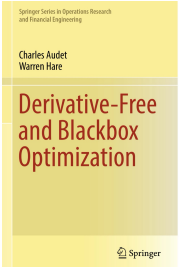
- Each call to the simulation may be expensive
- The simulation can fail
- Sometimes $f(\mathbf{x}) \neq f(\mathbf{x})$
- Derivatives are not available and cannot be approximated

Blackboxes as illustrated by a Boeing engineer



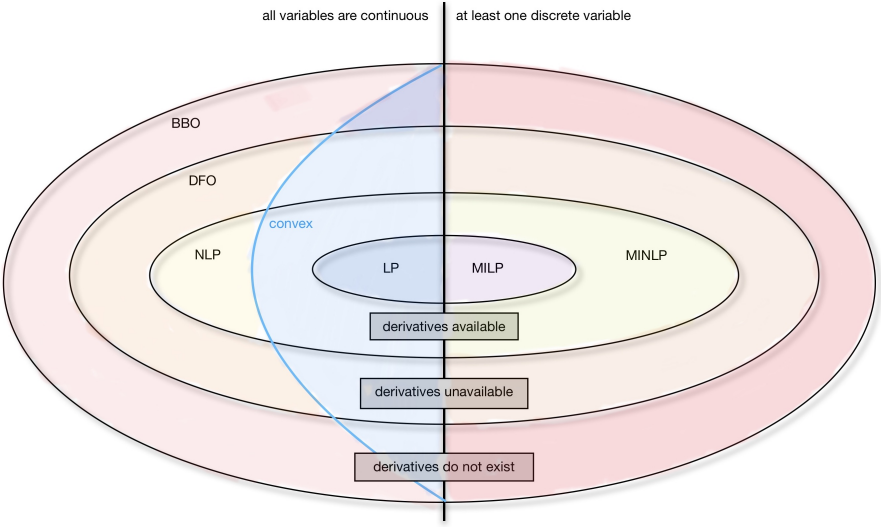
Terms

- *"Derivative-Free Optimization (DFO) is the mathematical study of optimization algorithms that do not use derivatives"* [Audet and Hare, 2017]
  - Optimization without using derivatives
  - Derivatives may exist but are not available
  - Obj./constraints may be analytical or given by a blackbox

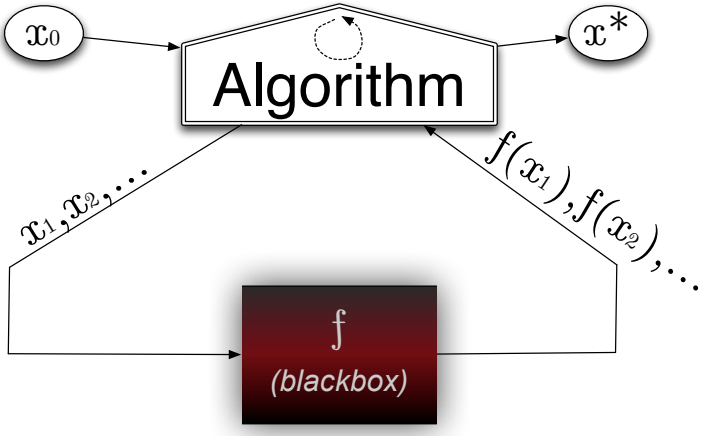


- *"Blackbox Optimization (BBO) is the study of design and analysis of algorithms that assume the objective and/or constraints functions are given by blackboxes"* [Audet and Hare, 2017]
  - A simulation, or a blackbox, is involved
  - Obj./constraints may be analytical functions of the outputs
  - Derivatives may be available (ex.: PDEs)
  - Sometimes referred as *Simulation-Based Optimization (SBO)*

Optimization: Global view



Typical setting



Unconstrained case, with one initial starting solution

Algorithms for blackbox optimization

A method for blackbox optimization should ideally:

- Be efficient given a limited budget of evaluations
- Be robust to noise and blackbox failures
- Natively handle general constraints
- Deal with multiobjective optimization
- Deal with integer and categorical variables
- Easily exploit parallelism
- Have a publicly available implementation
- Have convergence properties ensuring first-order local optimality in the smooth case – otherwise why using it on more complicated problems?

Families of methods

- *"Computer science"* methods:
  - Heuristics such as genetic algorithms
  - No convergence properties
  - Cost a **lot** of evaluations
  - Should be used only in last resort for desperate cases
- Statistical methods:
  - Design of experiments
  - Bayesian optimization: EGO algorithm based on surrogates and expected improvement
  - Still limited in terms of dimension
  - Does not natively handle constraints
  - Good to use these tools in conjunction with DFO methods
- Derivative-Free Optimization methods (DFO)

DFO methods

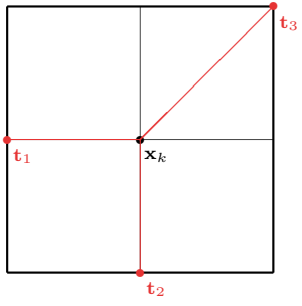
- **Model-based methods:**
  - Derivative-Free Trust-Region methods
  - Based on quadratic models or radial-basis functions
  - Use of a trust-region
  - Better for { DFO \ BBO }
  - Not resilient to noise and *hidden constraints*
  - Not easy to parallelize

- **Direct-search methods:**
  - Classical methods: Coordinate search, Nelder-Mead – the *other* simplex method
  - Modern methods: Generalized Pattern Search, Generating Set Search, Mesh Adaptive Direct Search (MADS)

So far, the size of the instances (variables and constraints) is typically limited to $\simeq 50$, and we target local optimization

MADS illustration with $n = 2$: Poll step

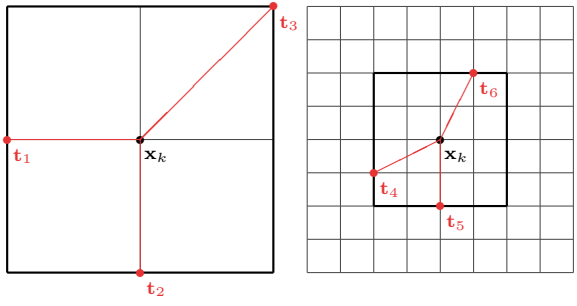
$$\delta^k = \Delta^k = 1$$



poll trial points=$\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\}$

MADS illustration with $n = 2$: Poll step

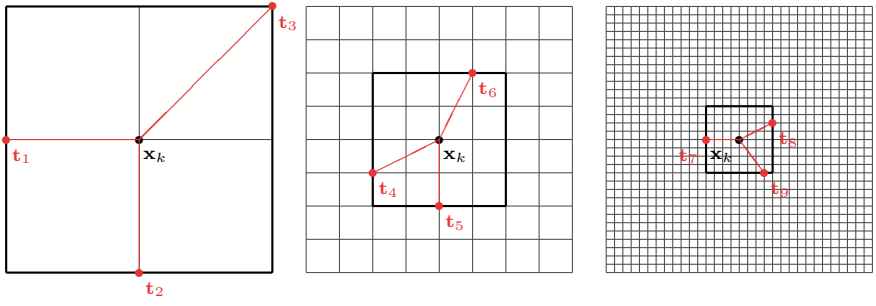
$\delta^k = \Delta^k = 1$ $\quad$ $\delta^{k+1} = 1/4$ $\quad$ $\Delta^{k+1} = 1/2$



poll trial points$=\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\}$ $\quad = \{\mathbf{t}_4, \mathbf{t}_5, \mathbf{t}_6\}$

MADS illustration with $n = 2$: Poll step

$\delta^k = \Delta^k = 1$ $\quad$ $\delta^{k+1} = 1/4$, $\Delta^{k+1} = 1/2$ $\quad$ $\delta^{k+2} = 1/16$, $\Delta^{k+2} = 1/4$



