

# Derivative-free optimization (DFO) and Blackbox optimization (BBO)

an introduction taken from MTH8418

IICAP Conference Series

Sébastien Le Digabel, Polytechnique Montréal

2019-11-07

# Plan

**Problem definition**

**Algorithms for DFO and BBO**

**Example 1: Snow water equivalent estimation**

**Example 2: Aircraft takeoff trajectories**

**Example 3: Characterization of objects from radiographs**

**Example 4: Hyperparameter optimization**

**Software for DFO and BBO**

## Problem definition

Algorithms for DFO and BBO

Example 1: Snow water equivalent estimation

Example 2: Aircraft takeoff trajectories

Example 3: Characterization of objects from radiographs

Example 4: Hyperparameter optimization

Software for DFO and BBO

## General optimization problem

**Mathematical optimization** is the field that studies problems of the form

$$\min_{x \in \mathcal{X}} \{f(x) : x \in \Omega\}$$

where

- ▶  $\mathcal{X}$  is a  $n$ -dimensional space, corresponding to the **optimization variables**  $(x_1, x_2, \dots, x_n)$
- ▶  $\Omega \subseteq \mathcal{X}$  is the set of **feasible points**, defined by **constraints**. Typically

$$\Omega = \{x \in \mathcal{X} : c_j(x) \leq 0, j \in \{1, 2, \dots, m\}\}$$

- ▶ The **objective function**  $f$  takes its values on  $\mathcal{X}$

## Types of problems

$$\min_{x \in \mathcal{X}} \{f(x) : x \in \Omega\}$$

- ▶ *Combinatorial or discrete optimization (MILP, MINLP)*: At least one variable in  $\mathbb{N}$  or  $\mathbb{Z}$  or  $\{0, 1\}$
- ▶ *Continuous optimization*:  $\mathcal{X} = \mathbb{R}^n$ 
  - ▶ *Linear optimization (LP)*: Functions  $f$  and  $c_j$ 's are linear.
  - ▶ *Nonlinear optimization (NLP)*: Functions  $f$  and  $c_j$ 's are differentiable.
  - ▶ *Derivative-free optimization (DFO)*: At least one of these functions has no available derivatives.
- ▶ Several objectives, noisy functions, categorical variables, etc.

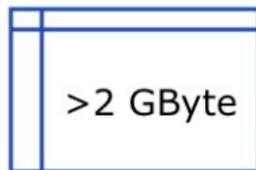
## Terms

- ▶ “*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)*.

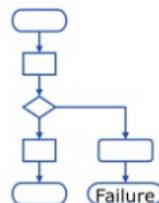
## Blackboxes as illustrated by a Boeing engineer



