

ISMP 2009

Ecole Polytechnique de Montréal

Black-Box Optimization with the **NOMAD** Software

Sébastien Le Digabel

Charles Audet

John Dennis

Quentin Reynaud

2009-08-25

Presentation outline

Context and MADS overview

NOMAD

Discussion

Presentation outline

Context and MADS overview

NOMAD

Discussion

Presentation outline

Context and MADS overview

NOMAD

Discussion

Context and MADS overview

NOMAD

Discussion

Black-box optimization problems

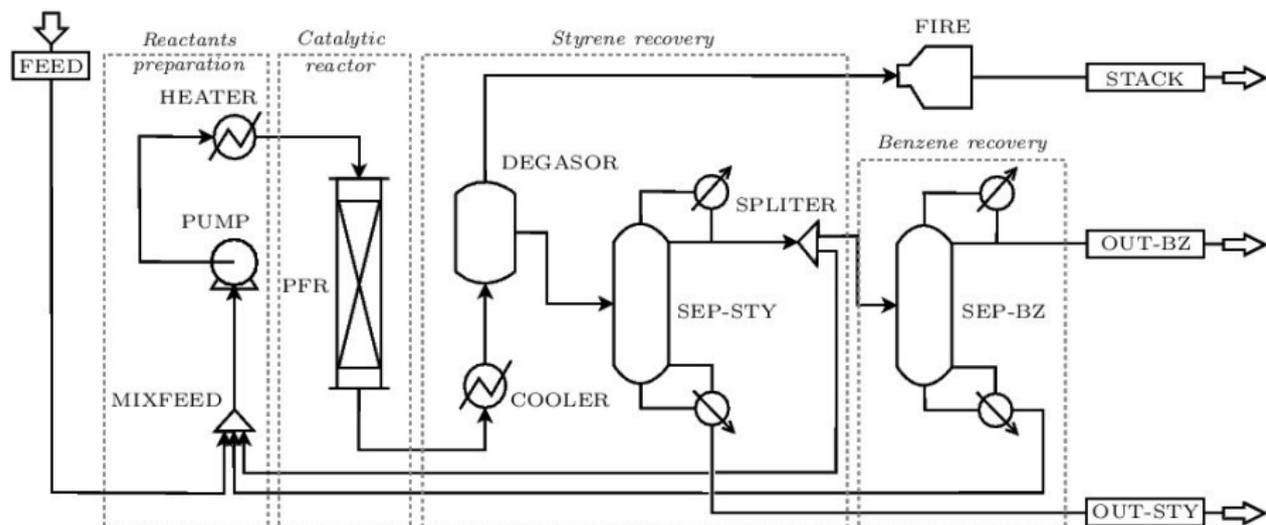
We consider the nonsmooth optimization problem:

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{subject to} & x \in \Omega, \end{array}$$

where evaluation of the functions are usually the result of a computer code (a black-box) costly to evaluate, without derivatives.