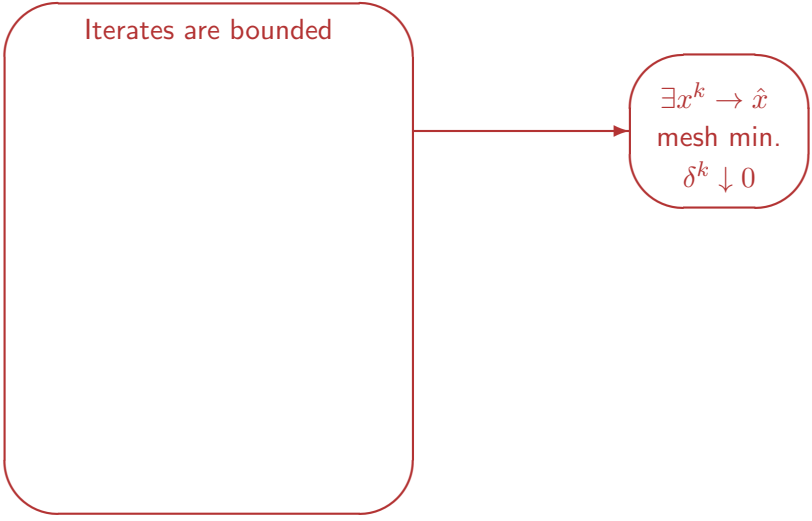
poll trial points$=\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\}$ $\quad = \{\mathbf{t}_4, \mathbf{t}_5, \mathbf{t}_6\}$ $\quad = \{\mathbf{t}_7, \mathbf{t}_8, \mathbf{t}_9\}$

[0] Initializations $(\mathbf{x}_0, \delta^0)$
[1] Iteration k
- **[1.1] Search** (flexible part)
  - select a finite number of mesh points
  - evaluate candidates opportunistically
- **[1.2] Poll** (if Search failed) ("rigid" part)
  - construct poll set $P_k = \{\mathbf{x}_k + \delta^k \mathbf{d} : \mathbf{d} \in D_k\}$
  - sort(P_k)
  - evaluate candidates opportunistically

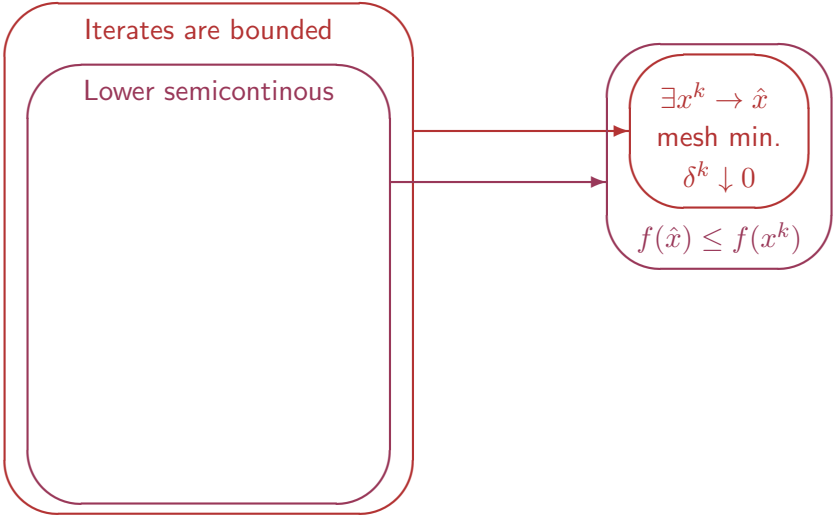
[2] Updates
- if success
  - $\mathbf{x}_{k+1} \leftarrow$ success point
  - increase δ^k
- else
  - $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k$
  - decrease δ^k
- $k \leftarrow k + 1$, stop or go to **[1]**

The MADS algorithm [Audet and Dennis, Jr., 2006]

