PP10

Ecole Polytechnique de Montréal

Parallel Algorithms for Black-Box Optimization

Sébastien Le Digabel
Charles Audet
John Dennis

2010–02–24

Presentation outline

Black-box optimization problems

Presentation outline

Presentation outline

Presentation outline

Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Codes

Presentation outline

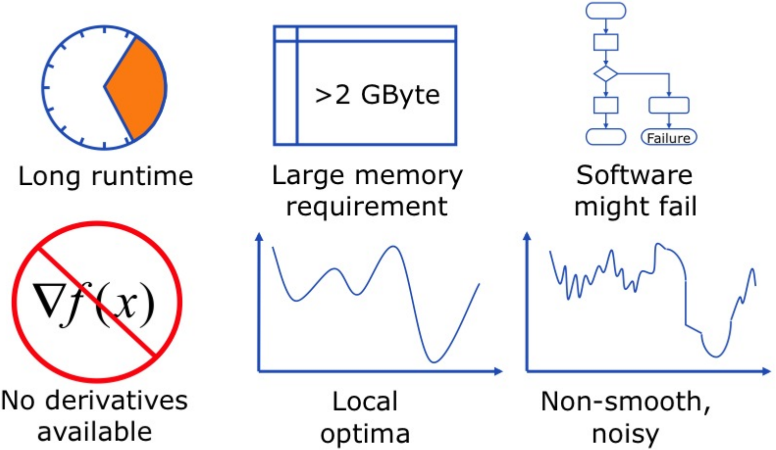
Black-box optimization problems

We consider the nonsmooth optimization problem:

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

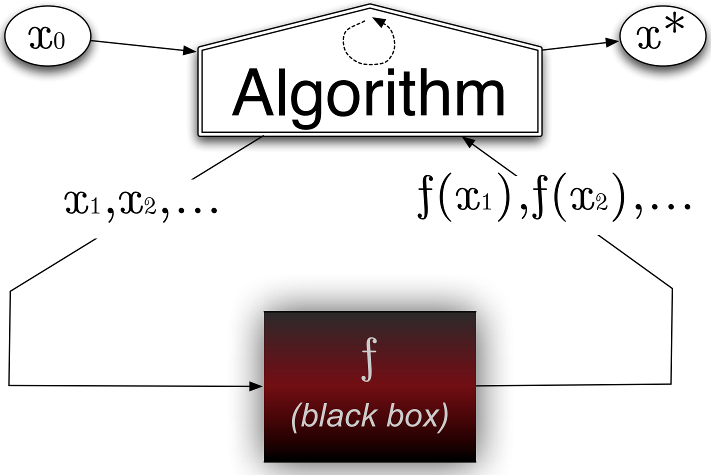
where evaluation of the functions are usually the result of a computer code (a black-box).

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



Copyright © 2009 Boeing. All rights reserved.

MADS and direct search principle

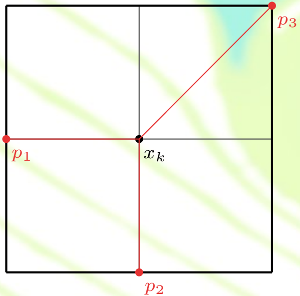


Mesh Adaptive Direct Search (MADS)

- Audet and Dennis [SIOPT, 2006]
- Iterative algorithm that evaluates the black-box functions at some trial points.
- Trial points are generated on a spatial discretization: the mesh.
- One iteration consists in generating a list of trial points constructed from poll directions. These directions grow dense.
- At the end of the iteration, the mesh size is reduced if no new iterate is found.
- Algorithm is backed by a convergence analysis based on the Clarke Calculus for nonsmooth functions.

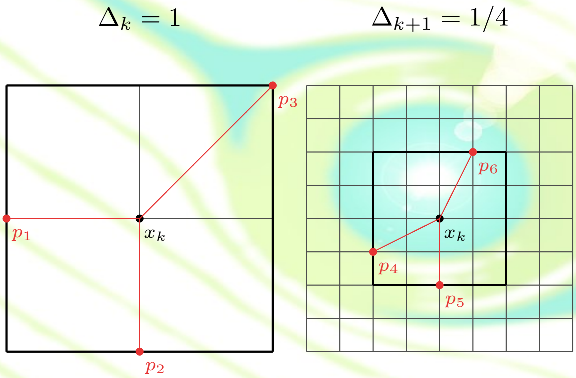
Poll illustration (successive fails and mesh shrink)

$\Delta_k = 1$



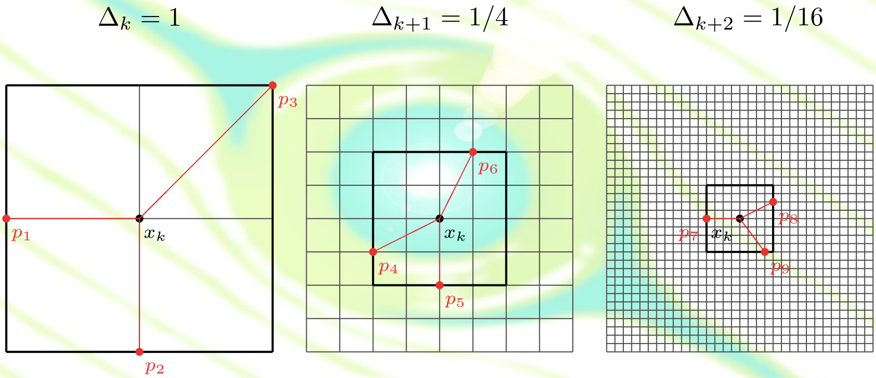
trial points$=\{p_1, p_2, p_3\}$

Poll illustration (successive fails and mesh shrink)