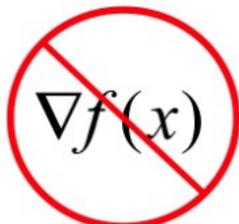
Long runtime



Large memory requirement



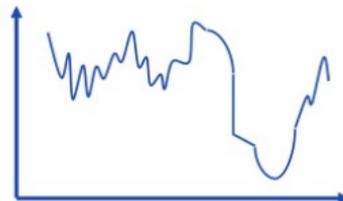
Software might fail



No derivatives available

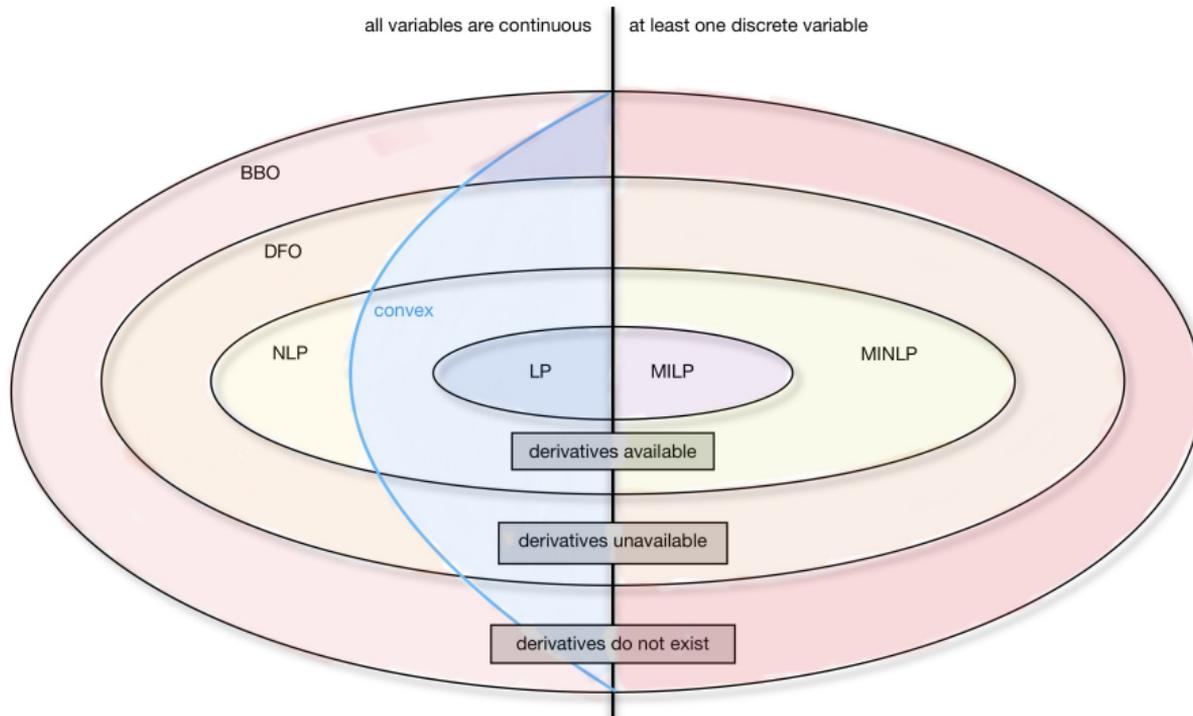


Local optima



Non-smooth, noisy

# Optimization: Global view



## Problem definition

## Algorithms for DFO and BBO

Example 1: Snow water equivalent estimation

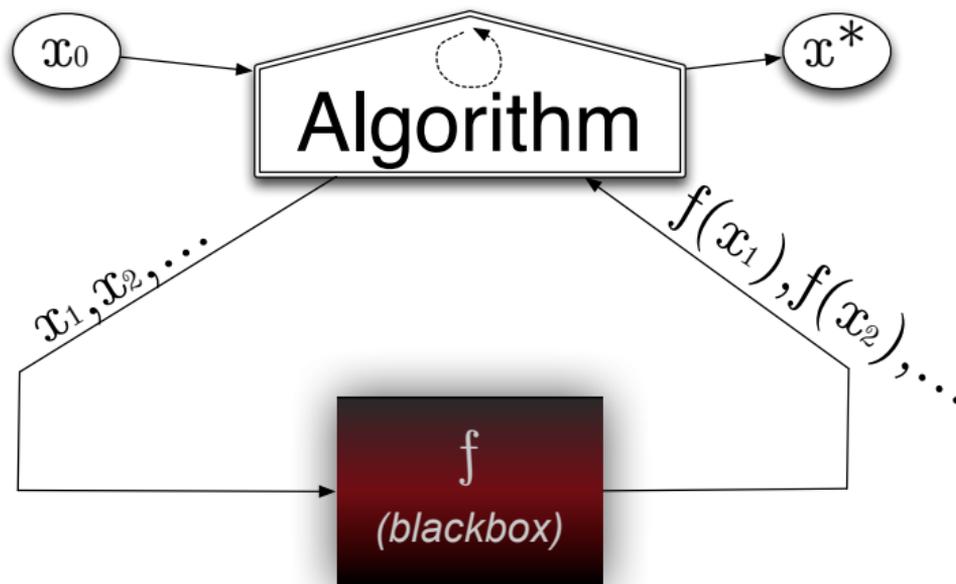
Example 2: Aircraft takeoff trajectories

Example 3: Characterization of objects from radiographs

Example 4: Hyperparameter optimization

Software for DFO and BBO

## 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 conjunction 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 \ BBO }.
- ▶ Not resilient to noise and *hidden constraints*.
- ▶ Not easy to parallelize.

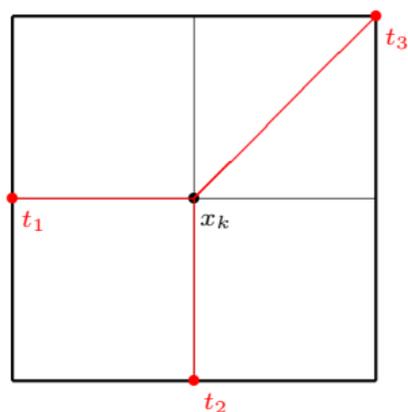
### ▶ Direct-search methods:

- ▶ 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 illustration with $n = 2$ : Poll step