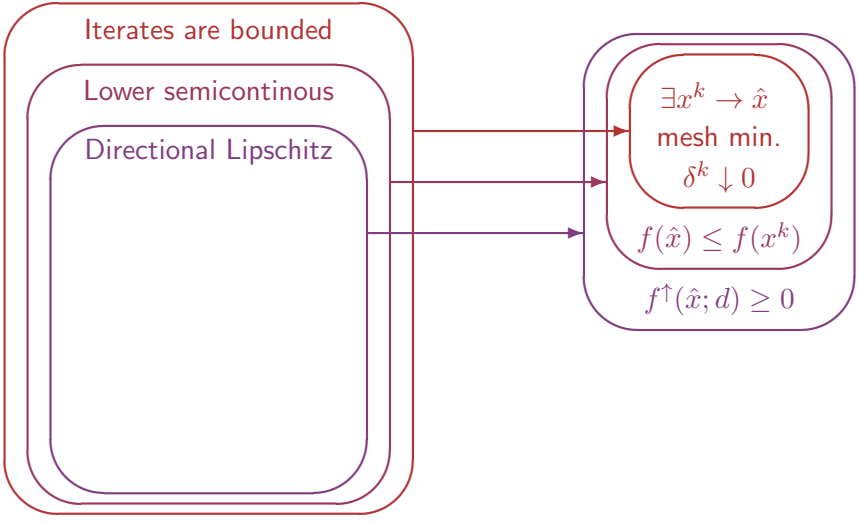
Hierarchical convergence



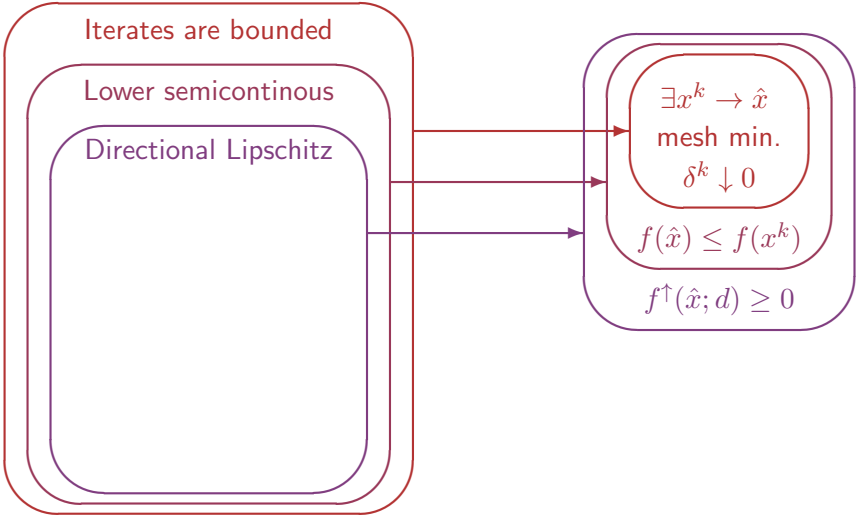
Hierarchical convergence



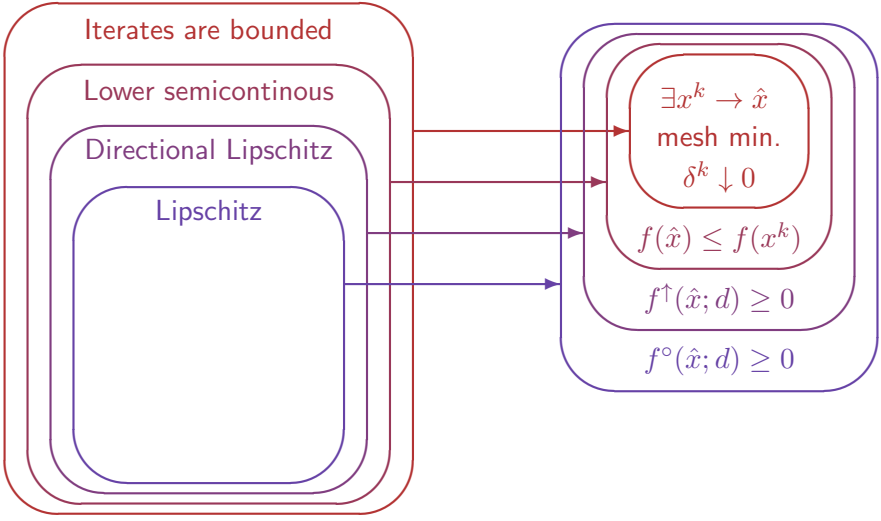
Hierarchical convergence



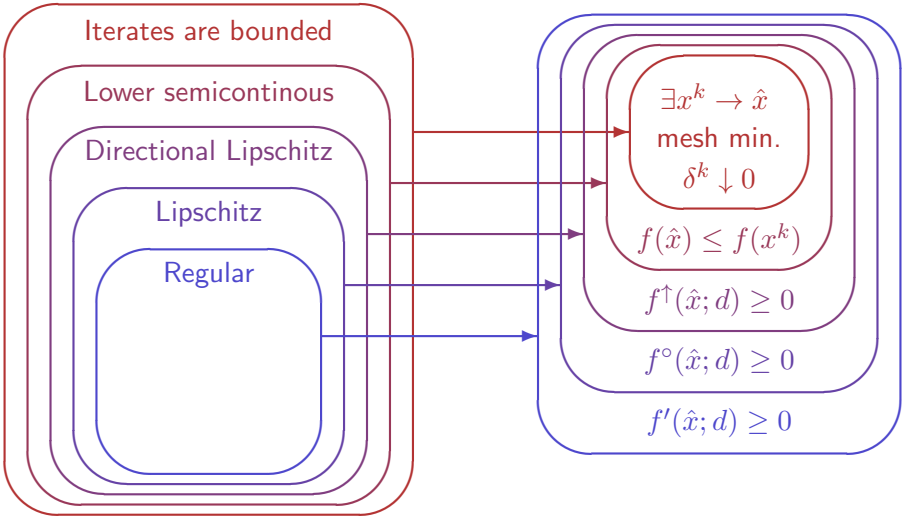
Hierarchical convergence



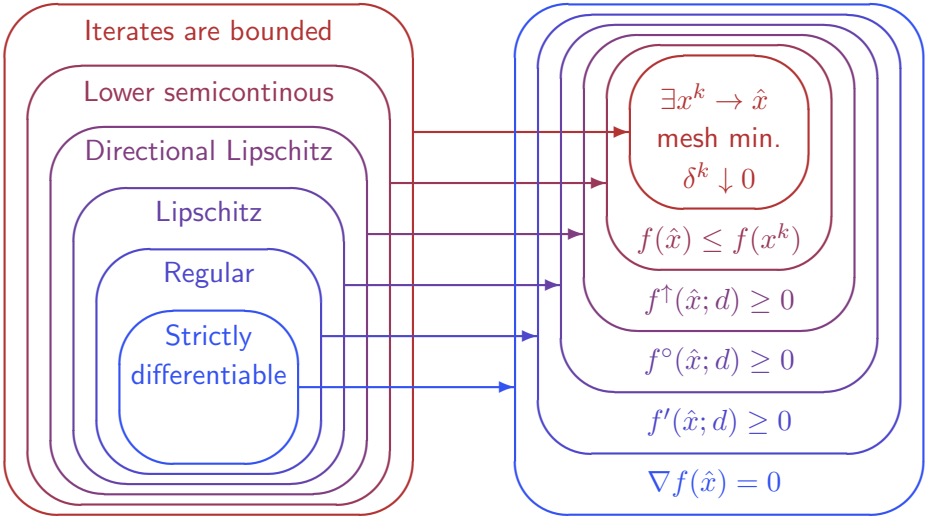
Hierarchical convergence



Hierarchical convergence



Hierarchical convergence



Features and evolution of MADS [Audet and Dennis, Jr., 2006]

- Constraints handling [Audet and Dennis, Jr., 2009, Audet et al., 2015, Le Digabel and Wild, 2024]
- Core of MADS [Abramson et al., 2009, Audet et al., 2014, Audet et al., 2016, Audet et al., 2025a]
- Global optimization [Audet et al., 2008a]
- Parallelism [Audet et al., 2008b, Le Digabel et al., 2010]
- Sensitivity analysis [Audet et al., 2012]
- Surrogates [Conn and Le Digabel, 2013, Gramacy and Le Digabel, 2015, Talgorn et al., 2015, Audet et al., 2018b, Talgorn et al., 2018, Amaioua et al., 2018, Audet et al., 2022b]
- Hyperparameters tuning [Lakhmiri et al., 2021, Lakhmiri and Le Digabel, 2022]
- Discrete/Categorical/Meta variables [Audet et al., 2019, Audet et al., 2023, Hallé-Hannan et al., 2025a, Hallé-Hannan et al., 2025b, Audet et al., 2025b]
- Multiobjective optimization [Audet et al., 2008c, Bigeon et al., 2021, Audet et al., 2021a, Bigeon et al., 2024, Le Digabel et al., 2025]
- Multifidelity [Alarie et al., 2022, Alarie et al., 2025]
- Stochastic blackboxes [Audet et al., 2018a, Alarie et al., 2021, Audet et al., 2021b, Audet et al., 2021b, Dzahini, 2022, Dzahini et al., 2023]
- `NOMAD` [Le Digabel, 2011, Audet et al., 2022a]
- Large dimension

Constraints – with taxonomy of [Le Digabel and Wild, 2024]

Domain: $\Omega = \{\mathbf{x} \in \mathcal{X} : c_j(\mathbf{x}) \leq 0, j \in J\} \subset \mathbb{R}^n$

- $\mathcal{X}$ corresponds to unrelaxable constraints

  Cannot be violated;

  Example: $x > 0$ when $\log x$ is used inside the simulation

Constraints – with taxonomy of [Le Digabel and Wild, 2024]

Domain: $\Omega = \{\mathbf{x} \in \mathcal{X} : c_j(\mathbf{x}) \leq 0, j \in J\} \subset \mathbb{R}^n$

- $\mathcal{X}$ corresponds to unrelaxable constraints
- $c_j(\mathbf{x}) \leq 0$: Relaxable and quantifiable constraints

  May be violated at intermediate designs

  $c_j(\mathbf{x})$ measures the violation

  Example: cost $\leq$ budget

Constraints – with taxonomy of [Le Digabel and Wild, 2024]

Domain: $\Omega = \{\mathbf{x} \in \mathcal{X} : c_j(\mathbf{x}) \leq 0, j \in J\} \subset \mathbb{R}^n$

- $\mathcal{X}$ corresponds to unrelaxable constraints
- $c_j(\mathbf{x}) \leq 0$: Relaxable and quantifiable constraints
- Hidden constraints

  when the simulation fails, even for points in Ω

  Example:

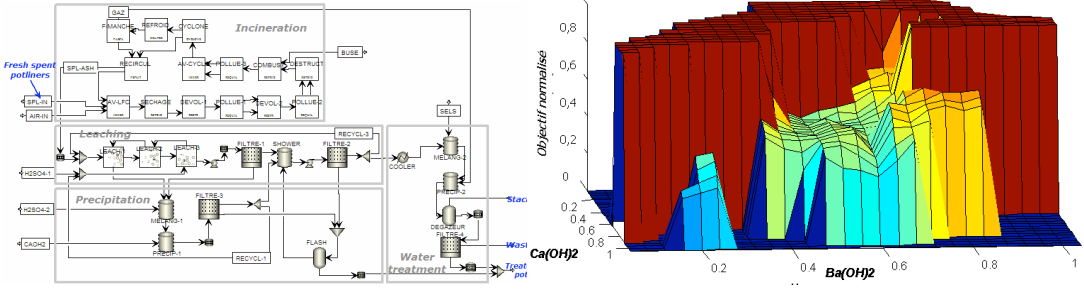
  ```
  Segmentation fault
  Bus error
  ERROR 42
  DIVISION BY ZERO
  ```

