Blackbox optimization: Part 4/4: Practical presentation

Sébastien Le Digabel

GROUPE D'ÉTUDES ET DE RECHERCHE EN ANALYSE DES DÉCISIONS

POLYTECHNIQUE MONTRÉAL TECHNOLOGICAL UNIVERSITY

ETICS 2023

Plan

- A first basic optimization with NOMAD
- The SOLAR simulator
- Benchmarking: From convergence plots to performance and data profiles

Plan

First tests

Performance and data profiles

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` (objective and two constraints)
- Example with $f(\mathbf{x}) = (1-x_1)^2 + 100(x_2 - x_1^2)^2$ (Rosenbrock function)

Run with NOMAD

- Installation of NOMAD: Download at `www.gerad.ca/nomad` or from GitHub
- NOMAD3 [Le Digabel, 2011] vs NOMAD4 [Audet et al., 2022]
- Edit a NOMAD parameter file
- All algorithmic parameters have default values

The SOLAR simulator

- Download at www.github.com/bbopt/solar
- Compilation
- Demo of the different options
- Optimization of SOLAR6 with NOMAD and CMA-ES [Hansen, 2006]

Benchmarking

- Latin Hypercube Sampling for getting 30 starting points that define 30 instances
- Convergence plots
- Performance and data profiles from [Moré and Wild, 2009]

Plan

First tests

Performance and data profiles

Profiles: Original version from the M&W paper

- $\mathcal{P}$: set of problems or instances
- $\mathcal{S}$: set of solvers, or algorithms, or methods
- Performance measure $t_{p,s} > 0$ available for each $p \in \mathcal{P}$ and $s \in \mathcal{S}$. Typically the number of evaluations required to satisfy a convergence test
- Small values of the performance measure are preferable
- Performance ratio for problem $p \in \mathcal{P}$ and solver $s \in \mathcal{S}$:

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,a} : a \in \mathcal{S}\}}$$

Convergence test

- One possible convergence test is, for the candidate solution $\mathbf{x}$:

$$f(\mathbf{x}_0) - f(\mathbf{x}) \geq (1 - \tau)(f(\mathbf{x}_0) - f_L)$$

- Where:
  - $\tau > 0$: tolerance
  - $\mathbf{x}_0$: **unique** and **feasible** starting point
  - f_L: smallest value of f obtained by any solver within a given budget of evaluations, for each $p \in \mathcal{P}$
- It requires that the reduction $f(\mathbf{x}_0) - f(\mathbf{x})$ achieved by $\mathbf{x}$ be at least $1 - \tau$ times the best possible reduction $f(\mathbf{x}_0) - f_L$
- τ represents the percentage decrease from $f(\mathbf{x}_0)$. As it decreases, the accuracy of $f(\mathbf{x})$ as an approximation to f_L increases

Performance profiles

- The best solver $s^* \in \mathcal{S}$ for a particular problem $p \in \mathcal{P}$ attains the lower bound $r_{p,s^*} = 1$
- $t_{p,s} = r_{p,s} = \infty$ when s fails to satisfy the convergence test on p
- The performance profile of s is the fraction of problems where the performance ratio is at most α:

$$\rho_s(\alpha) = \frac{1}{|\mathcal{P}|}\text{size}\{p \in \mathcal{P} : r_{p,s} \leq \alpha\}$$

- It is the probability distribution for the ratio $r_{p,s}$
- $\rho_s(1)$ is the fraction of problems for which s performs the best
- For α sufficiently large, $\rho_s(\alpha)$ is the fraction of problems solved by s
- Solvers with high values for ρ_s are preferable

Data profiles

- We are interested in the percentage of problems that can be solved, for a given tolerance τ with a variable budget of evaluations
- The data profile of Solver s is

$$d_s(\kappa) = \frac{1}{|\mathcal{P}|}\text{size}\left\{p \in \mathcal{P} : \frac{t_{p,s}}{n_p + 1} \leq \kappa\right\},$$

  where n_p is the number of variables in Problem p
- It represents the percentage of problems that can be solved with κ groups of $n_p + 1$ function evaluations, or simplex gradient estimates
- $n_p + 1$ is the number of evaluations needed to compute a one-sided finite-difference estimate of the gradient

References I

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

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.

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

Moré, J. and Wild, S. (2009).
Benchmarking Derivative-Free Optimization Algorithms.
SIAM Journal on Optimization, 20(1):172–191.