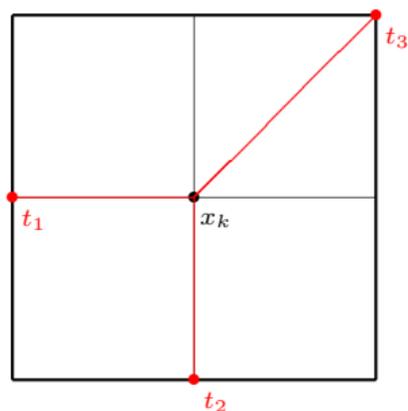
$$\Delta_k^m = \Delta_k^p = 1$$



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

## MADS illustration with $n = 2$ : Poll step

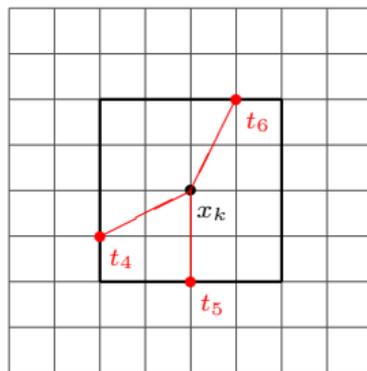
$$\Delta_k^m = \Delta_k^p = 1$$



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

$$\Delta_{k+1}^m = 1/4$$

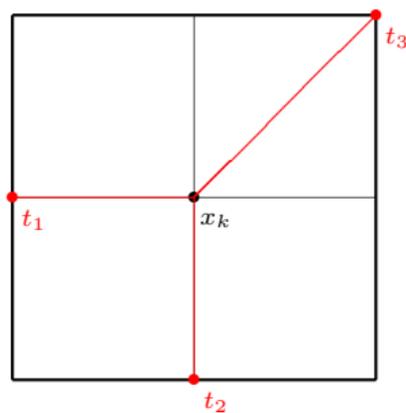
$$\Delta_{k+1}^p = 1/2$$



=  $\{t_4, t_5, t_6\}$

# MADS illustration with $n = 2$ : Poll step

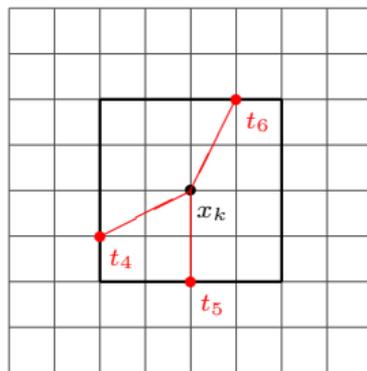
$$\Delta_k^m = \Delta_k^p = 1$$



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

$$\Delta_{k+1}^m = 1/4$$

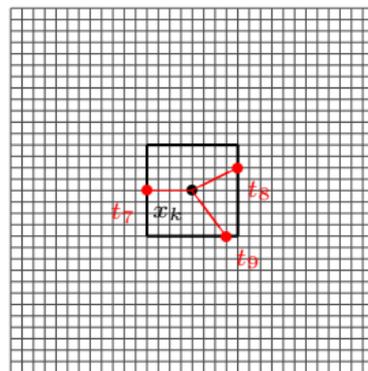
$$\Delta_{k+1}^p = 1/2$$



=  $\{t_4, t_5, t_6\}$

$$\Delta_{k+2}^m = 1/16$$

$$\Delta_{k+2}^p = 1/4$$



=  $\{t_7, t_8, t_9\}$

Problem definition

Algorithms for DFO and BBO

**Example 1: Snow water equivalent estimation**

Example 2: Aircraft takeoff trajectories

Example 3: Characterization of objects from radiographs

Example 4: Hyperparameter optimization

Software for DFO and BBO

## Snow water equivalent (SWE) estimation

- ▶ [Alarie et al., 2013]
- ▶ **Accurate estimate of water** stored in snow is crucial to optimize hydroelectric plants management.
- ▶ Exact snow measurement is impossible.
- ▶ SWE is **measured at specific sites** and next **interpolated over the territory**.
- ▶ **Territory is huge**: Hydro-Québec (HQ) operates 565 dams, 75 reservoirs, and 56 hydroelectric power plants, located over 90 watersheds and covering more than 550,000 km<sup>2</sup>



## Previous SWE estimation

- ▶ Done manually by weighing snow cores at specific sites.
- ▶ Each measurement campaign requires 2 weeks.
- ▶ Missing measurements due to adverse meteorological conditions.



## GMON device

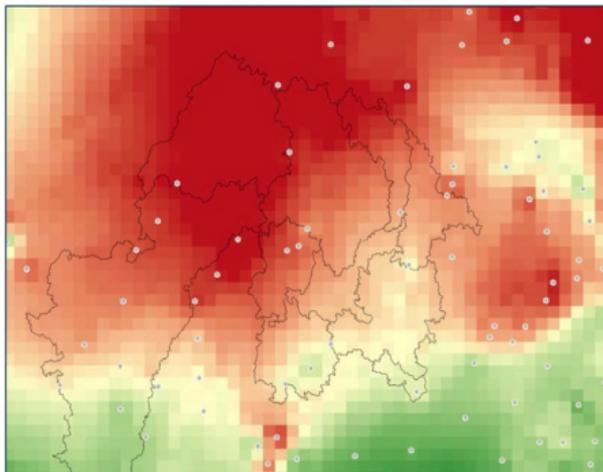
- ▶ A new measuring instrument that provides daily automatic SWE.
- ▶ **GMON** for Gamma-MONitoring device: it measures the natural Gamma radiation emitted from the soil.
- ▶ Communicates via satellites.
- ▶ **GMON stations described at Radio-Canada.**