Constraints – with taxonomy of [Le Digabel and Wild, 2024]

Domain: $\Omega = \{\mathbf{x} \in \mathcal{X} : c_j(\mathbf{x}) \leq 0, j \in J\} \subset \mathbb{R}^n$

- $\mathcal{X}$ corresponds to unrelaxable constraints
- $c_j(\mathbf{x}) \leq 0$: Relaxable and quantifiable constraints
- Hidden constraints

Example: Chemical process:



7 variables, 4 constraints. The ASPEN software fails on 43% of the calls

Two strategies to deal with constraints

- Extreme barrier (EB)

Treats the problem as being unconstrained,
by replacing the objective function $f(\mathbf{x})$ by

$$f_\Omega(\mathbf{x}) := \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega \\ \infty & \text{otherwise} \end{cases}$$

The problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f_\Omega(\mathbf{x})$$

is then solved.
Remark: this strategy can also be applied to a priori constraints in order to avoid the costly evaluation of $f(\mathbf{x})$

Two strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)

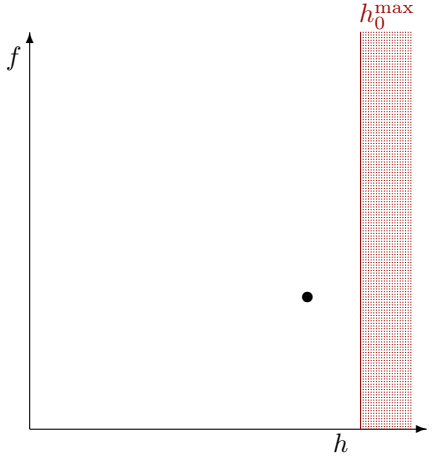
  Defined for relaxable and quantifiable constraints.
  As in the filter methods of Fletcher and Leyffer, it uses the non-negative constraint violation function $h : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$

$$h(\mathbf{x}) := \begin{cases} \sum_{j \in J} \left(\max(c_j(\mathbf{x}), 0)\right)^2 & \text{if } \mathbf{x} \in \mathcal{X} \\ \infty & \text{otherwise} \end{cases}$$

  At iteration k, points with $h(\mathbf{x}) > h_k^{\max}$ are rejected by the algorithm, and $h_k^{\max}$ decreases toward 0 as $k \to \infty$

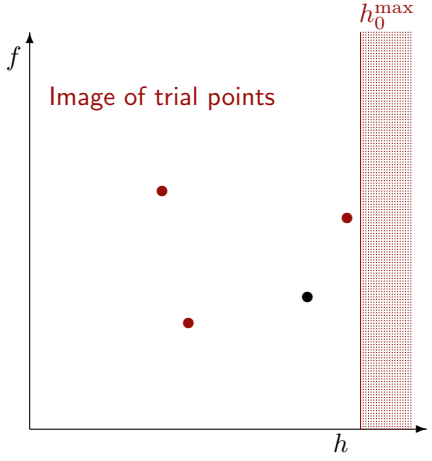
Two strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)



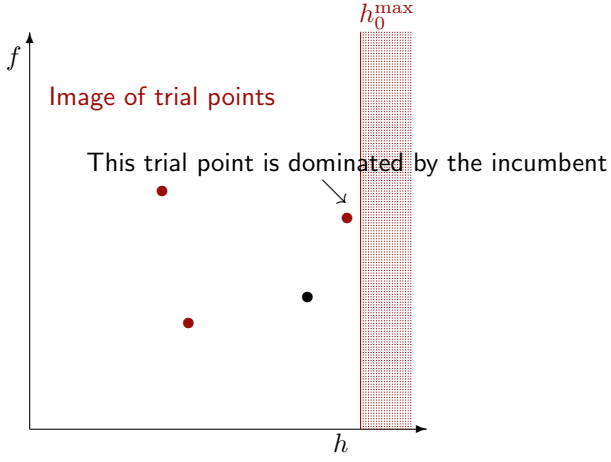
Two strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)



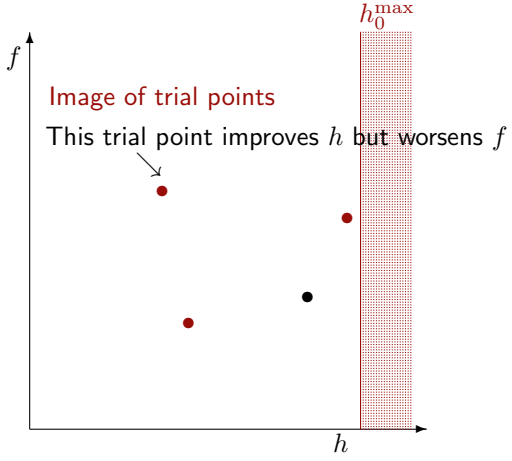
Two strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)



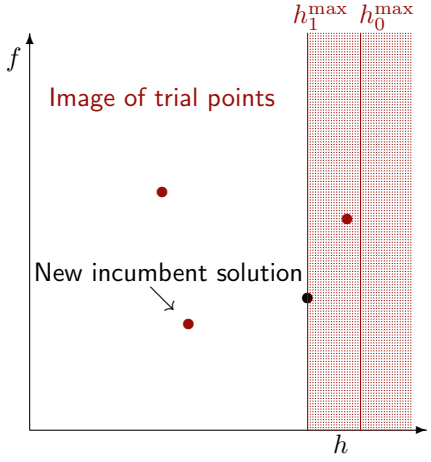
Two strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)



Two strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)



NOMAD (Nonlinear Optimization with MADS)

- C++ implementation of the MADS algorithm [Audet and Dennis, Jr., 2006]
- Standard C++. Runs on Linux, Mac OS X and Windows
- Parallel versions
- `PyNomad.py` and `NOMAD.jl` versions
- Open and free – LGPL license
- Download at `https://www.gerad.ca/nomad` or `https://github.com/bbopt/nomad`
- Support at `nomad@gerad.ca`
- Related articles in TOMS [Le Digabel, 2011] and [Audet et al., 2022a]

Main functionalities (1/2)

- Single or biobjective optimization
- Variables:
  - Continuous, integer, binary, categorical, granular
  - Periodic
  - Fixed
  - Groups of variables
- Searches:
  - Latin-Hypercube
  - Variable Neighborhood Search
  - Nelder-Mead Search
  - Quadratic models
  - Statistical surrogates
  - User search

Main functionalities (2/2)

- Constraints treated with 4 different methods:
  - Progressive Barrier (default)
  - Extreme Barrier
  - Progressive-to-Extreme Barrier
  - Filter method
- Several direction types:
  - Coordinate directions
  - LT-MADS
  - OrthoMADS
  - Hybrid combinations
- Sensitivity analysis

→ default values for all parameters

→ all items correspond to published or submitted papers

Blackbox conception (batch mode)

- Command-line program that takes in argument a file containing $\mathbf{x}$, and displays the values of $f(\mathbf{x})$ and the $c_j(\mathbf{x})$'s
- Can be coded in any language
- Typically: `> bb.exe x.txt` displays `f c1 c2` (objective and two constraints)

