Blackbox optimization with the MADS algorithm and the NOMAD software

Sébastien Le Digabel

EURO 24, 2024-07-01

Presentation outline

Introduction

The MADS algorithm

The NOMAD software package

Conclusion

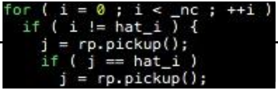
Context: Blackbox Optimization (BBO)

$$\min_{\mathbf{x}\in\mathcal{X}} \quad F(\mathbf{x}) \text{ s.t. } \mathbf{x}\in\Omega=\{\mathbf{x}\in\mathcal{X} : c_j(\mathbf{x})\le 0, j=1,2,\ldots,m\}$$

$\mathcal{X}$ is a n-dimensional space, F can have p components, and the evaluations of F and the c_j's are provided by a blackbox:

$\mathbf{x}\in\mathcal{X}$

n inputs

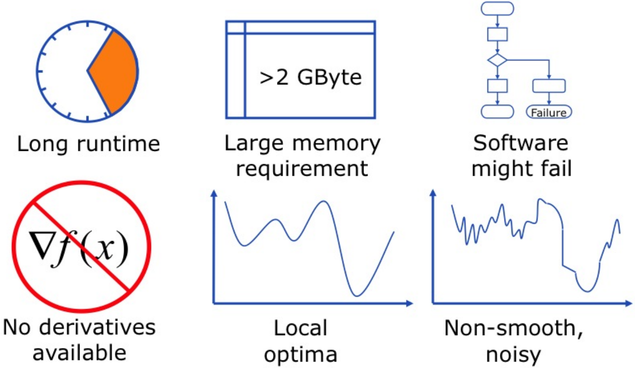


$F(\mathbf{x}), c_j(\mathbf{x}), j=1,2,\ldots,m$

$p+m$ outputs

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

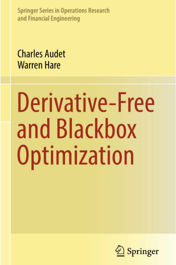
Blackboxes as illustrated by a Boeing engineer + SOLAR



The SOLAR problem [Andrés-Thió et al., 2024] is an example of a challenging blackbox application available at https://github.com/bbopt/solar

