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

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

Codes
Presentation outline

Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Codes

Discussion
Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Codes

Discussion
Black-box optimization problems

We consider the nonsmooth optimization problem:

Minimize $f(x)$
subject to $x \in \Omega$,

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

Codes

Discussion
MADS and direct search principle

Algorithm

$\mathbf{x}_0 \xrightarrow{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots} \mathbf{x}_* \xrightarrow{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots}$

$\mathbf{x}_0$ (black box)

PP10: black-box optimization 7/18
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)

\[ \Delta_k = 1 \]
\[ \Delta_{k+1} = \frac{1}{4} \]

trial points = \{p_1, p_2, p_3\} = \{p_4, p_5, p_6\}
Poll illustration (successive fails and mesh shrink)

\[ \Delta_k = 1 \quad \Delta_{k+1} = 1/4 \quad \Delta_{k+2} = 1/16 \]

trial points = \{p_1, p_2, p_3\} = \{p_4, p_5, p_6\} = \{p_7, p_8, p_9\}
Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Codes

Discussion
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
Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Codes

Discussion

Master

\[ \Delta P = 1 \]
\[ x_0 = x^* = [10 \ 10 \ 10 \ 10] \]
\[ f(x_0) = 10 \]

Pollster

\[ \Delta = 1 \]
\[ x_0 = [10 \ 10 \ 10 \ 10] \]
\[ f(x_0) = 10 \]
\[ y_1 = [11 \ 10 \ 10 \ 10] \]
\[ f(y_1) = 14 \]

Slave s2

\[ \Delta_0 = 1 \]
\[ x_0 = [10 \ 10 \ 10 \ 10] \]
\[ f(x_0) = 10 \]
\[ \Delta \text{min} = 1 \]
\[ N_2 = \{3, 4\} \]
\[ y_1 = [10 \ 10 \ 11 \ 10] \]

Slave s3

\[ \Delta_0 = 1 \]
\[ x_0 = [10 \ 10 \ 10 \ 10] \]
\[ f(x_0) = 10 \]
\[ \Delta \text{min} = 1 \]
\[ N_3 = \{2, 3\} \]
\[ y_1 = [10 \ 11 \ 10 \ 10] \]
Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Codes

Discussion

---

**Master**

\[ \Delta P = 1 \]
\[ x_0 = x^* = [10 \ 10 \ 10 \ 10] \]
\[ f(x_0) = 10 \]

\[ \Delta P = 1 \]

\[ x^* = [10 \ 10 \ 9 \ 10] \]
\[ f(x^*) = 9 \]

**Pollster**

\[ \Delta = 1 \]
\[ x_0 = [10 \ 10 \ 10 \ 10] \]
\[ f(x_0) = 10 \]

\[ y_1 = [11 \ 10 \ 10 \ 10] \]
\[ f(y_1) = 14 \]

\[ \text{stop (1 it.) it. fail} \]

\[ \Delta = 1/4 \]
\[ x_0 = [10 \ 10 \ 10 \ 10] \]
\[ f(x_0) = 10 \]

\[ y_1 = [10 \ 10 \ 9.75 \ 10] \]

**Slave s2**

\[ \Delta_0 = 1 \]
\[ x_0 = [10 \ 10 \ 10 \ 10] \]
\[ f(x_0) = 10 \]

\[ \Delta_{\text{min}} = 1 \]
\[ N_2 = \{3, 4\} \]

\[ y_1 = [10 \ 10 \ 11 \ 10] \]
\[ f(y_1) = 12 \]

\[ y_2 = [10 \ 10 \ 9 \ 10] \]
\[ f(y_2) = 9 \]

\[ y_3 = [10] \]

\[ \text{it. success} \]