Run NOMAD

```
> nomad parameters.txt
```

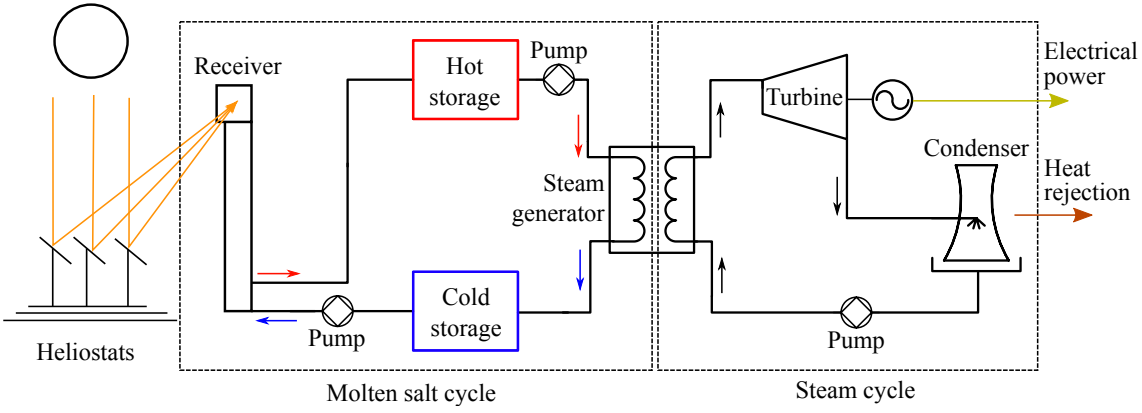
```
[iota ~/Desktop/2018_UQAC_NOMAD/demo_NOMAD/mac] > ../nomad.3.8.1/bin/nomad parameters.txt

NOMAD - version 3.8.1 has been created by {
        Charles Audet       - Ecole Polytechnique de Montreal
        Sebastien Le Digabel - Ecole Polytechnique de Montreal
        Christophe Tribes    - Ecole Polytechnique de Montreal
}

The copyright of NOMAD - version 3.8.1 is owned by {
        Sebastien Le Digabel - Ecole Polytechnique de Montreal
        Christophe Tribes    - Ecole Polytechnique de Montreal
}

NOMAD v3 has been funded by AFOSR, Exxon Mobil, Hydro Québec, Rio Tinto and
IVADO.

NOMAD v3 is a new version of NOMAD v1 and v2. NOMAD v1 and v2 were created
and developed by Mark Abramson, Charles Audet, Gilles Couture, and John E.
Dennis Jr., and were funded by AFOSR and Exxon Mobil.

License   : '$NOMAD_HOME/src/lgpl.txt'
User guide: '$NOMAD_HOME/doc/user_guide.pdf'
Examples  : '$NOMAD_HOME/examples'
Tools     : '$NOMAD_HOME/tools'

Please report bugs to nomad@gerad.ca

Seed: 0

MADS run {

        BBE     OBJ

        4       0.0000000000
        21      -1.0000000000
        23      -3.0000000000
        51      -4.0000000000
        563     -4.0000000000

} end of run (mesh size reached NOMAD precision)

blackbox evaluations                : 563
best infeasible solution (min. violation): ( 1.000000013 1.000000048 0.9999999797 0.999999992 -4 ) h=1.10134e-13 f=-4
best feasible solution              : ( 1 1 1 1 -4 ) h=0 f=-4
```

CSP power plant with molten salt thermal energy storage

- Work published in [Andrés-Thió et al., 2025]
- A large number of mirrors (heliostats) reflects solar radiation on a receiver at the top of a tower
- The heat collected from the concentrated solar flux is removed from the receiver by a stream of molten salt
- Hot molten salt is then used to feed thermal power to a conventional power block
- The photo shows the Thémis CSP power plant, the first built with this design

Source: https://commons.wikimedia.org/wiki/File:Themis_2.jpg

System dynamics



Features for BBO benchmarking

- Several numerical methods: real-world blackbox
- Reproducibility across all platforms
- Continuous and discrete variables
- Different types of constraints (quantifiable, relaxable, a priori, hidden)
- Stochastic and deterministic outputs
- Static surrogates with variable fidelity
- Number of replications is controllable

Ten instances

Instance	# of variables		n	# of obj.	# of constraints		m	# of stoch. outputs	Static
	cont.	discr. (cat.)		p	simu.	a priori (lin.)		(obj. or constr.)	surrogate
solar1	8	1 (0)	9	1	2	3 (2)	5	1	no
solar2[1]	12	2 (0)	14	1	9	4 (2)	13	3	yes
solar3	17	3 (1)	20	1	8	5 (3)	13	5	yes
solar4	22	7 (1)	29	1	9	7 (5)	16	6	yes
solar5	14	6 (1)	20	1	8	4 (3)	12	0	no
solar6	5	0 (0)	5	1	6	0 (0)	6	0	no
solar7	6	1 (0)	7	1	4	2 (1)	6	3	yes
solar8	11	2 (0)	13	2	4	5 (3)	9	3	yes
solar9	22	7 (1)	29	2	10	7 (5)	17	6	yes
solar10[2]	5	0 (0)	5	1	0	0 (0)	0	0	yes

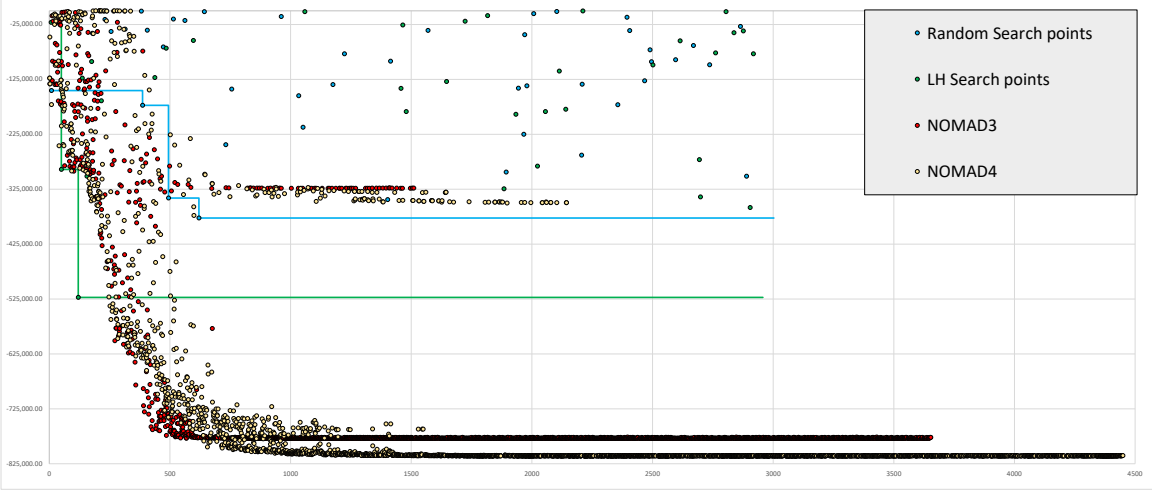
[1] analytic objective

[2] unconstrained

Feasibility with sampling and NOMAD

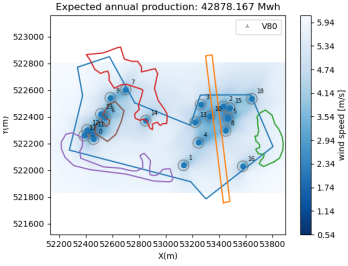
Instance	LH search (10k points)		NOMAD3		
	satisf. ap constr.	feas. pts	satisf. ap constr.	feas. pts	number of eval.
solar1	30%	0.35%	96%	74%	3,792
solar2	0%	0%	97%	0%	1,635
solar3	0.49%	0%	99%	9%	30,525
solar4	0%	0%	83%	0%	44,303
solar5	0%	0%	83%	59%	3,405
solar6	90%	5%	99%	0%	3,539
solar7	2%	1%	74%	72%	2,224
solar8	1%	0.03%			
solar9	1%	0%			

Optimization on solar1



Wind farm layout optimization

- Joint work with Joséphine Gobert, Léon Biner, Antoine Lesage-Landry, Merlin Keller and Julien Pelamatti.
- **Context:** Minimizing the Levelized Cost of Energy requires maximizing the Annual Energy Production (AEP), which depends on turbine layout
- **Problem:** Given a wind site and a fixed number of turbines, find the layout (positions) that maximizes the AEP, subject to placement and spacing constraints
- **Simulator:** `PyWake` [Pedersen et al., 2023] (DTU), an open-source wind farm flow simulator