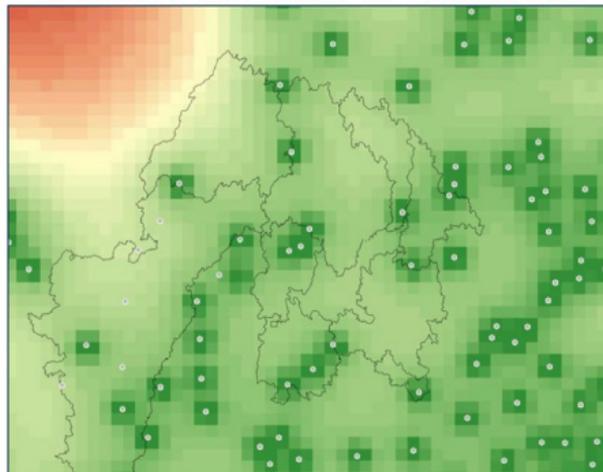
## SWE estimation from GMON measures

- ▶ Kriging interpolation is used to obtain SWE estimation together with an error map.
- ▶ How to find the device locations that minimize the kriging interpolation error of the SWE?

SWE estimation



standard deviation of estimation

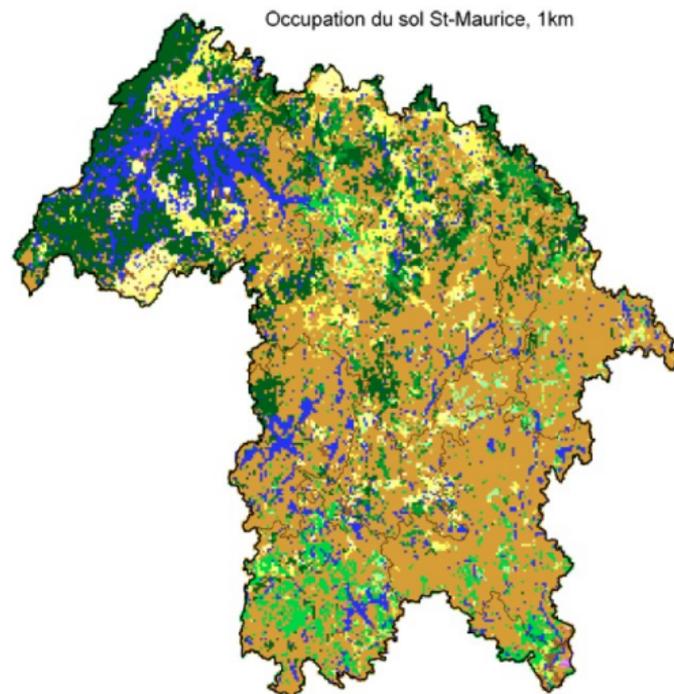


## Problem formulation

- ▶  $x \in \mathbb{R}^{2N}$  are the locations of  $N$  stations.
- ▶ Typically,  $N \leq 10$ , so we do not consider it as a variable.
- ▶  $\Omega \subseteq \mathbb{R}^2$  is the feasible domain where the stations can be located.
- ▶  $f(x)$  is a score based on the standard deviation map obtained by the kriging simulation and is considered as a blackbox.
- ▶ Each simulation requires  $\simeq 2$  seconds, and can only be launched within the Hydro-Québec research center (IREQ).

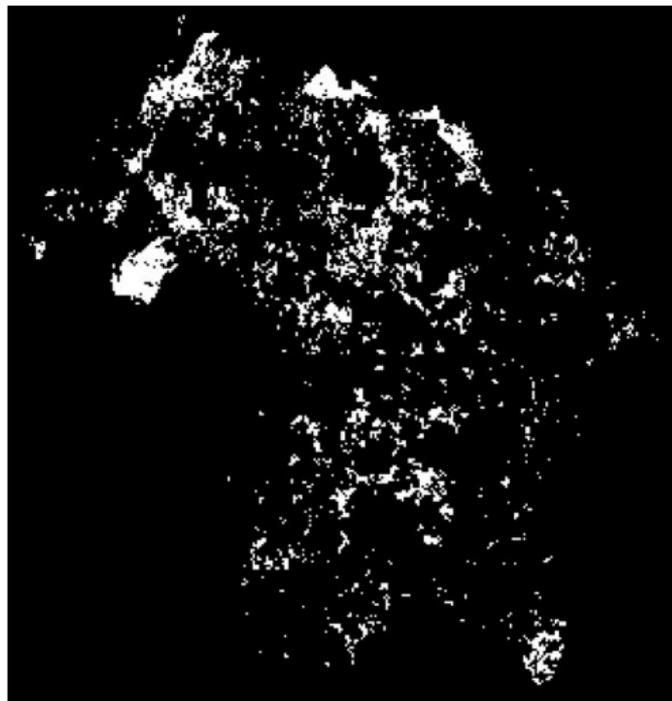
## Constraints

- ▶ GMON stations cannot be located anywhere.
- ▶ Restrictions on:
  - ▶ subsoil properties,
  - ▶ slope,
  - ▶ vegetation,
  - ▶ exploitation,
  - ▶ etc.



