Software

MTH8418

S. Le Digabel, Polytechnique Montréal

Winter 2020

(v2)
Introduction

List of solvers

Problems

Example 1: PSwarm

Example 2: NOMAD

References
Some ingredients for designing DFO software

- Evaluator
- Cache
- Parameters
- Algorithm
- Mesh
- Random generator
- Directions
- Trust-region
- Filter
- Models or surrogates
- Parallelism
- Software Name
Benchmarks in the literature

- [Fowler et al., 2008]
  - The GROUNDWATER problem, or *community* problem

- [Moré and Wild, 2009]
  - 22 CUTEst problems, 212 instances (smooth, nonsmooth, noisy)
  - Analytic expressions, no constraints
  - Website

- [Rios and Sahinidis, 2013]
  - Analytic expressions, no constraints
  - Website

- [Martelli and Amaldi, 2014]
  - Analytic functions
  - Two engineering applications (including STYRENE)

- The **BBComp competition**
Benchmarks

- Rarely a clear winner
- Results are always biased by the methodology and the choice of problems
- Constraints are barely considered
- Engineering applications are almost never considered
Plan

Introduction

List of solvers

Problems

Example 1: PSwarm

Example 2: NOMAD

References
Introduction

List of solvers

Problems

Example 1: PSwarm

Example 2: NOMAD

References
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. 2020]

- Websites listing solvers:
  - Wikipedia
  - Decision Tree for Optimization Software
  - The DFO course homepage
HOPSPACK

- Hybrid Optimization Parallel Search PACKage [Plantenga, 2009]
- Evolution of APPSPACK (Asynchronous Parallel Pattern Search) [Gray and Kolda, 2006]
- Tamara Kolda and Todd Plantenga from the Sandia National Laboratories
- Generating Set Search (GSS) [Kolda et al., 2003]
- Two parallel modes: Multiple (user-provided) solvers executed on different processes, or evaluations in parallel
- General and linear constraints
SID-PSM

- **SImplex Derivatives - Pattern Search Method**
- Developed by Ana Custódio and Luis Vicente [Custódio and Vicente, 2007]
- Generalized Pattern Search (GPS) with simplex derivatives used for ordering the points of the Poll step and quadratic model used in the Search step
- Simplex derivatives \( \simeq \) gradient of the quadratic model
- Minimum-Frobenius Norm (MFN) quadratic models in the under-determined case
- General constraints, but derivatives of the constraints must be provided
- Implemented in MATLAB
Codes by Mike Powell

- Mike Powell, University of Cambridge
- Derivative-Free Trust-Region (DFTR) model-based solvers
- FORTRAN codes available at Z. Zhang's website
- COBYLA (1992): Constrained Optimization BY Linear Approximation
- UOBYQA [Powell, 2002]: Unconstrained Optimization BY Quadratic Approximation
- NEWUOA [Powell, 2006]: Developed from UOBYQA
- BOBYQA [Powell, 2009]: Bound Optimization BY Quadratic Approximation Extension to bounds.
- LINCOA [Powell, 2014]: LINearly Constrained Optimization Algorithm. Extension to linear constraints
CMA-ES

- Covariance Matrix Adaptation Evolution Strategy
- [Hansen, 2006]
- Multiple implementations: C, C++, FORTRAN, JAVA, PYTHON, MATLAB, OCTAVE, R, SCILAB
- Bounds indirectly handled. No constraints
- Population-based stochastic method
- Global convergence with probability one
- Contrary to other population-based methods, there are few parameters to decide:
  - Starting point
  - Initial step size
  - Population size
NSGA-II

- Non-dominated Sorting Genetic Algorithm-II
- [Deb et al., 2002]
- Several implementations
- Biobjective optimization
- General constraints are handled
- Continuous and binary variables
- Stochastic population-based method
- Heuristic (no convergence theory)
Toolboxes

- **MATLAB:**
  - The Global Optimization Toolbox
  - The `fminsearch` function (Nelder-Mead)
  - The `fmincon` function: nonlinear methods (interior point, SQP, active set) using finite differences