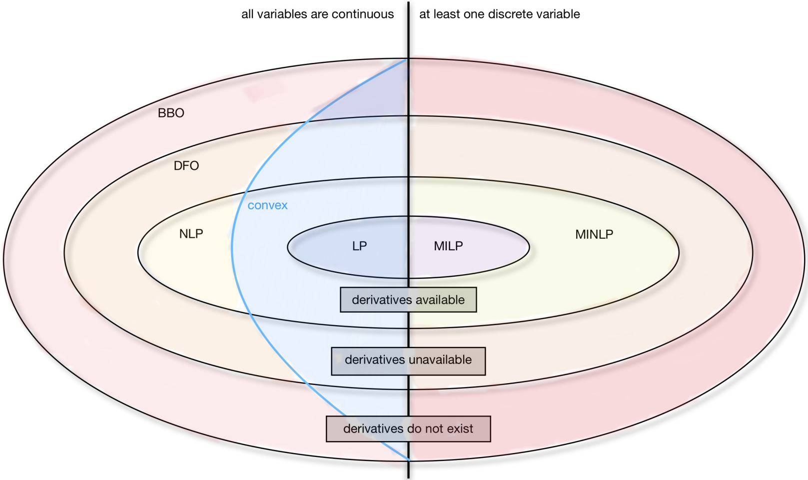
Terms

- *"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

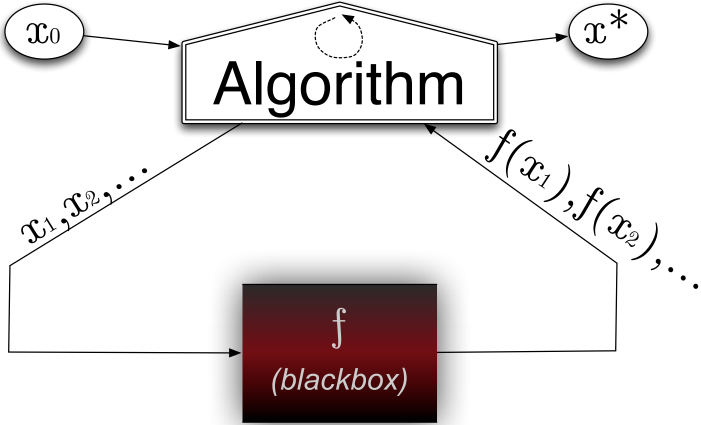


- *"Blackbox Optimization (BBO) is the study of design and analysis of algorithms that assume the objective and/or constraints functions are given by blackboxes"* [Audet and Hare, 2017]
  - A simulation, or a blackbox, is involved
  - Obj./constraints may be analytical functions of the outputs
  - Derivatives 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
- Have convergence properties ensuring first-order local optimality in the smooth case – otherwise why using it on more complicated problems?
- Easily exploit parallelism
- Deal with multiobjective optimization
- Deal with integer and categorical variables
- Have a publicly available implementation

Families of methods

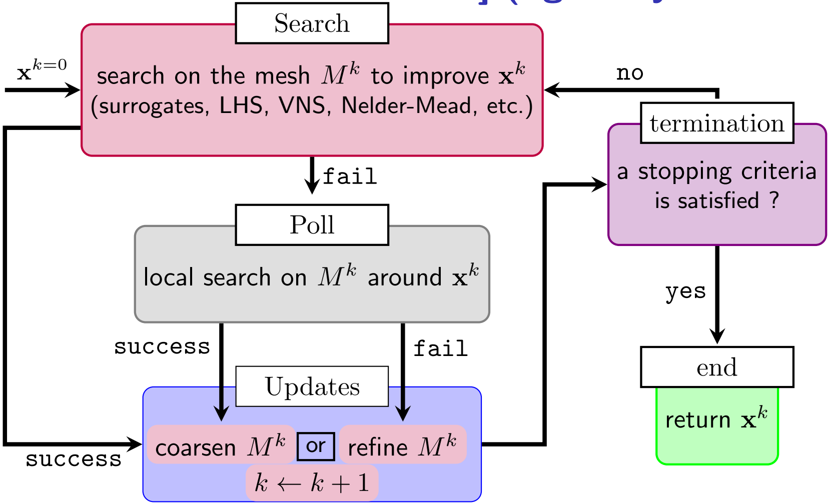
- *"Computer science"* methods:
  - Heuristics such as genetic algorithms
  - No convergence properties
  - Cost a **lot** of evaluations
  - Should be used only in last resort for desperate cases
- Statistical methods:
  - Design of experiments – out of date compared to modern DFO methods
  - Bayesian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

DFO methods

- **Model-based methods:**
  - Derivative-Free Trust-Region (DFTR) methods.
  - Based on quadratic models or radial-basis functions
  - Use of a trust-region
  - Better for $\{$ DFO $\setminus$ 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 (GPS), Generating Set Search (GSS), 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 [Audet and Dennis, Jr., 2006] (figure by Samuel Mendoza)



[0] Initializations $(\mathbf{x}^0, \delta^0)$

[1] Iteration k

[1.1] Search

select a finite number of mesh points

evaluate candidates opportunistically

[1.2] Poll (if Search failed)

construct poll set $P_k = \{\mathbf{x}^k + \delta^k d : d \in D_k\}$

sort(P_k)

evaluate candidates opportunistically

[2] Updates

if success

$\mathbf{x}^{k+1} \leftarrow$ success point

increase δ^k

else

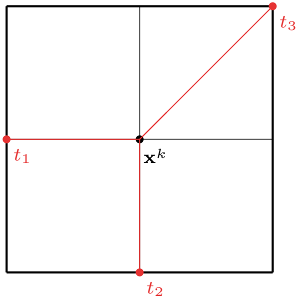
$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k$

decrease δ^k

$k \leftarrow k+1$, stop or go to **[1]**

MADS illustration with $n = 2$: Poll step

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



poll trial points$=\{t_1, t_2, t_3\}$

δ^k is the mesh size parameter

Δ^k is the frame size parameter

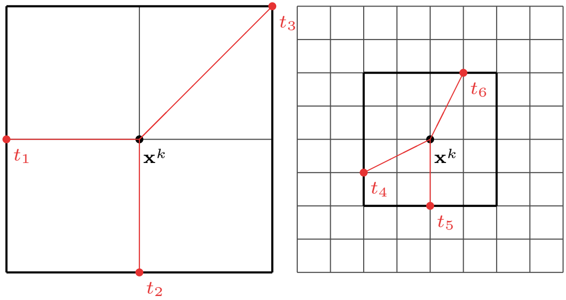
we keep $\delta^k < \Delta^k$ typically with $\Delta^k = \sqrt{\delta^k}$
and $\delta^{k+1} \leftarrow \delta^k \times 4$ (success)
or $\delta^{k+1} \leftarrow \delta^k / 4$ (fail)

