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

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:

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

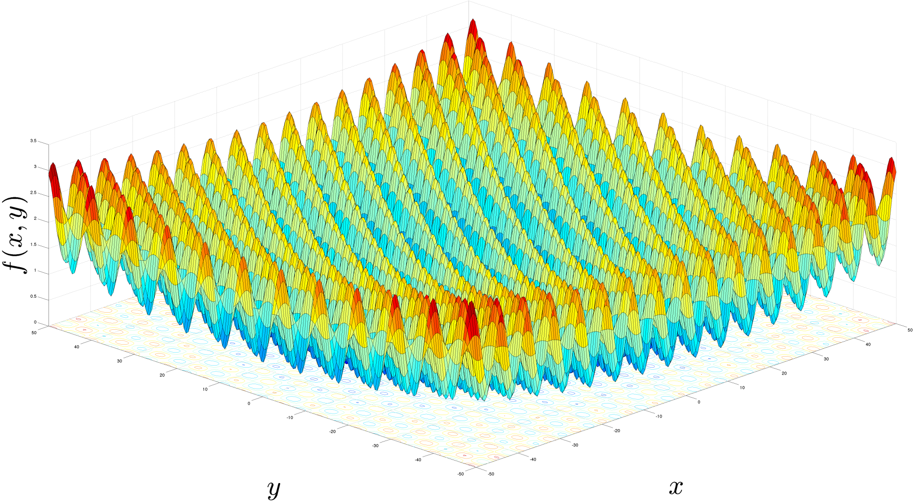
- No constraints other than bounds
- Many local optima
- Solution: $(x^*, y^*) = (0,0)$, $f(x^*, y^*) = 0$
- Budget of 2,000 evaluations

GRIEWANK problem

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

x	y	$f(x, y)$
1	1	0.589738091176242
33.33337681	67.39743216	2.120807858057156
93.08669623	46.99250073	3.813815924252485
25.18204021	23.43866215	1.943216300484015
44.70580342	3.771470648	2.169805625459778
3.77386892	16.30384942	1.479898240729798
79.90834487	36.21897984	3.102669439053978
71.28314081	89.08095617	4.809596569243840
11.40812019	80.72055663	2.315427422598158
59.28416324	55.59832326	2.610998296447039
0	0	0.0

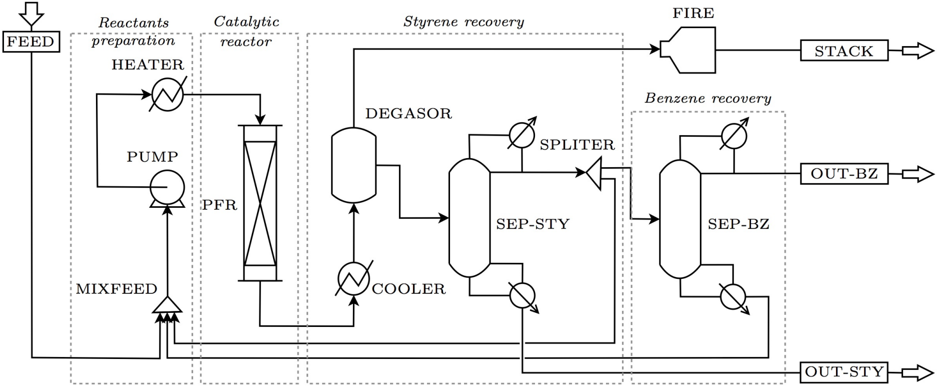
GRIEWANK function: Plot



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 $\simeq$ 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

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

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:
  - $\simeq 57\%$ of failures when sampling
  - $\simeq 20\%$ 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

Problems

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
- Tested version: `PSwarmM_v2_1`. Sept. 2014