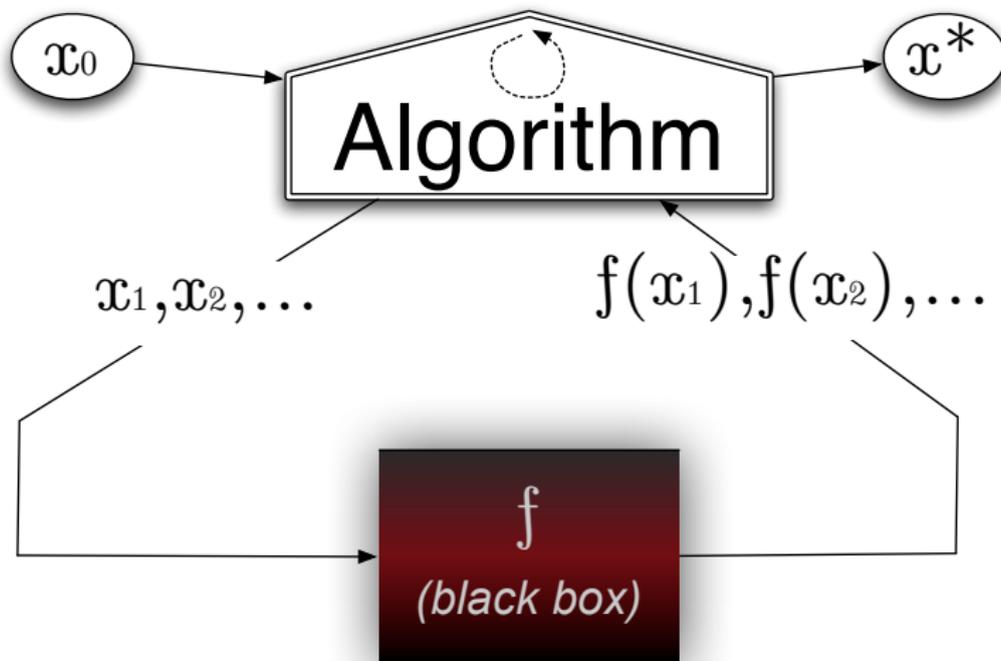
An engineering problem

Styrene production [Audet, Béchard, Le Digabel, JOGO 2008]



8 variables, 11 constraints. One evaluation \simeq 1s.

MADS and direct search principle



Mesh Adaptive Direct Search (MADS)

- ▶ Iterative algorithm that evaluates the black-box functions at some trial points.
- ▶ Trial points are generated on a mesh.
- ▶ **Search step:** trial points can be generated anywhere on the mesh.
- ▶ **Poll step:** directions are used to generate poll trial points. The normalized directions are dense in the unit sphere.
- ▶ **Updates step:** the mesh size is reduced if no new iterate is found.
- ▶ Algorithm backed by a solid convergence analysis.

Context and MADS overview

NOMAD

Discussion

NOMAD

- ▶ C++ implementation of MADS.
- ▶ Standard C++, no other package needed.
- ▶ Runs on Linux, Unix, Mac OS X and Windows.
- ▶ Distributed under the GPL license.
- ▶ Command-line interface.

NOMAD – history and team

- ▶ Developed since 2000.
- ▶ Algorithm designers:
 - ▶ M. Abramson, C. Audet, J. Dennis, and myself.
- ▶ Developers:
 - ▶ Versions 1 and 2: G. Couture.
 - ▶ Version 3 (2008): M. Sylla and Q. Reynaud, and myself.
- ▶ Current version: 3.2. Version 3.3 in development.

Main functionalities (1/2)

- ▶ Single or Biobjective optimization.
- ▶ Variables:
 - ▶ Continuous, integer, binary, categorical (v.3.3),
 - ▶ Periodic, fixed,
 - ▶ Groups of variables.
- ▶ Searches:
 - ▶ Latin-Hypercube (LH),
 - ▶ Variable Neighborhood (VNS),
 - ▶ User search.

Main functionalities (2/2)

- ▶ Constraints treated with 4 different methods:
 - ▶ Extreme Barrier,
 - ▶ Progressive Barrier (default),
 - ▶ Progressive-to-Extreme Barrier,
 - ▶ Filter method.
- ▶ Several direction types:
 - ▶ Coordinate directions,
 - ▶ LT-MADS,
 - ▶ OrthoMADS,
 - ▶ Hybrid combinations.
- ▶ Static surrogate functions: used to order the poll trial points, in VNS search, and in the extended poll for categorical variables (v.3.3).

(all items correspond to published or submitted papers).

NOMAD installation

- ▶ Pre-compiled executables are available for Windows and Mac.
- ▶ Installation programs copy these executables.
- ▶ On Unix/Linux, after download, launch the `configure` command that creates a makefile for compilation.
- ▶ Two ways to use NOMAD: batch mode or library mode.

Black-box conception (batch mode)

- ▶ 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 g1 g2`.

- ▶ Example on the styrene problem:

```
> ./black_box/truth/truth.exe points/x0.txt .
```