MADS illustration with $n = 2$: Poll step

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

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

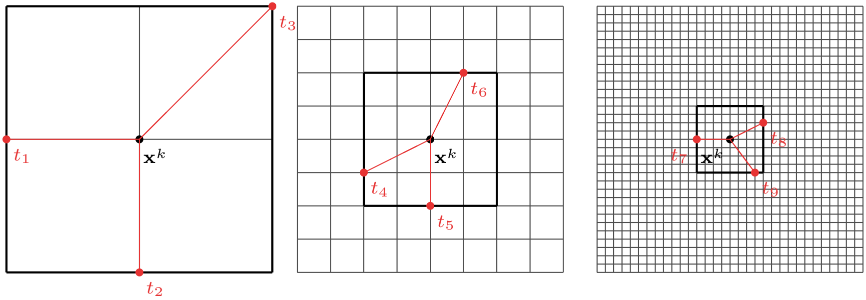
$\Delta^{k+1} = 1/2$



poll trial points=$\{t_1, t_2, t_3\}$ $= \{t_4, t_5, t_6\}$

MADS illustration with $n = 2$: Poll step

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



poll trial points$=\{t_1, t_2, t_3\}$ $= \{t_4, t_5, t_6\}$ $= \{t_7, t_8, t_9\}$

Special features of MADS

- Constraints handling with the Progressive Barrier [Audet and Dennis, Jr., 2009]
  - and hidden constraints [Audet et al., 2022a]
- Surrogates [Talgorn et al., 2015]
- Categorical variables [Abramson, 2004]
- Granular and discrete variables [Audet et al., 2019]
- Global optimization [Audet et al., 2008a]
- Parallelism [Le Digabel et al., 2010, Audet et al., 2008b]
- Multiobjective optimization [Bigeon et al., 2022]
- Sensitivity analysis [Audet et al., 2012]
- Handling of stochastic blackboxes [Alarie et al., 2021, Audet et al., 2021]

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 with MPI
- MATLAB versions; Multiple interfaces (Python, Excel, etc.)
- Open and free – LGPL license
- Download at `https://www.gerad.ca/nomad`
- Support at `nomad@gerad.ca`
- Related articles in TOMS [Le Digabel, 2011, Audet et al., 2022b]

NOMAD: **History and team**

- Developed since 2000
- Current versions: 3.9 (June 2018) and 4.4 (January 2024)
- Algorithm designers, developers:
  - M. Abramson, C. Audet, G. Couture, J. Dennis, S. Le Digabel, V. Rochon-Montplaisir, C. Tribes
- Developers:
  - Versions 1 and 2: G. Couture
  - **Version 3 (2008)**: S. Le Digabel, C. Tribes
  - **Version 4 (2021)**: V. Rochon-Montplaisir, C. Tribes

Main functionalities (1/2)

- Single ($p = 1$) or multiobjective optimization ($p > 1$)
- 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

(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` (one 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
```

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
- Features: Constraints, biobjective, global optimization, surrogates, several types of variables, parallelism
- Fast support at nomad@gerad.ca
- NOMAD has become a baseline for benchmarking DFO algorithms

References I

Abramson, M. (2004).
Mixed Variable Optimization of a Load-Bearing Thermal Insulation System Using a Filter Pattern Search Algorithm.
Optimization and Engineering, 5(2):157–177.

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.

Andrés-Thió, N., Audet, C., Diago, M., Gheribi, A., Le Digabel, S., Lebeuf, X., Lemyre Garneau, M., and Tribes, C. (2024).
`solar`: A solar thermal power plant simulator for blackbox optimization benchmarking.
Technical Report G-2024-37, Les cahiers du GERAD.

Audet, C., Batailly, A., and Kojtych, S. (2022a).
Escaping Unknown Discontinuous Regions in Blackbox Optimization.
SIAM Journal on Optimization, 32(3):1843–1870.

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

References II

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.

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

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

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

References III

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.

Bigeon, J., Le Digabel, S., and Salomon, L. (2022).
Handling of constraints in multiobjective blackbox optimization.
Technical Report G-2022-10, Les cahiers du GERAD.
To appear in *Computational Optimization and Applications*.

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.

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.