## Constraints

- ▶ GMON stations cannot be located anywhere.
- ▶ Restrictions on:
  - ▶ subsoil properties,
  - ▶ slope,
  - ▶ vegetation,
  - ▶ exploitation,
  - ▶ etc.
- ▶ Highly fragmented domain.



## Problem definition

## Algorithms for DFO and BBO

### Example 1: Snow water equivalent estimation

### **Example 2: Aircraft takeoff trajectories**

### Example 3: Characterization of objects from radiographs

### Example 4: Hyperparameter optimization

## Software for DFO and BBO

## Aircraft takeoff trajectories

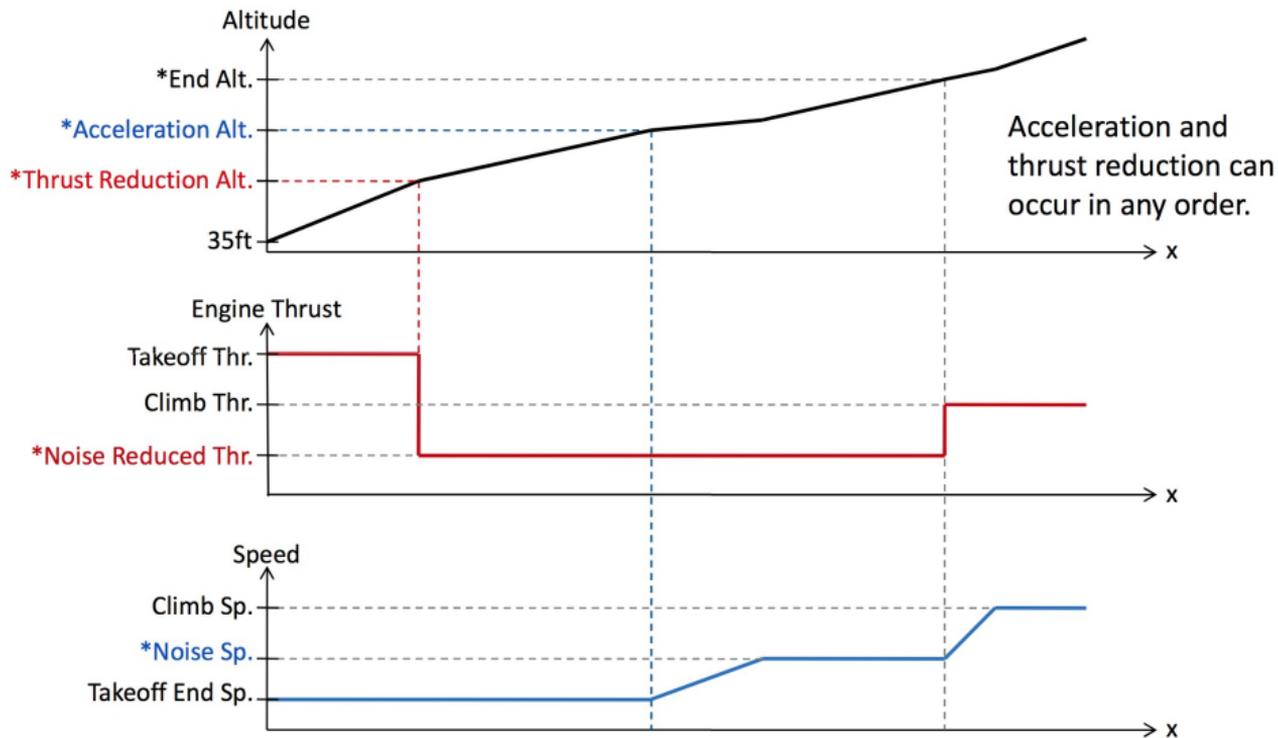


- ▶ [Torres et al., 2011]
- ▶ **AIRBUS** problem involving (among others): O. Babando, C. Bes, J. Chaptal, J.-B. Hiriart-Urruty, B. Talgorn, B. Tessier, and R. Torres.
- ▶ Biobjective optimization.
- ▶ Must execute on different platforms including some old Solaris distributions.