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

Poll illustration (successive fails and mesh shrink)



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

pMADS

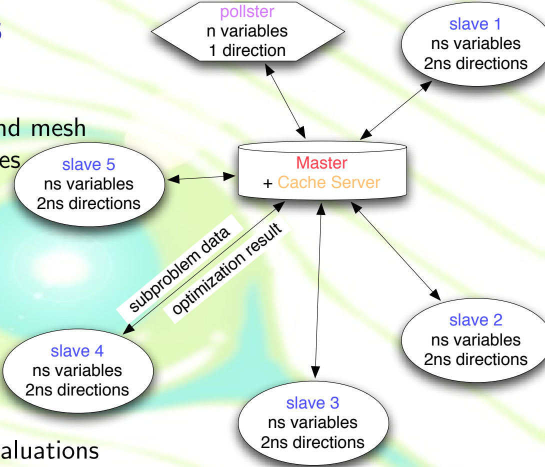
- Idea: simply evaluate the trial points in parallel.
- Synchronous version:
  - The iteration is ended only when all the evaluations in progress are terminated.
  - Processes can be idle between two evaluations.
  - The algorithm is identical to the scalar version.
- Asynchronous version:
  - If a new 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

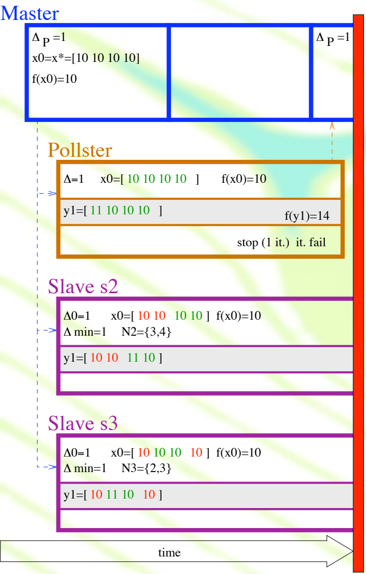
- **PSD:** Parallel Space Decomposition.
- Idea: each process executes a MADS algorithm on a subproblem and has responsibility of small groups of variables.
- Based on the block-Jacobi method [Bertsekas, Tsitsiklis 1989] and on the Parallel Variable Distribution [Ferris, Mangasarian 1994].
- Objective: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

