JOPT 2010

Ecole Polytechnique de Montréal

Parallel Versions of the MADS Algorithm for Black-Box Optimization

Sébastien Le Digabel
Mark A. Abramson
Charles Audet
John E. Dennis

2010–05–10

Presentation outline

Black-box optimization problems

Presentation outline

Black-box optimization problems

The MADS algorithm

Presentation outline

Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Presentation outline

Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Numerical tests

Presentation outline

Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Numerical tests

Discussion

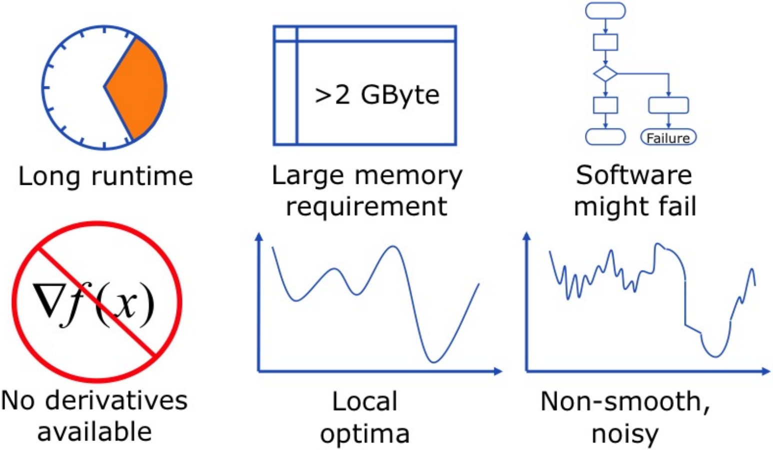
Black-box optimization problems

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).

Black-boxes as illustrated by J. Simonis [ISMP 2009]



Long runtime

Large memory requirement

Software might fail

No derivatives available

Local optima

Non-smooth, noisy

Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Numerical tests

Discussion

Mesh Adaptive Direct Search (MADS)

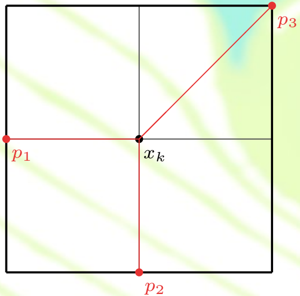
- Iterative algorithm that evaluates the black-box functions at some trial points generated on the mesh

$$M(\Delta_k) = \left\{x_k + \Delta_k D z : z \in \mathbb{N}^{n_D}\right\} \subset \mathbb{R}^n$$

where x_k is the current iterate, $\Delta_k \in \mathbb{R}^+$ is the mesh size parameter, and D a fixed set of n_D directions in $\mathbb{R}^n$.
- **Search step:** trial points can be generated anywhere on the mesh. It is typically user-provided but can also be generic.
- **Poll step:** directions are used to generate poll trial points. The normalized directions become dense in the unit sphere.
- **Updates step:** the mesh size is reduced if no new iterate is found.
- Algorithm backed by a convergence analysis based on the Clarke Calculus for nonsmooth functions.

Poll illustration (successive fails and mesh shrink)

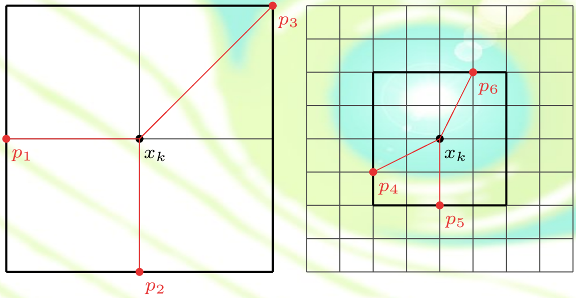
$\Delta_k = 1$



trial points$=\{p_1, p_2, p_3\}$

Poll illustration (successive fails and mesh shrink)

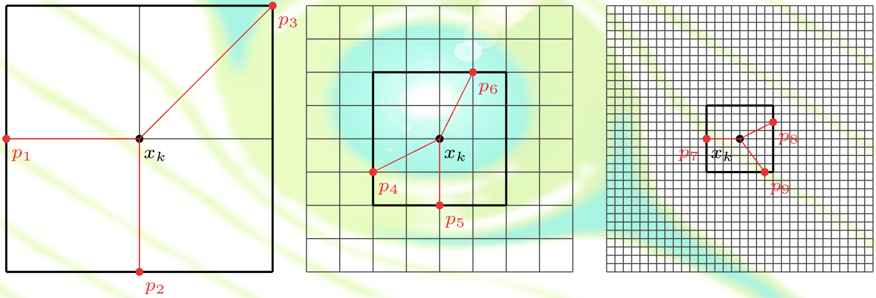
$\Delta_k = 1$ $\Delta_{k+1} = 1/4$



trial points= $\{p_1, p_2, p_3\}$ $= \{p_4, p_5, p_6\}$

Poll illustration (successive fails and mesh shrink)