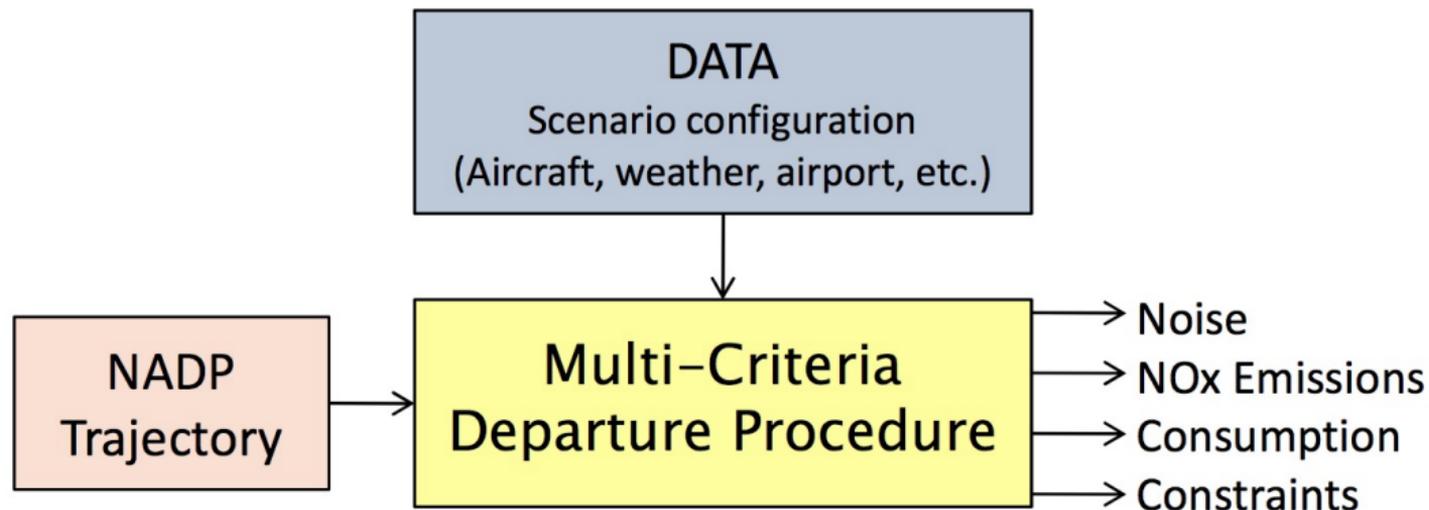
## Definition of the optimization problem

- ▶ Concept : Optimization of vertical flight path based on procedures designed to reduce noise emission at departure to protect airport vicinity.
- ▶ Minimization of environmental and economical impact: **Noise** and **fuel consumption**.
- ▶ **NADP (Noise Abatement Departure Procedure), variables**: During departure phase, the aircraft will target its climb configuration:
  - ▶ Increase the speed up to climb speed (acceleration phase).
  - ▶ Reduce the engine rate to climb thrust (reduction phase).
  - ▶ Gain altitude.

# Parametric Trajectory: 5 optimization variables (\*)



## The blackbox: MCDP: Multi-Criteria Departure Procedure



One evaluation  $\simeq$  2 seconds.

## Problem definition

## Algorithms for DFO and BBO

### Example 1: Snow water equivalent estimation

### Example 2: Aircraft takeoff trajectories

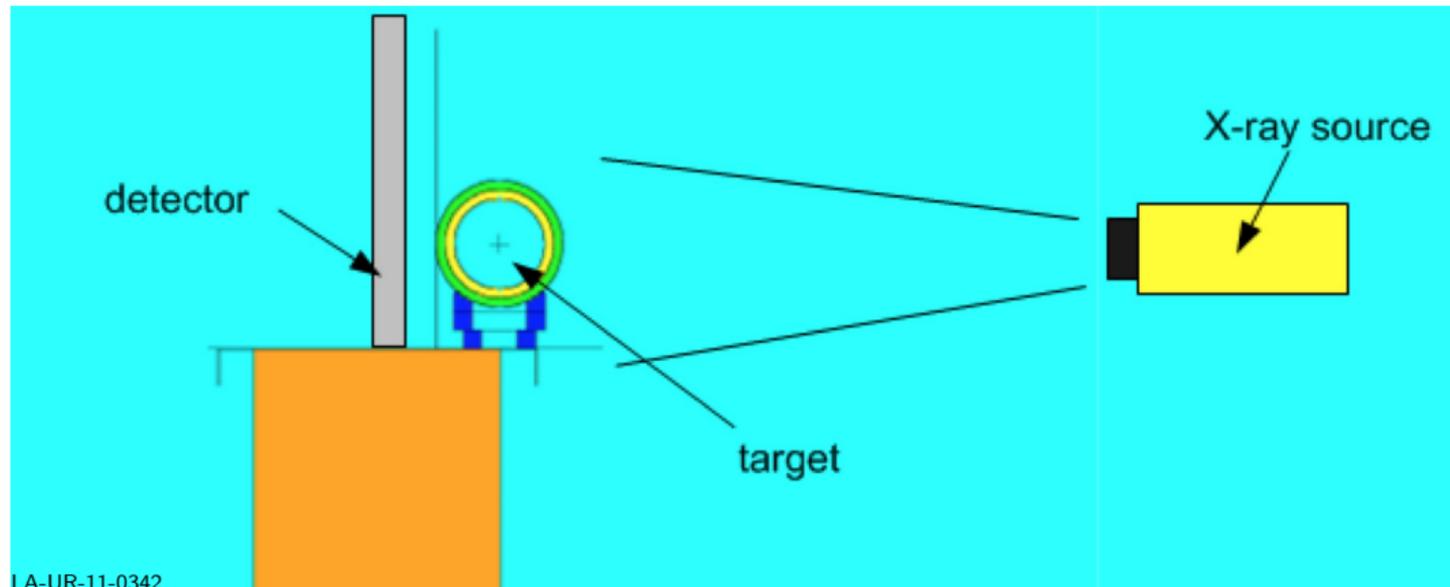
### **Example 3: Characterization of objects from radiographs**