- **DFL:** The Derivative-Free Library. FORTRAN and C. Di Pillo, Fasano, Liuzzi, Lucidi, Piccialli, Rinaldi, Sciandrone

- The **Opti Toolbox** for MATLAB and the **OpenSolver** for Excel

- **DAKOTA**

- **COINOR**

- **NLopt**
Other solvers (1/2)

- **DFO:**
  - DFTR in FORTRAN
  - Andy Conn, Katya Scheinberg, Philippe Toint

- **Tim Kelley codes:**
  - **IFFCO:** Implicit Filtering for Constrained Optimization (bounds)
  - **IMFIL:** IMplicit FILtering
  - **DIRECT:** DIviding RECTangles

- **ORBIT:**
  - DFTR with RBFs, MATLAB
  - Stefan Wild, Rommel Regis
Other solvers (2/2)

- **SNOBFIT:**
  - SQP and quadratic models, diversification with branching
  - MATLAB
  - Arnold Neumaier

- **CONDOR:**
  - Implementation of UOBYQA in MATLAB
  - Vanden Berghen and Bersini, 2005

- **Wedge:**
  - DFTR in MATLAB
  - Marazzi and Nocedal, 2002

- **NMSMAX:**
  - Nelder-Mead, MATLAB
  - Nick Higham, The matrix computation toolbox
Commercial software

▶ iSight:
  ▶ Pointer: Automatic selection of one of these methods: Evolutionary, Nelder-Mead, SQP, Tabu

▶ TOMLAB. Three packages:
  ▶ CGO
  ▶ LGO
  ▶ GENO

▶ HEEDS:
  ▶ Sherpa: Same principle as Pointer with Simulated Annealing, Genetic Algorithm, SQP, Response Surface

▶ Artelys Knitro
Introduction

List of solvers

Problems

Example 1: PSwarm

Example 2: NOMAD

References
GRIEWANK problem

- Andreas Griewank [Griewank, 1981]

- $n$ is generic, $m = 0$. We choose $n = 2$

- The problem:

$$\min_{(x,y) \in [-600;600]^2} f(x, y) = 1 + \frac{x^2 + y^2}{4000} - \cos(x) \cos\left(\frac{y}{\sqrt{2}}\right)$$

- No constraints other than bounds

- Many local optima

- Solution: $(x^*, y^*) = (0, 0), f(x^*, y^*) = 0$

- Budget of 2,000 evaluations
GRIEWANK problem

Set of 10 starting points given by LHS (9 points) and the standard starting point $x_0 = (1, 1)$

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$f(x, y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.589738091176242</td>
</tr>
<tr>
<td>33.33337681</td>
<td>67.39743216</td>
<td>2.120807858057156</td>
</tr>
<tr>
<td>93.08669623</td>
<td>46.99250073</td>
<td>3.813815924252485</td>
</tr>
<tr>
<td>25.18204021</td>
<td>23.43866215</td>
<td>1.943216300484015</td>
</tr>
<tr>
<td>44.70580342</td>
<td>3.771470648</td>
<td>2.169805625459778</td>
</tr>
<tr>
<td>3.77386892</td>
<td>16.30384942</td>
<td>1.479898240729798</td>
</tr>
<tr>
<td>79.90834487</td>
<td>36.21897984</td>
<td>3.102669439053978</td>
</tr>
<tr>
<td>71.28314081</td>
<td>89.08095617</td>
<td>4.809596569243840</td>
</tr>
<tr>
<td>11.40812019</td>
<td>80.72055663</td>
<td>2.315427422598158</td>
</tr>
<tr>
<td>59.28416324</td>
<td>55.59832326</td>
<td>2.610998296447039</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
GRIEWANK function: Plot
STYRENE problem