$\Delta_k = 1$ $\Delta_{k+1} = 1/4$ $\Delta_{k+2} = 1/16$



trial points$=\{p_1, p_2, p_3\}$ $= \{p_4, p_5, p_6\}$ $= \{p_7, p_8, p_9\}$

Two ways of generating dense sets of poll directions

- LTMADS:
  - Random directions.
  - Random seed.
- OrthoMADS:
  - Deterministic directions.
  - Orthogonal directions with a better space coverage.
  - Default polling method in the NOMAD software.
  - $\simeq$ coordinate directions for initial values of Δ_k^m.
  - Halton seed.

pMADS

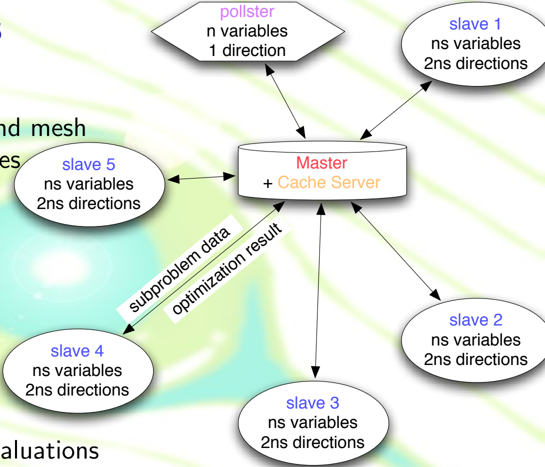
- Idea: simply evaluate the trial points in parallel.
- Synchronous version pMADS-S:
  - The iteration is over only when all the evaluations in progress are terminated.
  - Processes can be idle between two evaluations.
  - The algorithm is identical to the scalar version.
- Asynchronous version pMADS-A:
  - If a new best point is found, the iteration is terminated even if there are evaluations in progress. New trial points are then generated.
  - Processes never wait between two evaluations.
  - 'Old' evaluations are considered when they are finished.
  - The algorithm is slightly reorganized.

PSD-MADS

- **PSD:** Parallel Space Decomposition.
- Idea: each process executes a MADS algorithm on a subproblem and has responsibility of small groups of variables.
- Based on the block-Jacobi method [Bertsekas, Tsitsiklis 1989] and on the Parallel Variable Distribution [Ferris, Mangasarian 1994].
- Objective: solve larger problems ($\simeq 50 - 500$ instead of $\simeq 10 - 20$).
- (Almost) asynchronous method.
- Convergence analysis.
- Audet, Dennis, Le Digabel [SIOPT 2008].

PSD-MADS: processes

- Master
  - receives all slave's signals
  - updates current solution and mesh
  - decides subproblem variables
  - sends subproblem data
- Slaves
  - receive subproblem data
  - optimize subproblem
  - send optimization data
- Cache server
  - memorizes all black-box evaluations
  - allows the "cache search" in the pollster



COOP-MADS

- Uses a simplified version of the PSD-MADS parallel framework.
- Processes run in parallel on the original problem with different seeds in order to produce different behaviours.
- The cache server allows to share evaluations and the cache search is performed by all processes.
- (Almost) asynchronous method.

Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Numerical tests

Discussion

Numerical tests

- NOMAD: Nonsmooth Optimization with MADS:
  - MADS, pMADS-A/S, COOP-MADS and PSD-MADS (release 3.4 in May).
  - Parallelism with MPI.
  - http://www.gerad.ca/nomad (LGPL).

- Computer: 6 cores with Hyper-Threading $\simeq$ 12 processors.