### Example 4: Hyperparameter optimization

## Software for DFO and BBO

## Characterization of objects from radiographs - LANL

We want to identify an unknown **object** inside a box, using a **x-ray source** that gives an image on a **detector**.

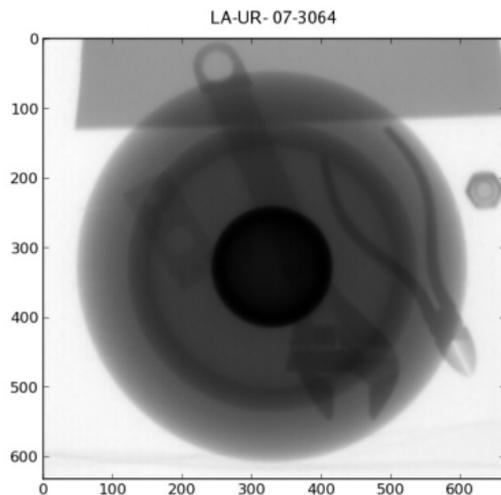


LA-UR-11-0342

In this work, the unknown object is supposed to be **spherical**.

# Radiograph

A **radiograph** is the observed image on the detector. For example:



## Description of the problem

- ▶ The problem consist to **identify the unknown object** with sufficient precision so that the object can be classified as dangerous or not.
- ▶ Must work **rapidly**.
- ▶ Must work for radiographs **not created on a well-controlled experimental environment**.
- ▶ Must **not crash** for unreasonable user inputs.

## Definition of the optimization problem

### ▶ Variables:

- ▶ They represent a **spherical object**.
- ▶ **Categorical variables**: Number of layers and type of material of each layer.
- ▶ Continuous variables: Radius of each layer.
- ▶ The **number of variables can change** depending on the number of layers.

### ▶ Objective function:

- ▶ A score associated to the difference between the observed image on the detector, and a simulated image obtained from the candidate object (**inverse problem**).
- ▶ A numerical code – **the blackbox** – produces this simulated radiograph, using raytracing.
- ▶ Quick to compute.

Problem definition

Algorithms for DFO and BBO

Example 1: Snow water equivalent estimation

Example 2: Aircraft takeoff trajectories

Example 3: Characterization of objects from radiographs

**Example 4: Hyperparameter optimization**

Software for DFO and BBO

## Hyperparameter optimization (HPO)

- ▶ PhD project of Dounia Lakhmiri.
- ▶ We focus on the HPO of deep neural networks.
- ▶ Our advantages:
  - ▶ Blackbox optimization problem:  
*One blackbox call = One training + one validation for a fixed set of hyperparameters.*
  - ▶ Presence of categorical variables (*ex.: number of layers*).
  - ▶ Existing methods are mostly heuristics  
*(grid search, random search, GAs, GPs, etc.)*

- ▶ Average results on MNIST:
  - ▶ Random search: 94.0%
  - ▶ RBFOpt [Diaz et al., 2017]: 95.7%.
  - ▶ **HyperNOMAD**: 95.4% (w/o categorical), **97.5%** (categorical).

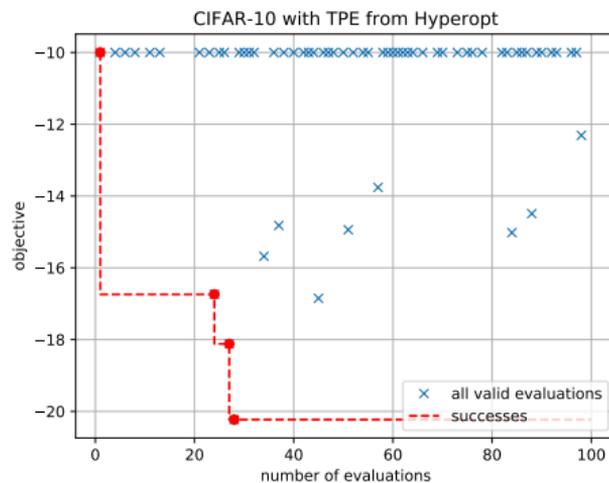
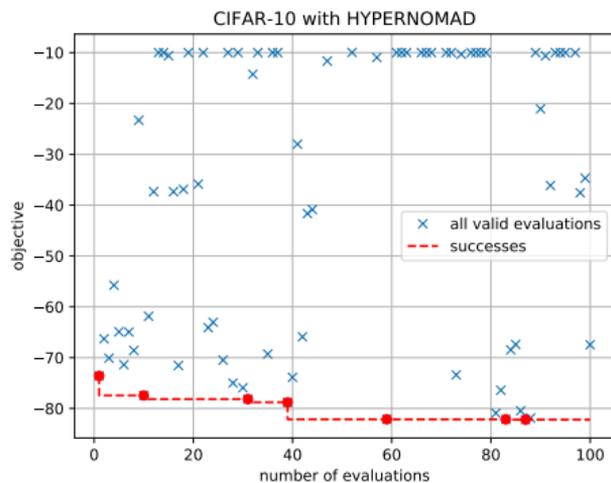