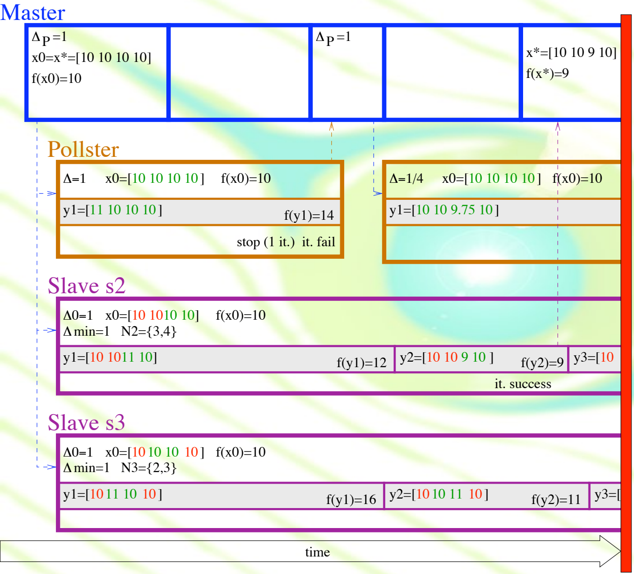
PSD-MADS: processes

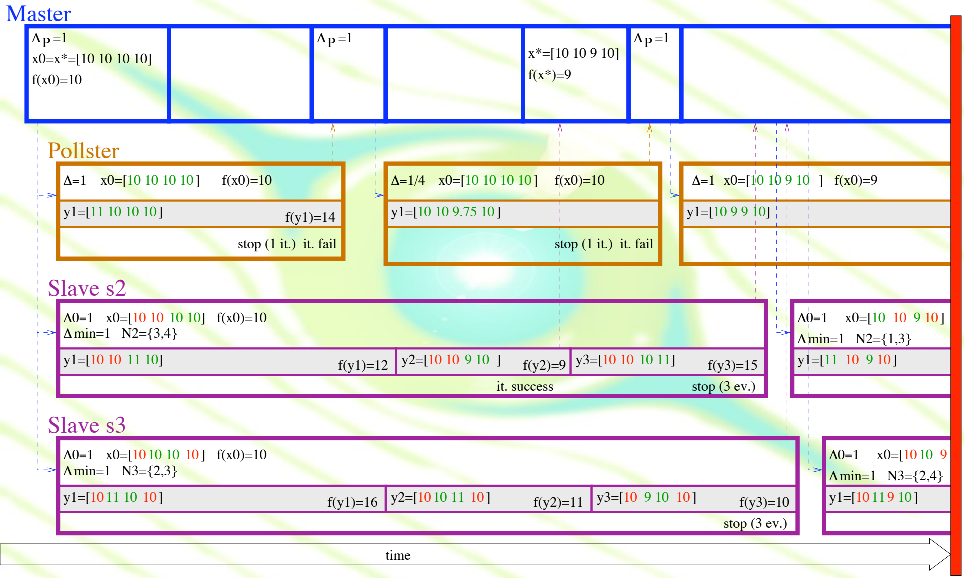
- Master
  - receives all slave's signals
  - updates current solution and mesh
  - decides subproblem variables
  - sends subproblem data
- Slaves
  - receive subproblem data
  - optimize subproblem
  - send optimization data
- Cache server
  - memorizes all black-box evaluations
  - allows the "cache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




Master
Δ P =1
x0=x*=[10 10 10 10]
f(x0)=10
time


Master
Δ P =1
x0=x*=[10 10 10 10]
f(x0)=10
Δ P =1
Pollster
Δ=1 x0=[10 10 10 10] f(x0)=10
y1=[11 10 10 10] f(y1)=14
stop (1 it.) it. fail
Slave s2
Δ0=1 x0=[10 10 10 10] f(x0)=10
Δ min=1 N2={3,4}
y1=[10 10 11 10]
Slave s3
Δ0=1 x0=[10 10 10 10] f(x0)=10
Δ min=1 N3={2,3}
y1=[10 11 10 10]
time


Master
Δ P =1
x0=x*=[10 10 10 10]
f(x0)=10
Δ P =1
x*=[10 10 9 10]
f(x*)=9
Pollster
Δ=1 x0=[10 10 10 10] f(x0)=10
y1=[11 10 10 10] f(y1)=14
stop (1 it.) it. fail
Δ=1/4 x0=[10 10 10 10] f(x0)=10
y1=[10 10 9.75 10]
Slave s2
Δ0=1 x0=[10 1010 10] f(x0)=10
Δ min=1 N2={3,4}
y1=[10 1011 10] f(y1)=12
y2=[10 10 9 10] f(y2)=9
y3=[10
it. success
Slave s3
Δ0=1 x0=[10 10 10 10] f(x0)=10
Δ min=1 N3={2,3}
y1=[10 11 10 10] f(y1)=16
y2=[10 10 11 10] f(y2)=11
y3=[
time


Master
Δp=1
x0=x*=[10 10 10 10]
f(x0)=10
Δp=1
x*=[10 10 9 10]
f(x*)=9
Δp=1
Pollster
Δ=1 x0=[10 10 10 10] f(x0)=10
y1=[11 10 10 10] f(y1)=14
stop (1 it.) it. fail
Δ=1/4 x0=[10 10 10 10] f(x0)=10
y1=[10 10 9.75 10]
stop (1 it.) it. fail
Δ=1 x0=[10 10 9 10] f(x0)=9
y1=[10 9 9 10]
Slave s2
Δ0=1 x0=[10 10 10 10] f(x0)=10
Δmin=1 N2={3,4}
y1=[10 10 11 10] f(y1)=12
y2=[10 10 9 10] f(y2)=9
y3=[10 10 10 11] f(y3)=15
it. success
stop (3 ev.)
Δ0=1 x0=[10 10 9 10]
Δmin=1 N2={1,3}
y1=[11 10 9 10]
Slave s3
Δ0=1 x0=[10 10 10 10] f(x0)=10
Δmin=1 N3={2,3}
y1=[10 11 10 10] f(y1)=16
y2=[10 10 11 10] f(y2)=11
y3=[10 9 10 10] f(y3)=10
stop (3 ev.)
Δ0=1 x0=[10 10 9
Δmin=1 N3={2,4}
y1=[10 11 9 10]
time

Free C++ codes (1/2)

- NOMAD: Nonsmooth Optimization with MADS:
  - Polytechnique Montreal.
  - MPI.
  - MADS, pMADS and PSD-MADS (March).
  - http://www.gerad.ca/nomad.
- APPSPACK: Asynchronous Parallel Pattern Search:
  - Sandia National Laboratories.
  - MPI.
  - Generalized Pattern Search.
  - http://software.sandia.gov/appspack.

Free C++ codes (2/2)

- HOPSPACK: Hybrid Optimization Parallel Search PACKage:
  - Sandia National Laboratories.
  - MPI and multithreading.
  - Multiple solvers.
  - http://software.sandia.gov/trac/hopspack.
- DAKOTA: Design Analysis Kit for Optimization and Terascale Applications:
  - Sandia National Laboratories.
  - MPI.
  - Multiple solvers.
  - http://www.cs.sandia.gov/DAKOTA.