**Slave s3**

\[ \Delta_0 = 1 \]
\[ x_0 = [10 \ 10 \ 10 \ 10] \]
\[ f(x_0) = 10 \]

\[ \Delta_{\text{min}} = 1 \]
\[ N_3 = \{2, 3\} \]

\[ y_1 = [10 \ 11 \ 10 \ 10] \]
\[ f(y_1) = 16 \]

\[ y_2 = [10 \ 10 \ 11 \ 10] \]
\[ f(y_2) = 11 \]

\[ y_3 = [\ldots] \]

\[ \text{time} \]
### Black-box optimization problems

<table>
<thead>
<tr>
<th>Master</th>
<th>Pollster</th>
<th>Slave s2</th>
<th>Slave s3</th>
</tr>
</thead>
</table>
| $\Delta P = 1$  
$x_0 = x^* = [10, 10, 10, 10]$  
f($x_0$) = 10 |
| $\Delta = 1$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10 |
| $\Delta = 1/4$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10 |
| $\Delta = 1$  
x_0 = [10, 10, 9, 10]  
f(x_0) = 9 |
| $\Delta = 0$  
x_0 = [10, 10, 11, 10]  
f(x_0) = 10 |
| $\Delta = 1$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10 |
| $\Delta = 0$  
x_0 = [10, 10, 9, 10]  
f(x_0) = 10 |

### The MADS algorithm

| $\Delta P = 1$  
x_0 = x^* = [10, 10, 10, 10]  
f(x_0) = 10 |
| $\Delta = 1/4$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10 |
| $\Delta = 0$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10 |

### Parallel versions of MADS

<table>
<thead>
<tr>
<th>Codes</th>
<th>Discussion</th>
</tr>
</thead>
</table>

### Discussion

- **Master**
  - $\Delta P = 1$
  - $x_0 = x^* = [10, 10, 10, 10]$  
f($x_0$) = 10
- **Pollster**
  - $\Delta = 1$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10
  - $\Delta = 1/4$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10
  - $\Delta = 1$  
x_0 = [10, 10, 9, 10]  
f(x_0) = 9
- **Slave s2**
  - $\Delta = 0$  
x_0 = [10, 10, 11, 10]  
f(x_0) = 10
  - $\Delta = 1$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10
  - $\Delta = 0$  
x_0 = [10, 10, 9, 10]  
f(x_0) = 10
- **Slave s3**
  - $\Delta = 0$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10
  - $\Delta = 1$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10
  - $\Delta = 0$  
x_0 = [10, 10, 9, 10]  
f(x_0) = 10

### Codes

- Parallel versions of MADS
- Discussion

### Discussion

- **Master**
  - $\Delta P = 1$
  - $x_0 = x^* = [10, 10, 10, 10]$  
f($x_0$) = 10
- **Pollster**
  - $\Delta = 1$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10
  - $\Delta = 1/4$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10
  - $\Delta = 1$  
x_0 = [10, 10, 9, 10]  
f(x_0) = 9
- **Slave s2**
  - $\Delta = 0$  
x_0 = [10, 10, 11, 10]  
f(x_0) = 10
  - $\Delta = 1$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10
  - $\Delta = 0$  
x_0 = [10, 10, 9, 10]  
f(x_0) = 10
- **Slave s3**
  - $\Delta = 0$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10
  - $\Delta = 1$  
x_0 = [10, 10, 10, 10]  
f(x_0) = 10
  - $\Delta = 0$  
x_0 = [10, 10, 9, 10]  
f(x_0) = 10
Black-box optimization problems

The MADS algorithm

Parallel versions of MADS

Codes

Discussion
Free C++ codes (1/2)

- NOMAD: Nonsmooth Optimization with MADS:
  - Polytechnique Montreal.
  - MPI.
  - MADS, pMADS and PSD-MADS (March).

- APPSPACK: Asynchronous Parallel Pattern Search:
  - Sandia National Laboratories.
  - MPI.
  - Generalized Pattern Search.
Free C++ codes (2/2)

- **HOPSPACK**: Hybrid Optimization Parallel Search PACKage:
  - Sandia National Laboratories.
  - MPI and multithreading.
  - Multiple solvers.

- **DAKOTA**: Design Analysis Kit for Optimization and Terascale Applications:
  - Sandia National Laboratories.
  - MPI.
  - Multiple solvers.
<table>
<thead>
<tr>
<th>Black-box optimization problems</th>
<th>The MADS algorithm</th>
<th>Parallel versions of MADS</th>
<th>Codes</th>
<th>Discussion</th>
</tr>
</thead>
</table>

**Black-box optimization problems**

**The MADS algorithm**

**Parallel versions of MADS**

**Codes**

**Discussion**