## CIFAR-10 (1/2)

- ▶  $5 \times n1 + n2 + 10$  variables:
  - ▶ 2 categorical variables:  $n1$  (number of convolution layers) and  $n2$  (number of fully connected layers).
  - ▶ Type of optimizer, 4 HPs related to the optimizer (example: learning rate, momentum, weight decay, dampening), Dropout rate, activation function, batch size, number of epochs.
  - ▶ For each layer: `n_output`, `kernel_size`, `stride`, `padding`, `pooling` and `output_size`.
  
- ▶ Training with 50,000 images, validation on 10,000 images.
  
- ▶ One evaluation (training+test)  $\simeq$  2 hours  
(CPU: i7-6700 @ 3.4 GHz, GPU: Nvidia GeForce GTX 1070)
  
- ▶ Current best solution: 96.58%



# CIFAR-10 (2/2): Comparison with Hyperopt



## Problem definition

## Algorithms for DFO and BBO

### Example 1: Snow water equivalent estimation

### Example 2: Aircraft takeoff trajectories

### Example 3: Characterization of objects from radiographs

### Example 4: Hyperparameter optimization

## Software for DFO and BBO

## List of solvers

- ▶ We list the solvers that can be applied to blackbox optimization.
- ▶ Most of them:
  - ▶ Are available for free on the internet.
  - ▶ Are possibly inside an optimization toolbox.
  - ▶ Seem to be actively maintained [Feb. 2019].
- ▶ Websites listing solvers:
  - ▶ [Wikipedia](#).
  - ▶ [Decision Tree for Optimization Software](#).
  - ▶ The [DFO course homepage](#).

## 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](mailto:nomad@gerad.ca)
  
- ▶ Related article in TOMS [Le Digabel, 2011] (WoS Highly Cited Paper).



## Blackbox conception

- ▶ Command-line program that takes in argument a file containing  $x$ , and displays the values of  $f(x)$  and the  $c_j(x)$ '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 : '$NOMAD_HOME/src/lqpl.txt'
User guide: '$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
```

# References I



Alarie, S., Audet, C., Garnier, V., Le Digabel, S., and Leclaire, L.-A. (2013).  
Snow water equivalent estimation using blackbox optimization.  
*Pacific Journal of Optimization*, 9(1):1–21.



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



Conn, A., Scheinberg, K., and Vicente, L. (2009).  
*Introduction to Derivative-Free Optimization*.  
MOS-SIAM Series on Optimization. SIAM, Philadelphia.



Diaz, G., Fokoue, A., Nannicini, G., and Samulowitz, H. (2017).  
An effective algorithm for hyperparameter optimization of neural networks.  
*IBM Journal of Research and Development*, 61(4):9:1–9:11.



Fermi, E. and Metropolis, N. (1952).  
Numerical solution of a minimum problem.  
Los Alamos Unclassified Report LA-1492, Los Alamos National Laboratory, Los Alamos, USA.



Jones, D., Schonlau, M., and Welch, W. (1998).  
Efficient Global Optimization of Expensive Black Box Functions.  
*Journal of Global Optimization*, 13(4):455–492.

## References II



Kolda, T., Lewis, R., and Torczon, V. (2003).

Optimization by direct search: New perspectives on some classical and modern methods.  
*SIAM Review*, 45(3):385–482.



Lakhmiri, D. (2019).

HyperNOMAD.

<https://github.com/DouniaLakhmiri/HyperNOMAD>.



Lakhmiri, D., Digabel, S. L., and Tribes, C. (2019).

HyperNOMAD: Hyperparameter optimization of deep neural networks using mesh adaptive direct search.  
Technical Report G-2019-46, Les cahiers du GERAD.



Le Digabel, S. (2011).

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



Nelder, J. and Mead, R. (1965).

A simplex method for function minimization.  
*The Computer Journal*, 7(4):308–313.



Torczon, V. (1997).

On the convergence of pattern search algorithms.  
*SIAM Journal on Optimization*, 7(1):1–25.

## References III



Torres, R., Bès, C., Chaptal, J., and Hiriart-Urruty, J.-B. (2011).  
Optimal, Environmentally-Friendly Departure Procedures for Civil Aircraft.  
*Journal of Aircraft*, 48(1):11–22.