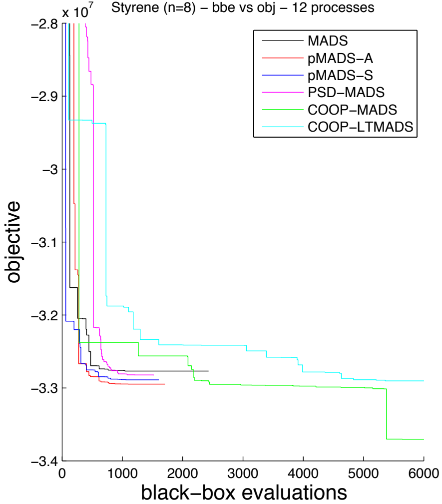
Algorithms parameters

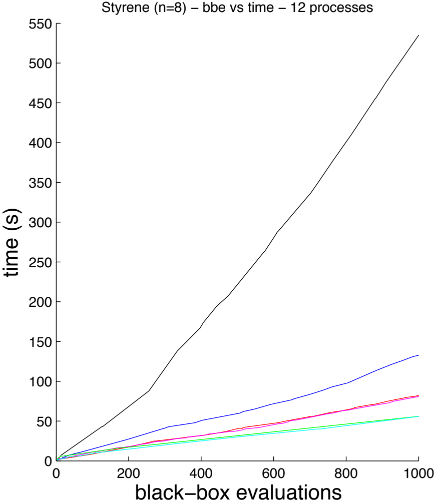
- MADS and pMADS-A/S: default parameters, `np=13` (1 master+12 slaves).
- PSD-MADS with the SIOPT paper settings:
  - `np=14`: 1 master + 1 cache server + 1 pollster slave + 11 regular slaves.
  - Max. number of evaluations for each regular slave: 10.
  - Number of free variables for each regular slave: 2.
  - Groups of variables: randomly chosen.
- COOP-MADS, `np=13` (1 cache server+12 workers).
  - COOP-MADS: each process uses OrthoMADS polling directions with different Halton seeds.
  - COOP-LTMADS: LTMADS with different random seeds.
- Stopping criteria: maximum number of black-box evaluations or a minimal mesh size.

Styrene production simulation [JOGO 2008]

- Maximize the net present value while satisfying industrial and environmental regulations.
- Written by a chemical engineer.
- Uses some common methods such as Runge-Kutta, Newton, fixed points, secant, bisection, and many other chemical engineering related solvers.
- 8 bound constrained variables,
  4 boolean unrelaxable constraints,
  7 relaxable constraints.
- 14% of trial points violate a hidden constraint.
- Evaluations can take different evaluation times (up to $\simeq 3$ seconds).

Styrene, n=8, 6,000 evaluations





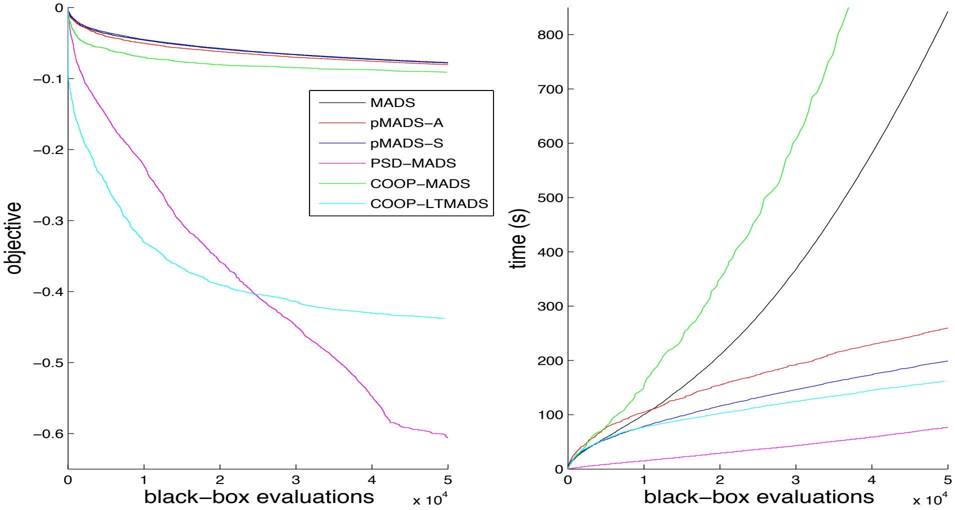
Test Problem G2 [Hedar & Fukushima, JOGO 2006]

$$\min_{x \in \mathbb{R}^n} f(x) = \left| \frac{\sum_{i=1}^{n} \cos^4 x_i - 2 \prod_{i=1}^{n} \cos^2 x_i}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$$

$$s.t. \begin{cases} g_1(x) = -\prod_{i=1}^{n} x_i + 0.75 \leq 0 \\ g_2(x) = \sum_{i=1}^{n} x_i - 7.5n \leq 0 \end{cases}$$

$n = 500$, $0 \leq x \leq 10$, $x_0 = [5\ 5\ \ldots\ 5]^T$

Problem G2, n=500, 50,000 evaluations



Results analysis

- PSD-MADS is much more efficient on the large problem.
- COOP-MADS gives the best result on the Styrene problem, and COOP-MADS is better than COOP-LTMADS.
- The bad behaviour of COOP-MADS on the large problem is due to the fact that processes never go to small meshes where OrthoMADS directions are different even for different Halton seeds: all processes are evaluating almost exactly the same points and the cache server makes them wait.
- MADS and pMADS seem equivalent. In fact, many other tests suggest that MADS gives better solutions than pMADS. This is due to the opportunistic strategy that the scalar version exploits better.

Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Numerical tests

Discussion

Discussion

- Parallel versions of MADS seem efficient and allow to solve black-box optimization problems faster and sometimes better.
- The basic parallel method pMADS gives similar results than the scalar version, but much faster.
- More evolved parallel strategies are needed to improve the method's efficiency.
- COOP-MADS seems a good method for small problems, and PSD-MADS for larger ones.
- The current PSD-MADS algorithm is based on random subproblems: statistic methods identifying the important variables should improve its efficiency even more.