- [Audet et al., 2008]
- Chemical engineering simulator for styrene production
- Download here
- $n = 8$ variables and $m = 11$ constraints
- Numerical methods in the simulation: Runge-Kutta, Newton, fixed-point, secant, bisection, and other chemical engineering related solvers
- One evaluation $\approx 2$ seconds. Budget of 1,000 evaluations
- Results are machine-dependent
- Good candidate for robust optimization
- A static surrogate is available
STYRENE flowsheet
**STYRENE problem: Objective**

- Objective: Net Present Value (NPV) of the process:

\[ f(x) = \sum_{y=0}^{Y} \frac{(S_y - C_y)(1 - T_a) - I_y + D_y}{(1 + T_r)y} \]

- Direct simulator outputs for year \( y \) in \([0; Y]\):
  - \( S_y \): sales
  - \( C_y \): operating costs
  - \( I_y \): investment
  - \( D_y \): depreciation

- \( T_a = 0.4, T_r = 0.1 \): income tax and actualization rates (constants)

- Best known value: \(-33,539,100\)
## STYRENE variables

<table>
<thead>
<tr>
<th>Variable description</th>
<th>$lb$</th>
<th>$x_0$</th>
<th>$x^*$</th>
<th>$ub$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outlet temperature in block HEATER (K)</td>
<td>600</td>
<td>870</td>
<td>1,100</td>
<td>1,100</td>
</tr>
<tr>
<td>Length of reactor in block PFR (m)</td>
<td>2</td>
<td>13.88</td>
<td>16.98</td>
<td>20</td>
</tr>
<tr>
<td>Light key fraction in block SEP-STY</td>
<td>1E-4</td>
<td>0.08601</td>
<td>0.09683</td>
<td>0.1</td>
</tr>
<tr>
<td>Light key fraction in block SEP-BZ</td>
<td>1E-4</td>
<td>0.008092</td>
<td>1E-4</td>
<td>0.1</td>
</tr>
<tr>
<td>Outlet pressure of block PUMP (atm)</td>
<td>2</td>
<td>7.22</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>Split fraction in block SPLITTER</td>
<td>0.01</td>
<td>0.2599</td>
<td>0.2247</td>
<td>0.5</td>
</tr>
<tr>
<td>Air excess fraction in block FIRE</td>
<td>0.1</td>
<td>1.668</td>
<td>1.963</td>
<td>5</td>
</tr>
<tr>
<td>Cooling temperature of block COOLER (K)</td>
<td>300</td>
<td>330</td>
<td>403.0</td>
<td>500</td>
</tr>
<tr>
<td>Objective value</td>
<td>failure</td>
<td>-10,942,600</td>
<td>-33,539,100</td>
<td>failure</td>
</tr>
</tbody>
</table>

All variables are scaled from their original bounds to $[0; 100]$
STYRENE problem: Constraints

- The 11 (simulation) constraints:
  - Requirements on the structural configuration of the columns SEP-STY and SEP-BZ (2x)
  - Conditions if the mixture in block FIRE can burn (1x)
  - Regulations on CO and NOX emissions (1x)
  - Minimal purity of produced styrene and benzene (2x)
  - Minimal overall ethylbenzene conversion into styrene (1x)
  - Maximal payout time (1x)
  - Minimal discounted cash-flow rate of return (1x)
  - Maximal total investment (1x)
  - Maximal annual equivalent cost (1x)

- 4 boolean constraints and 7 quantifiable/relaxable constraints

- Hidden constraints:
  - \( \approx 57\% \) of failures when sampling
  - \( \approx 20\% \) when optimizing
Constraints handling

▶ Several solvers do not handle general constraints

▶ In this case, the Extreme Barrier (EB) approach may be used

▶ It consists to replace the objective function $f$ with

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

▶ More elaborate strategies exist (see Lecture #9)
Introduction

List of solvers

Example 1: PSwarm

Example 2: NOMAD

References
PSwarm

- Ismael Vaz and Luís Vicente, 2007 [Vaz and Vicente, 2007].
  Same origin than SID-PSM

- Free at http://www.norg.uminho.pt/aivaz/pswarm/

- Bounds and linear constraints

- Pattern Search and Particle Swarm

- Global optimization

- C and MATLAB versions

- Library mode

- Interfaces with AMPL, Python, and R

- Parallelism in C with MPI. Vectorized evaluations in MATLAB
PSwarm: MATLAB version

- **GRIEWANK**: `griewank.m` and `griewank_run.m`. The analytic function is explicitly expressed.