Important parameters

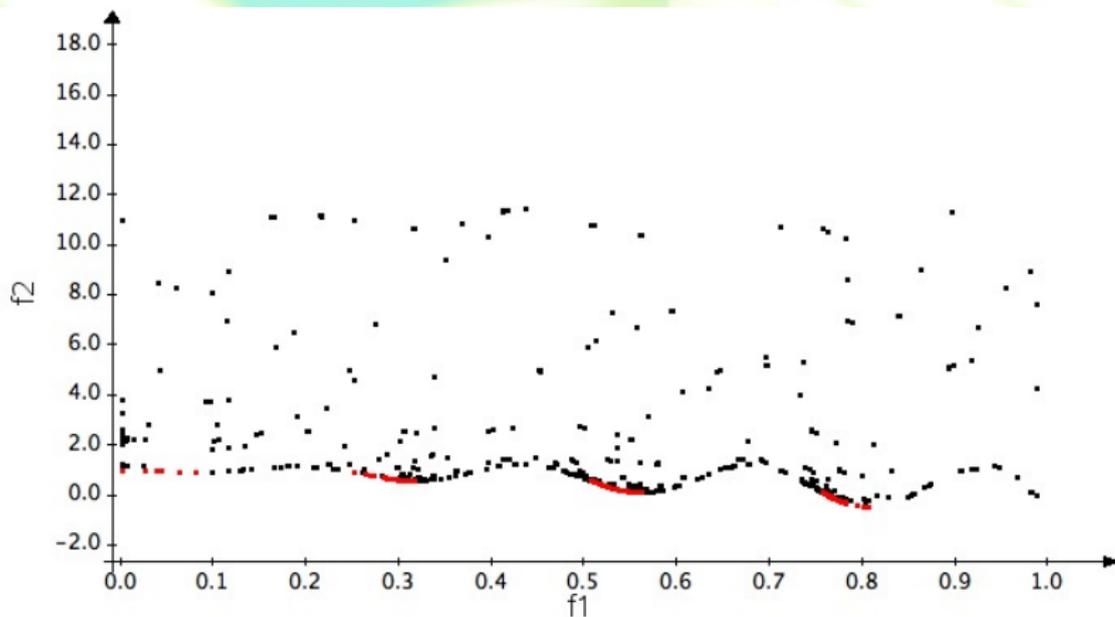
- ▶ Necessary parameters: dimension (n), the black-box characteristics, and the starting point (x_0).
- ▶ All algorithmic parameters have default values. The most important are:
 - ▶ Maximum number of black-box evaluations,
 - ▶ Starting point (more than one can be defined),
 - ▶ Types of directions (more than one can be defined),
 - ▶ Initial mesh size,
 - ▶ Constraints types,
 - ▶ Latin-Hypercube sampling,
 - ▶ Seeds.
- ▶ See the user guide for the description of all parameters, or use the `nomad -h` option.

Run NOMAD

- ▶ `> nomad parameters.txt`
- ▶ Example on the styrene problem.

Biobjective optimization

- ▶ BiMADS algorithm: series of MADS runs.
- ▶ Example with a disconnected Pareto front.



Advanced functionalities (library mode)

- ▶ No system calls: the code executes faster (more than twice).
- ▶ Easy to program multiple runs in parallel with different seeds.
- ▶ The user can program its own search strategy.
- ▶ The user can pre-process all evaluation points before they are evaluated.
- ▶ The user can decide the priority with which trial points are evaluated.
- ▶ The user can indicate user-functions that will be called at some events (new success, new iteration, new MADS run in bi-objective optimization) [V.3.3]

Examples included in the NOMAD package

These examples illustrate another possibilities:

- ▶ Multi-start from points generated with LH sampling.
- ▶ Compatibility with previous version of NOMAD.
- ▶ Problems used in library mode and coded as:
 - ▶ a Windows DLL,
 - ▶ a GAMS program,
 - ▶ a CUTEr problem,
 - ▶ a Matlab function,
 - ▶ a FORTRAN code [V.3.3].
- ▶ A GUI prototype in JAVA [V.3.3].

GUI prototype (example included with V.3.3)

- ▶ Illustrates how easy it is to define user-functions.
- ▶ Developed in JAVA.
- ▶ First prototype for a future GUI in JAVA or QT.

Context and MADS overview

NOMAD

Discussion

Future functionalities

- ▶ Version 3.3 (October): Categorical variables.
- ▶ Versions 3.4 and above:
 - ▶ PMADS and PSD-MADS (parallel versions),
 - ▶ Compatibility with AMPL,
 - ▶ Simplex gradients,
 - ▶ Dynamic surrogates,
 - ▶ GUI.

References

- ▶ MADS references and a technical paper describing the code are available from the NOMAD website.
- ▶ The NOMAD user guide is included in the package.
- ▶ Recent NOMAD tests: Globalizations strategies (COAP) and OrthoMADS (SIOPT).