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

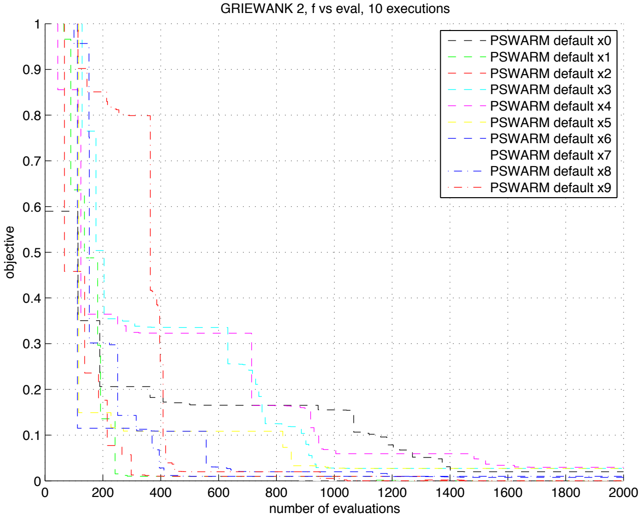
```
>> griewank_run
Initial delta for pattern search
   240

Generating initial population

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

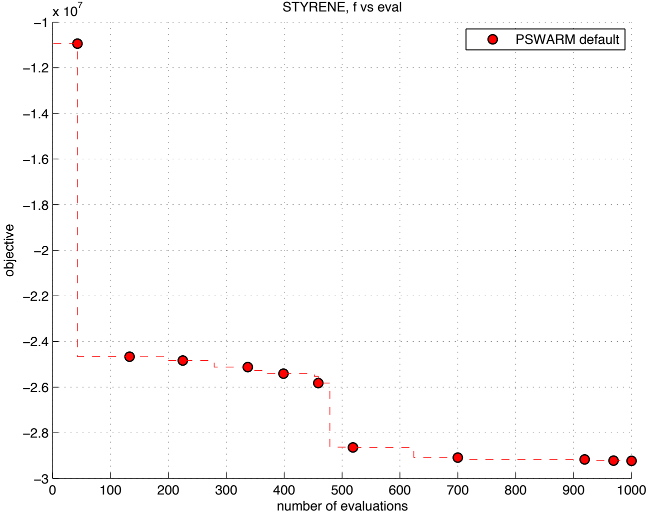
Budget of 2,000 evaluations. Additional evaluations have been ignored

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

GRIEWANK: PSwarm default



STYRENE: 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 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
```

Other MADS distributions

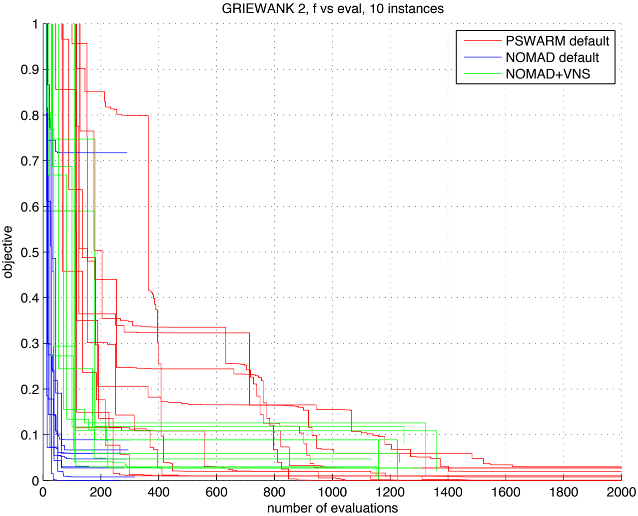
- MATLAB version within the Opti Toolbox package.
  http://www.i2c2.aut.ac.nz/Wiki/OPTI
- Available in the MATLAB Optimization Toolbox.
  Old version, not maintained
- NOMADM by M. Abramson. Old version, not maintained
- Excel with the OpenSolver tool.
  http://www.opensolver.org (GPLv3)