- We maintain our own history of evaluations via a global variable.

- **STYRENE**: `styrene.m` and `styrene_run.m`. The function is a separated executable called via the `system` command. This implies the use of temporary input and output files.

- Constraints are checked for the application of the EB.

PSwarm: Important parameters

- Starting point(s): InitPop(1).x
- Budget of evaluations: Options.MaxObj. With the default parameters, this budget is always spent, and may be not respected exactly
- Population size: Options.Size
- Search step type: Options.SearchType:
  - 0: No search
  - 1: Particle swarm
  - 2: Radial Basis Function (RBF) surrogates
  - 3: Quadratic models
- Initial mesh size: Options.InitialDelta
- Additional search directions (in file InitPatternSearch.m)
- Evaluation by blocks: Options.Vectorized
PSwarm: **Execution example**

```
>> griewank_run
Initial delta for pattern search
  240

Generating initial population

<table>
<thead>
<tr>
<th>Iter</th>
<th>Act</th>
<th>Leader</th>
<th>Objective</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
<td>1</td>
<td>Inf</td>
<td>2.400000e+02</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>1</td>
<td>5.897381e-01</td>
<td>2.400000e+02</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>1</td>
<td>1.660206e-01</td>
<td>3.750000e+00</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>1</td>
<td>1.184868e-01</td>
<td>5.859375e-02</td>
</tr>
<tr>
<td>30</td>
<td>16</td>
<td>9</td>
<td>7.496583e-02</td>
<td>7.324219e-03</td>
</tr>
<tr>
<td>40</td>
<td>16</td>
<td>9</td>
<td>6.011678e-02</td>
<td>1.171875e-01</td>
</tr>
<tr>
<td>50</td>
<td>16</td>
<td>9</td>
<td>5.917925e-02</td>
<td>3.662109e-03</td>
</tr>
<tr>
<td>60</td>
<td>16</td>
<td>9</td>
<td>5.917818e-02</td>
<td>1.144409e-04</td>
</tr>
<tr>
<td>70</td>
<td>16</td>
<td>10</td>
<td>4.565290e-02</td>
<td>1.430511e-05</td>
</tr>
<tr>
<td>80</td>
<td>16</td>
<td>10</td>
<td>4.508964e-02</td>
<td>7.324219e-03</td>
</tr>
<tr>
<td>90</td>
<td>16</td>
<td>10</td>
<td>2.738246e-02</td>
<td>5.859375e-02</td>
</tr>
</tbody>
</table>

Elapsed time is 0.572921 seconds.
Maximum number of iterations or objective function evaluations reached

ObjFunCounter: 2001
PenaltyFunCounter: 2001
RealObjFunCounter: 2001
PollSteps: 95
SuccPollSteps: 57
Degenerate: 0
IterCounter: 100
```
**GRIEWANK:** PSwarm default