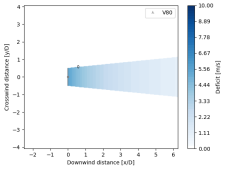


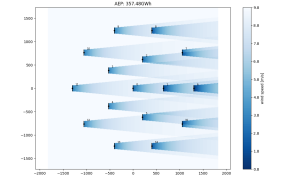
Wind farm model: PyWake and wind speed modeling

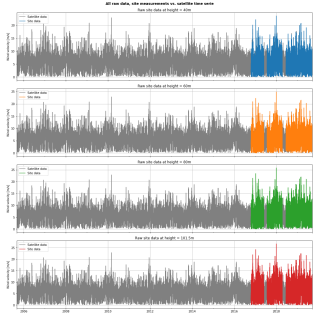
- Inputs: onsite wind speed time-series (uncertain) and turbine layout (design variable)
- Output: power production time-series
- Simulator: PyWake + power curve of the turbine

PyWake:

- Superposition of wake deficit functions
- Static: each time-step treated independently
- Faster (but less accurate) than CFD solvers







BBO Model: Objective, Variables, and Constraints

Decision variables:

- Positions (coordinates) of the turbines
- Continuous variables only

Objective: maximize the AEP, computed as a weighted sum of turbine power outputs over all wind direction and speed combinations, using probabilities from a wind rose

Constraints:

1. *Placement in constructible zone:*
   - A priori, binary: each turbine position must be admissible (checked before `PyWake` call)
   - Or relaxable/quantifiable: sum of distances to the constructible zone
2. *Minimal spacing between turbines:*
   - Each pair of turbines must be at least 2 diameters apart
   - A priori and quantifiable; treated as relaxable or unrelaxable

NOMAD vs Monte Carlo

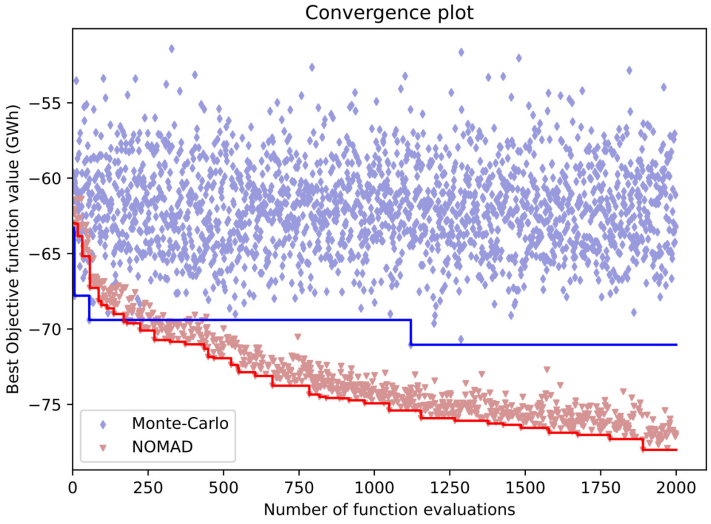
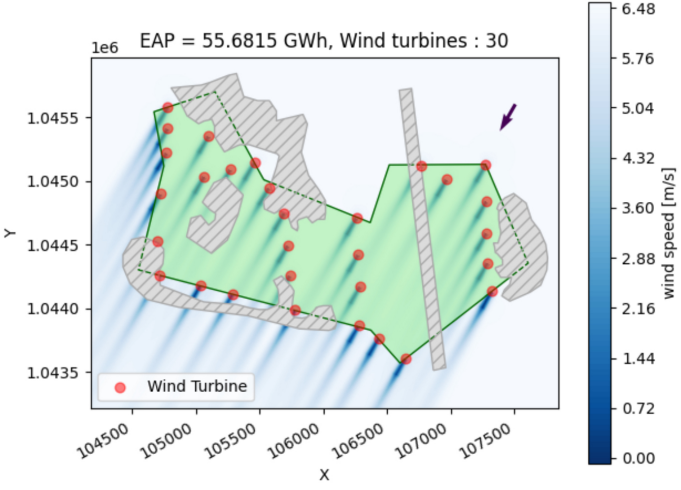
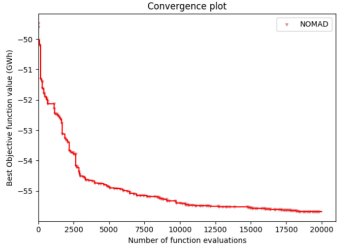


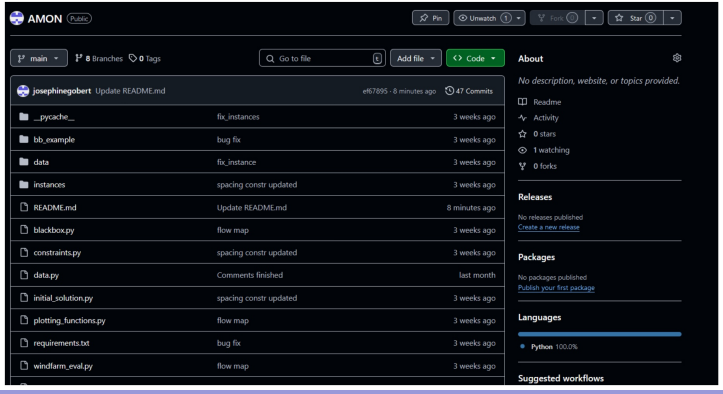
Illustration of a solution



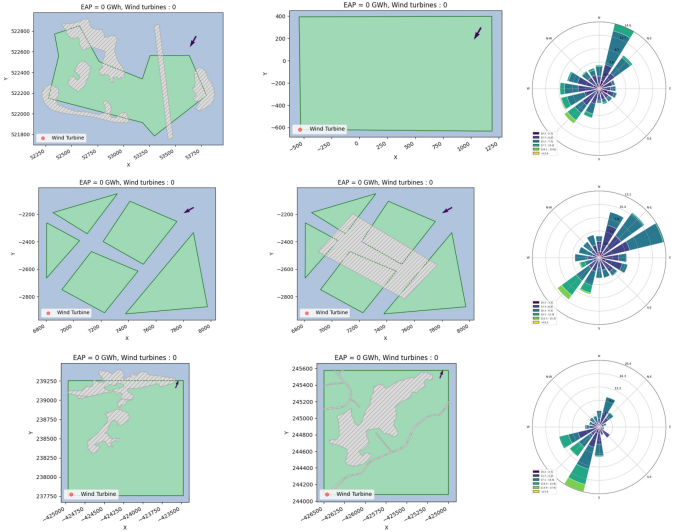


The amon package

- Available at https://github.com/josephinegobert/AMON
- Several instances have been defined as realistic benchmarking instances for the BBO community

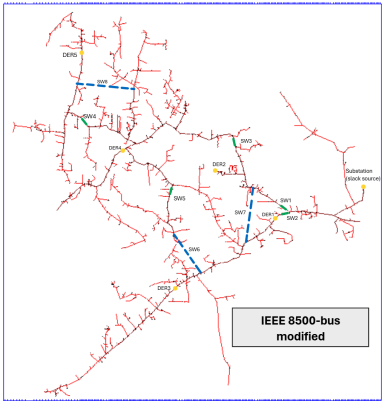


The amon package: Instances



Distribution network reconfiguration with DERs[3]

- PhD project of Christina Soldati
- Collaboration with Miguel Anjos and Antoine Lesage-Landry
- Presentation at SIAM OP26 (Wednesday 9:15am)
- **Problem:** Unbalanced phases in power distribution networks, amplified by DER integration, increase power losses
- **Goal:** Reconfigure the network topology (switch states) and DER injections to minimize total generation losses