GRIEWANK

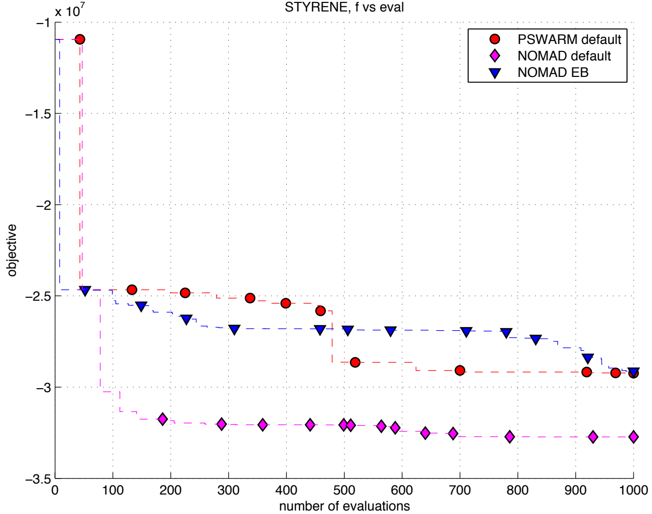
NOMAD, default and with VNS. Limit of 2,000 evaluations, never reached because it converged to the minimum mesh size of 10^{-13}

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

GRIEWANK: NOMAD



STYRENE: NOMAD



References I

Audet, C., Béchard, V., and Le Digabel, S. (2008).
Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search.
Journal of Global Optimization, 41(2):299–318.
(STYRENE).

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, Berlin.

Custódio, A. and Vicente, L. (2007).
Using Sampling and Simplex Derivatives in Pattern Search Methods.
SIAM Journal on Optimization, 18(2):537–555.
(SID-PSM).

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002).
A fast and elitist multiobjective genetic algorithm: NSGA-II.
IEEE Transactions on Evolutionary Computation, 6(2):182–197.

References II

Fowler, K., Reese, J., Kees, C., Dennis Jr., J., Kelley, C., Miller, C., Audet, C., Booker, A., Couture, G., Darwin, R., Farthing, M., Finkel, D., Gablonsky, J., Gray, G., and Kolda, T. (2008).
Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems.
Advances in Water Resources, 31(5):743–757.
(Benchmarks).

Gray, G. and Kolda, T. (2006).
Algorithm 856: APPSPACK 4.0: Asynchronous parallel pattern search for derivative-free optimization.
ACM Transactions on Mathematical Software, 32(3):485–507.

Griewank, A. (1981).
Generalized descent for global optimization.
Journal of Optimization Theory and Applications, 34(1):11–39.

Hansen, N. (2006).
The CMA Evolution Strategy: A Comparing Review.
In Lozano, J., Larrañaga, P., Inza, I., and Bengoetxea, E., editors, *Towards a New Evolutionary Computation*, volume 192 of *Studies in Fuzziness and Soft Computing*, pages 75–102. Springer Berlin Heidelberg.
(CMA-ES).

References III

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.
(GSS).

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

Martelli, E. and Amaldi, E. (2014).
PGS-COM: A hybrid method for constrained non-smooth black-box optimization problems: Brief review, novel algorithm and comparative evaluation.
Computers and Chemical Engineering, 63:108–139.
(Benchmarks).

Moré, J. and Wild, S. (2009).
Benchmarking derivative-free optimization algorithms.
SIAM Journal on Optimization, 20(1):172–191.
(Benchmarks).

Plantenga, T. (2009).
HOPSPACK 2.0 User Manual.
Technical Report SAND2009-6265, Sandia National Laboratories, Livermore, CA.

References IV

Powell, M. (2002).
UOBYQA: Unconstrained optimization by quadratic approximation.
Mathematical Programming, 92(3):555–582.

Powell, M. (2006).
The NEWUOA software for unconstrained optimization without derivatives.
In Pardalos, P., Pillo, G., and Roma, M., editors, *Large-Scale Nonlinear Optimization*, volume 83 of *Nonconvex Optimization and Its Applications*, pages 255–297. Springer.

Powell, M. (2009).
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.

Powell, M. (2014).
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).

Rios, L. and Sahinidis, N. (2013).
Derivative-free optimization: a review of algorithms and comparison of software implementations.
Journal of Global Optimization, 56(3):1247–1293.
(Benchmarks).

References V

Vaz, A. and Vicente, L. (2007).
A particle swarm pattern search method for bound constrained global optimization.
Journal of Global Optimization, 39(2):197–219.
(PSwarm).