Budget of 2,000 evaluations. Additional evaluations have been ignored

<table>
<thead>
<tr>
<th>$x_0$</th>
<th>$y_0$</th>
<th>$f(x_0, y_0)$</th>
<th>eval.</th>
<th>sol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.59</td>
<td>2,000</td>
<td>0.019719</td>
</tr>
<tr>
<td>33.33</td>
<td>67.40</td>
<td>2.12</td>
<td>2,000</td>
<td>1.8538E-11</td>
</tr>
<tr>
<td>93.09</td>
<td>46.99</td>
<td>3.81</td>
<td>2,000</td>
<td>2.0959E-11</td>
</tr>
<tr>
<td>25.18</td>
<td>23.44</td>
<td>1.94</td>
<td>2,000</td>
<td>0.027125</td>
</tr>
<tr>
<td>44.71</td>
<td>3.77</td>
<td>2.17</td>
<td>2,000</td>
<td>0.029584</td>
</tr>
<tr>
<td>3.77</td>
<td>16.30</td>
<td>1.48</td>
<td>2,000</td>
<td>0.027125</td>
</tr>
<tr>
<td>79.91</td>
<td>36.22</td>
<td>3.10</td>
<td>2,000</td>
<td>0.007396</td>
</tr>
<tr>
<td>71.28</td>
<td>89.08</td>
<td>4.81</td>
<td>2,000</td>
<td>5.5463E-12</td>
</tr>
<tr>
<td>11.41</td>
<td>80.72</td>
<td>2.32</td>
<td>2,000</td>
<td>0.0098647</td>
</tr>
<tr>
<td>59.28</td>
<td>55.60</td>
<td>2.61</td>
<td>2,000</td>
<td>3.3307E-16</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>2,000</td>
<td>0.0</td>
</tr>
</tbody>
</table>
**GRIEWANK**: PSwarm default

![GRIEWANK 2, f vs eval, 10 executions](image)

**PSwarm**

- GRIEWANK default x0
- GRIEWANK default x1
- GRIEWANK default x2
- GRIEWANK default x3
- GRIEWANK default x4
- GRIEWANK default x5
- GRIEWANK default x6
- GRIEWANK default x7
- GRIEWANK default x8
- GRIEWANK default x9
**STYRENE**: PSwarm default

- **STYRENE, f vs eval**

- **Objective vs number of evaluations**

- **PSWARM default**
Introduction

List of solvers

Problems

Example 1: PSwarm

Example 2: NOMAD

References
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 article in TOMS [Le Digabel, 2011]
Main functionalities (1/2)

- Single or biobjective optimization

- Variables:
  - Continuous, integer, binary, categorical, granular
  - Periodic
  - Fixed
  - Groups of variables

- Searches:
  - Latin-Hypercube (LH)
  - Variable Neighborhood Search (VNS)
  - 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)
NOMAD installation

- Pre-compiled executables are available for Windows and Mac
- Installation programs copy these executables
- On Unix/Linux, after download, launch an installation script
- Two ways to use NOMAD: batch mode or library mode
Blackbox 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 c1 c2` (objective and two constraints)
Important parameters

▶ Necessary parameters: Blackbox characteristics (dimension $n$, number of constraints, etc.), starting point ($x_0$)

▶ All algorithmic parameters have default values. The most important are:
  ▶ Maximum number of blackbox evaluations
  ▶ Starting point (more than one can be defined)
  ▶ Types of directions (more than one can be defined)
  ▶ Initial mesh size
  ▶ Constraint types
  ▶ Surrogate searches
  ▶ Seeds

▶ See the user guide for the description of all parameters, or use the `nomad -h` option
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 Quebec, 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