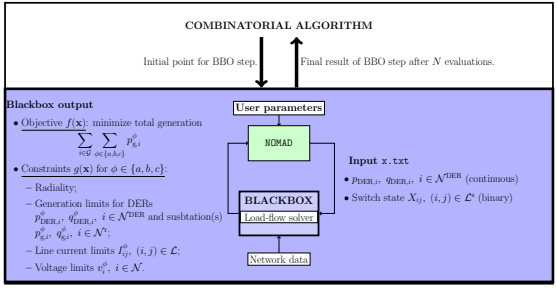


[3]Distributed Energy Resources

The BBO Model

Decision variables :

- Continuous: active/reactive DER power injections
- Binary: switch states: drives the combinatorial complexity



Objective: minimize total active power generation (losses)

Constraints: via the EMTP[4] load-flow solver:

- Connectivity
- Radiality
- Voltage limits
- Line current limits
- DER and substation generation limits

[4] [Mahseredjian et al., 2007]

Resolution Methods

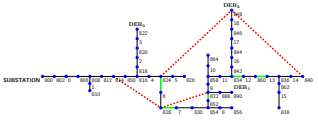
NOMAD is the core of all methods. The binary variables are handled by combining BBO with combinatorial strategies:

- BBO
- BBO-VNS
- BBO-B&B
- BBO-VNS-B&B
- BBO-B&B-VNS
- L-BBO-VNS
- L-BBO-MST

- VNS: randomly permutes binary variables (switch states) and re-optimizes continuous variables with BBO
- B&B: fixes one binary variable at a time, optimizes the rest with BBO
- List-based: pre-generates feasible topologies (spanning trees); (VNS or MST)-inspired heuristic selects the next topology

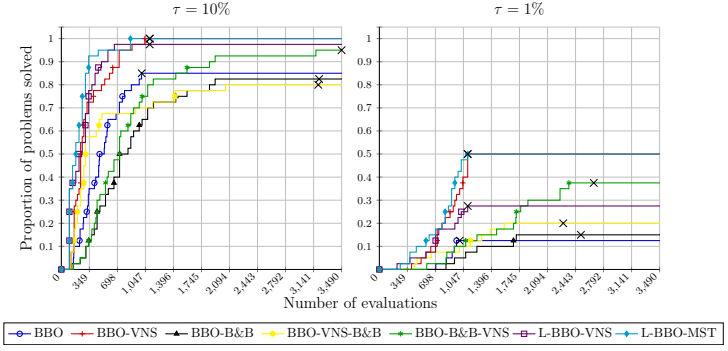
→ BBO serves as a warm start and baseline; combinatorial methods improve exploration of the binary space

IEEE 34-bus results

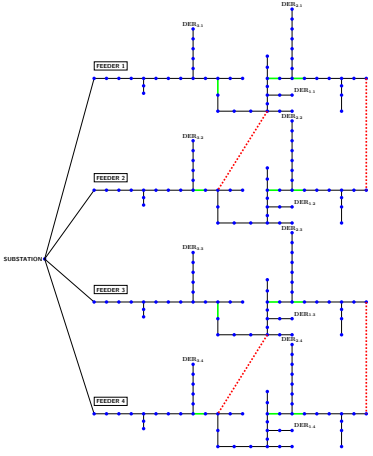


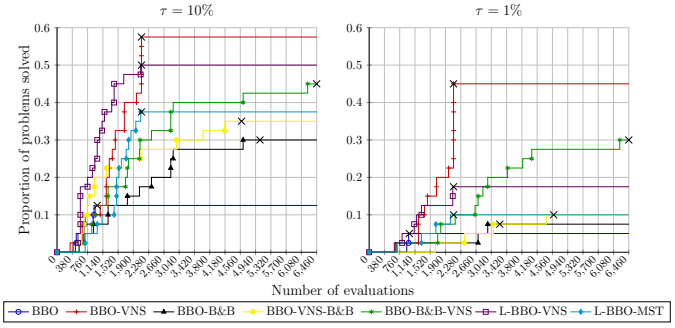
15 variables:

- 6 continuous
- 9 binary



136-bus results

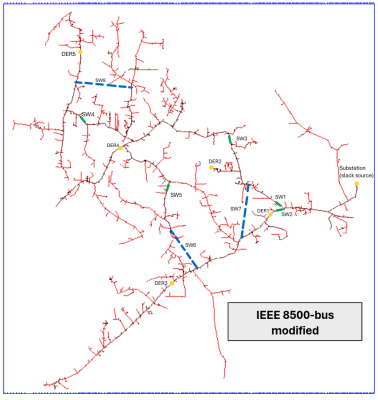


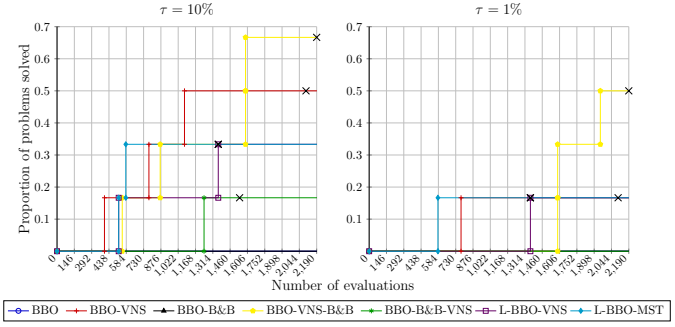


40 variables:

- 24 continuous
- 16 binary

IEEE 8500-bus results





18 variables:

- 10 continuous
- 8 binary

Summary

- Blackbox optimization motivated by industrial applications
- Algorithmic features backed by mathematical convergence analyses and published in optimization journals
- NOMAD: Software package implementing MADS
- Open source; LGPL license
- Fast support at nomad@gerad.ca
- NOMAD has become a baseline for benchmarking DFO algorithms
- MADS-related talks at SIAM OP26:
  - MS126: **Christina Soldati**, *Blackbox Optimization for Distribution Network Reconfiguration with Distributed Energy Resources*
  - MS191: **Tudor Buhut**, *Chance-constrained Blackbox Optimization*
  - MS232: **Charles Audet**, *Adaptive Direct Search Algorithms for Constrained Optimization*

References I


Abramson, M., Audet, C., Dennis, Jr., J., and Le Digabel, S. (2009).
OrthoMADS: A Deterministic MADS Instance with Orthogonal Directions.
SIAM Journal on Optimization, 20(2):948–966.


Alarie, S., Audet, C., Bouchet, P.-Y., and Le Digabel, S. (2021).
Optimisation of stochastic blackboxes with adaptive precision.
SIAM Journal on Optimization, 31(4):3127–3156.


Alarie, S., Audet, C., Diago, M., Le Digabel, S., and Lebeuf, X. (2025).
Inter-DS: A cost saving algorithm for expensive constrained multi-fidelity blackbox optimization.
Computational Optimization and Applications, 90(3):607–629.


Alarie, S., Audet, C., Jacquot, P., and Le Digabel, S. (2022).
Hierarchically constrained blackbox optimization.
Operations Research Letters, 50(5):446–451.


Amaioua, N., Audet, C., Conn, A., and Le Digabel, S. (2018).
Efficient solution of quadratically constrained quadratic subproblems within a direct-search algorithm.
European Journal of Operational Research, 268(1):13–24.


Andrés-Thió, N., Audet, C., Diago, M., Gheribi, A., Le Digabel, S., Lebeuf, X., Lemyre Garneau, M., and Tribes, C. (2025).
Solar: a solar thermal power plant simulator for blackbox optimization benchmarking.
Optimization and Engineering, 26(3):1815–1861.

References II

Audet, C., Béchard, V., and Le Digabel, S. (2008a).
Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search.
Journal of Global Optimization, 41(2):299–318.

Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., and Salomon, L. (2021a).
Performance indicators in multiobjective optimization.
European Journal of Operational Research, 292(2):397–422.
Invited Review.

Audet, C. and Dennis, Jr., J. (2006).
Mesh Adaptive Direct Search Algorithms for Constrained Optimization.
SIAM Journal on Optimization, 17(1):188–217.