NOMAD run {
    BBX 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): ( 1.0000000000 1.0000000001 0.9999999997 0.9999999992 -4 ) h=-1.10134e-13 f=-4
best feasible solution : ( 1 1 1 1 -4 ) h=0 f=-4
```
Other MADS distributions

- MATLAB version within the Opti Toolbox package.
  http://www.i2c2.aut.ac.nz/Wiki/OPTI

- Available in the MATLAB Optimization Toolbox.
  Old version, not maintained

- NOMADM by M. Abramson. Old version, not maintained

- Excel with the OpenSolver tool.
  http://www.opensolver.org (GPLv3)
GRIEWANK

NOMAD, default and with VNS. Limit of 2,000 evaluations, never reached because it converged to the minimum mesh size of $10^{-13}$

<table>
<thead>
<tr>
<th>$f(x_0, y_0)$</th>
<th>PSwarm eval.</th>
<th>PSwarm sol.</th>
<th>NOMAD eval.</th>
<th>NOMAD sol.</th>
<th>NOMAD (VNS) eval.</th>
<th>NOMAD (VNS) sol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.59</td>
<td>2000</td>
<td>0.019719</td>
<td>266</td>
<td>0.0</td>
<td>1159</td>
<td>0.0</td>
</tr>
<tr>
<td>2.12</td>
<td>2000</td>
<td>1.8538E-11</td>
<td>302</td>
<td>0.088782</td>
<td>1323</td>
<td>0.0073960</td>
</tr>
<tr>
<td>3.81</td>
<td>2000</td>
<td>2.0959E-11</td>
<td>290</td>
<td>0.046835</td>
<td>1154</td>
<td>0.0098647</td>
</tr>
<tr>
<td>1.94</td>
<td>2000</td>
<td>0.027125</td>
<td>318</td>
<td>0.0073960</td>
<td>1232</td>
<td>0.039459</td>
</tr>
<tr>
<td>2.17</td>
<td>2000</td>
<td>0.029584</td>
<td>291</td>
<td>0.71736</td>
<td>1225</td>
<td>0.0</td>
</tr>
<tr>
<td>1.48</td>
<td>2000</td>
<td>0.027125</td>
<td>295</td>
<td>0.066584</td>
<td>1138</td>
<td>0.046835</td>
</tr>
<tr>
<td>3.10</td>
<td>2000</td>
<td>0.007396</td>
<td>271</td>
<td>0.059178</td>
<td>1248</td>
<td>0.078878</td>
</tr>
<tr>
<td>4.81</td>
<td>2000</td>
<td>5.5463E-12</td>
<td>280</td>
<td>0.027125</td>
<td>1276</td>
<td>0.027125</td>
</tr>
<tr>
<td>2.32</td>
<td>2000</td>
<td>0.0098647</td>
<td>288</td>
<td>0.059178</td>
<td>1361</td>
<td>0.019719</td>
</tr>
<tr>
<td>2.61</td>
<td>2000</td>
<td>3.3307E-16</td>
<td>281</td>
<td>0.02958</td>
<td>1311</td>
<td>0.027125</td>
</tr>
<tr>
<td>0.0</td>
<td>2000</td>
<td>0.0</td>
<td>194</td>
<td>0.0</td>
<td>873</td>
<td>0.0</td>
</tr>
<tr>
<td>avg</td>
<td>2000</td>
<td>0.017034</td>
<td>280</td>
<td>0.100183</td>
<td>1209</td>
<td>0.023309</td>
</tr>
</tbody>
</table>
GRIEWANK: NOMAD

GRIEWANK 2, f vs eval, 10 instances

PSWARM default
NOMAD default
NOMAD+VNS
**STYRENE: NOMAD**

![Graph showing objective vs number of evaluations for STYRENE with three solvers: PSwarm default, NOMAD default, and NOMAD EB. The graph demonstrates the performance of these solvers in terms of objective improvement over the number of evaluations.](image-url)
Introduction

List of solvers

Problems

Example 1: PSwarm

Example 2: NOMAD

References
References I

Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search.
(STYRENE).

Audet, C. and Dennis, Jr., J. (2006).

Derivative-Free and Blackbox Optimization.

Using Sampling and Simplex Derivatives in Pattern Search Methods.
(SID-PSM).

A fast and elitist multiobjective genetic algorithm: NSGA-II.
References II


References III

Optimization by direct search: New perspectives on some classical and modern methods.
(GSS).

Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm.

*Computers and Chemical Engineering, 63*:108–139.
(Benchmarks).

Benchmarking derivative-free optimization algorithms.
(Benchmarks).

HOPSPACK 2.0 User Manual.
References IV

UOBYQA: Unconstrained optimization by quadratic approximation.
*Mathematical Programming*, 92(3):555–582.

The NEWUOA software for unconstrained optimization without derivatives.

The BOBYQA algorithm for bound constrained optimization without derivatives.
Technical Report DAMTP 2009/NA06, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, England.

On fast trust region methods for quadratic models with linear constraints.
Technical Report DAMTP 2014/NA02, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, England.
(LINCOA).

Derivative-free optimization: a review of algorithms and comparison of software implementations.
(Benchmarks).