Audet, C. and Dennis, Jr., J. (2009).
A Progressive Barrier for Derivative-Free Nonlinear Programming.
SIAM Journal on Optimization, 20(1):445–472.

Audet, C., Dennis, Jr., J., and Le Digabel, S. (2008b).
Parallel Space Decomposition of the Mesh Adaptive Direct Search Algorithm.
SIAM Journal on Optimization, 19(3):1150–1170.

Audet, C., Dennis, Jr., J., and Le Digabel, S. (2012).
Trade-off studies in blackbox optimization.
Optimization Methods and Software, 27(4–5):613–624.

References III


Audet, C., Denorme, T., Diouane, Y., Le Digabel, S., and Tribes, C. (2025a).
Adaptive direct search algorithms for constrained optimization.
Technical Report G-2025-53, Les cahiers du GERAD.


Audet, C., Diouane, Y., Hallé-Hannan, E., Le Digabel, S., and Tribes, C. (2025b).
CatMADS: Mesh Adaptive Direct Search for constrained blackbox optimization with categorical variables.
Technical Report G-2025-42, Les cahiers du GERAD.


Audet, C., Dzahini, K., Kokkolaras, M., and Le Digabel, S. (2021b).
Stochastic mesh adaptive direct search for blackbox optimization using probabilistic estimates.
Computational Optimization and Applications, 79(1):1–34.


Audet, C., Hallé-Hannan, E., and Le Digabel, S. (2023).
A General Mathematical Framework for Constrained Mixed-variable Blackbox Optimization Problems with Meta and Categorical Variables.
Operations Research Forum, 4(12).


Audet, C. and Hare, W. (2017).
Derivative-Free and Blackbox Optimization.
Springer Series in Operations Research and Financial Engineering. Springer, Cham, Switzerland.


Audet, C., Ianni, A., Le Digabel, S., and Tribes, C. (2014).
Reducing the Number of Function Evaluations in Mesh Adaptive Direct Search Algorithms.
SIAM Journal on Optimization, 24(2):621–642.

References IV

Audet, C., Ihaddadene, A., Le Digabel, S., and Tribes, C. (2018a).
Robust optimization of noisy blackbox problems using the Mesh Adaptive Direct Search algorithm.
Optimization Letters, 12(4):675–689.

Audet, C., Kokkolaras, M., Le Digabel, S., and Talgorn, B. (2018b).
Order-based error for managing ensembles of surrogates in mesh adaptive direct search.
Journal of Global Optimization, 70(3):645–675.

Audet, C., Le Digabel, S., and Peyrega, M. (2015).
Linear equalities in blackbox optimization.
Computational Optimization and Applications, 61(1):1–23.

Audet, C., Le Digabel, S., Rochon Montplaisir, V., and Tribes, C. (2022a).
Algorithm 1027: NOMAD version 4: Nonlinear optimization with the MADS algorithm.
ACM Transactions on Mathematical Software, 48(3):35:1–35:22.

Audet, C., Le Digabel, S., and Saltet, R. (2022b).
Quantifying uncertainty with ensembles of surrogates for blackbox optimization.
Computational Optimization and Applications, 83:29–66.

Audet, C., Le Digabel, S., and Tribes, C. (2016).
Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization.
Optimization and Engineering, 17(2):333–358.

References V

Audet, C., Le Digabel, S., and Tribes, C. (2019).
The Mesh Adaptive Direct Search Algorithm for Granular and Discrete Variables.
SIAM Journal on Optimization, 29(2):1164–1189.

Audet, C., Savard, G., and Zghal, W. (2008c).
Multiobjective Optimization Through a Series of Single-Objective Formulations.
SIAM Journal on Optimization, 19(1):188–210.

Bigeon, J., Le Digabel, S., and Salomon, L. (2021).
DMulti-MADS: Mesh adaptive direct multisearch for bound-constrained blackbox multiobjective optimization.
Computational Optimization and Applications, 79(2):301–338.

Bigeon, J., Le Digabel, S., and Salomon, L. (2024).
Handling of constraints in multiobjective blackbox optimization.
Computational Optimization and Applications, 89(1):69–113.

Conn, A. and Le Digabel, S. (2013).
Use of quadratic models with mesh-adaptive direct search for constrained black box optimization.
Optimization Methods and Software, 28(1):139–158.

Dzahini, K. (2022).
Expected complexity analysis of stochastic direct-search.
Computational Optimization and Applications, 81(1):179–200.

References VI

Dzahini, K., Kokkolaras, M., and Le Digabel, S. (2023).
Constrained stochastic blackbox optimization using a progressive barrier and probabilistic estimates.
Mathematical Programming, 198(1):675–732.

Gramacy, R. and Le Digabel, S. (2015).
The mesh adaptive direct search algorithm with treed Gaussian process surrogates.
Pacific Journal of Optimization, 11(3):419–447.

Hallé-Hannan, E., Audet, C., Diouane, Y., Le Digabel, S., and Saves, P. (2025a).
A distance for mixed-variable and hierarchical domains with meta variables.
Neurocomputing, 653:131208.

Hallé-Hannan, E., Audet, C., Diouane, Y., Le Digabel, S., and Tribes, C. (2025b).
Cat-Suite: A collection of optimization problems with categorical and quantitative variables for benchmarking.
Technical Report G-2025-39, Les cahiers du GERAD.

Lakhmiri, D. and Le Digabel, S. (2022).
Use of static surrogates in hyperparameter optimization.
Operations Research Forum, 3(11).

Lakhmiri, D., Le Digabel, S., and Tribes, C. (2021).
HyperNOMAD: Hyperparameter Optimization of Deep Neural Networks Using Mesh Adaptive Direct Search.
ACM Transactions on Mathematical Software, 47(3).

References VII


Le Digabel, S. (2011).
Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm.
ACM Transactions on Mathematical Software, 37(4):44:1–44:15.


Le Digabel, S., Abramson, M., Audet, C., and Dennis, Jr., J. (2010).
Parallel Versions of the MADS Algorithm for Black-Box Optimization.
In *Optimization days*, Montréal.
Slides available at `https://www.gerad.ca/Sebastien.Le.Digabel/talks/2010_JOPT_25mins.pdf`.


Le Digabel, S., Lesage-Landry, A., Salomon, L., and Tribes, C. (2025).
Efficient search strategies for constrained multiobjective blackbox optimization.
Technical Report G-2025-28, Les cahiers du GERAD.


Le Digabel, S. and Wild, S. (2024).
A taxonomy of constraints in black-box simulation-based optimization.
Optimization and Engineering, 25(2):1125–1143.


Mahseredjian, J., Dennetière, S., Dubé, L., Khodabakhchian, B., and Gérin-Lajoie, L. (2007).
On a new approach for the simulation of transients in power systems.
Electric Power Systems Research, 77(11):1514–1520.


Pedersen, M., Forsting, A., van der Laan, P., Riva, R., Romàn, L. A., Criado Risco, J., Friis-Møller, M., Quick, J., Schøler Christiansen, J., Valotta Rodrigues, R., Tobias Olsen, B., and Réthoré, P.-E. (2023).
PyWake 2.5.0: An open-source wind farm simulation tool.
Software available at `https://gitlab.windenergy.dtu.dk/TOPFARM/PyWake`.

References VIII


Talgorn, B., Audet, C., Kokkolaras, M., and Le Digabel, S. (2018).
Locally weighted regression models for surrogate-assisted design optimization.
Optimization and Engineering, 19(1):213–238.


Talgorn, B., Le Digabel, S., and Kokkolaras, M. (2015).
Statistical Surrogate Formulations for Simulation-Based Design Optimization.
Journal of Mechanical Design, 137(2):021405–1–021405–18.


Vicente, L. and Custódio, A. (2012).
Analysis of direct searches for discontinuous functions.
Mathematical Programming, 133(1-2):299–325.