Parallel Space Decomposition of the Mesh Adaptive Direct Search algorithm

Sébastien Le Digabel
Charles Audet
John Dennis

ICCOPT-MOPTA, August 2007
## Presentation Outline

### Introduction

MADS and PSD Overview

Adaptation of PSD for MADS

Convergence analysis

Preliminary Results

Discussion
Presentation Outline

Introduction

MADS and PSD Overview

Adaptation of PSD for MADS

Convergence analysis

Preliminary Results

Discussion
Presentation Outline

Introduction

MADS and PSD Overview

Adaptation of PSD for MADS

Convergence analysis

Preliminary Results

Discussion
Presentation Outline

Introduction

MADS and PSD Overview

Adaptation of PSD for MADS

Convergence analysis

Preliminary Results

Discussion
Presentation Outline

Introduction

MADS and PSD Overview

Adaptation of PSD for MADS

Convergence analysis

Preliminary Results

Discussion
Presentation Outline

Introduction

MADS and PSD Overview

Adaptation of PSD for MADS

Convergence analysis

Preliminary Results

Discussion
Summary of the talk

- MADS is a direct search algorithm for nonsmooth optimization

- Parallel Space Decomposition (PSD) methods based on the block-Jacobi method [Bertsekas, Tsitsiklis 1989] are generic and parallel frameworks for optimization

- We propose to apply a PSD method to MADS in order to solve medium-size problems ($50 < n < 500$) $\rightarrow$ PSD-MADS
Target Class of Problems

\[\min_{x \in \Omega \subseteq \mathbb{R}^n} f(x)\]

where

- \( f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\} \)
- function \( f \) and constraints defining \( \Omega \) are black-box functions
  - nonsmooth
  - noisy
  - problematic derivative approximation
  - can possibly fail to evaluate
  - costly to evaluate (seconds, minutes, days)
  - usually the result of a computer code
MADS Overview

- **Mesh Adaptive Direct Search** [Audet, Dennis 2005]
- NOMAD project: [www.gerad.ca/nomad] [Audet, Couture]
- Extends the **Generalized Pattern Search** [Torczon 1997]
- Direct Search method: derivative are not evaluated nor approximated
- Iterative algorithm in two steps where the black-box functions are evaluated at some trial points, which are either accepted as new iterates or rejected
MADS Mesh

- All trial points at iteration $k$ are constructed to lie on a mesh

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

where $\Delta_k \in \mathbb{R}^+$ is the mesh size parameter and $D$ a fixed set of $n_D$ directions in $\mathbb{R}^n$

- After each iteration, $\Delta_k$ is reduced when no new iterate has been found (iteration failure)
Poll and Search

At iteration $k$: two steps: the Poll and the Search

- **Poll**
  - local exploration on the mesh near the best current iterate $x_k$
  - use of MADS directions (at least one is necessary to ensure convergence)
  - MADS directions are not the fixed set of directions $D$

- **Search**
  - global and flexible exploration strategy
  - has only to generate a finite number of trial points lying on the mesh
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)

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

trial points: \( \{p_1, p_2, p_3\} \)
\( = \{p_4, p_5, p_6\} \)
\( = \{p_7, p_8, p_9\} \)
MADS Convergence

- Constraints are handled with the barrier approach: if $x \notin \Omega$, $f(x)$ is considered to be $+\infty$.
- A hierarchical convergence based on $f$ differentiability analysis is available for MADS with barrier.
- **Main convergence result**: MADS leads to a Clarke stationnary point $\hat{x} \in \Omega$ if $f$ is Lipschitz near $\hat{x}$:
  $$f^\circ(\hat{x}; d) \geq 0 \text{ for all } d \in T_{\Omega}^{Cl}(\hat{x})$$
- **Corollary for unconstrained case**: if the function is strictly differentiable, then $\nabla f(\hat{x}) = 0$.
PSD methods

- Block-Jacobi method in [Bertsekas, Tsitsiklis 1989]
- Generic and parallel optimization framework
- Idea: each process works on a subproblem and has responsibility of small groups of variables
- Iterative algorithm with two steps:
  - decomposition: subproblems with a reduced number of variables are optimized in parallel
  - synchronization: results of subproblems are gathered; a new iterate is constructed
- Parallel Variable Distribution [Ferris, Mangasarian 1994]
PSD methods

Subproblem $\mathcal{P}_p : \min_{x \in \Omega_p(x^*)} f(x)$

$\Omega_p(x^*) = \{x \in \Omega : x_i = x_i^* \ \forall \ i \in \{1, 2, ..., n\} \setminus N_p\}$

Initializations
- $x_0$, lists $N_p$ of subproblems variables

Iteration $k$

[1] **Parallel Decomposition [slave $s_p$]**
- optimizes subproblem $\mathcal{P}_p$ with $x^* = x_k$
- $y_i \leftarrow$ solution of optimization

[2] **Synchronization [master]**
- $x_{k+1} \leftarrow$ new iterate from all solutions $y_p$’s
- $k \leftarrow k + 1$

**goto [1]** until some stopping condition is met
Adaptation of PSD for MADS (PSD-MADS)

- MADS is used to optimize subproblems
- The synchronization step is removed
- Main parameters:
  - $bbe$: maximum number of black-box evaluations for each MADS optimization (does not include cache hits)
  - $ns$: number of variables in subproblems
Slaves = Regulars slaves + Pollster slave

- Regular slaves: solve subproblems with standard MADS directions on a reduced number of variables ($N_p$)
- Pollster slave
  - all $n$ variables are considered
  - single-polls: only one direction per poll
  - terminates after one iteration
  - the set of normalized poll directions grows dense in the unit-sphere
Processes occupation

- **Master**
  - receives all slave’s signals
  - updates current solution and mesh
  - decides subproblem variables
  - sends subproblem data

- **Slaves (regular and pollster)**
  - receive subproblem data
  - optimize subproblem
  - send optimization data

- **Cache server**
  - memorizes all black-box evaluations
  - allows the “cache search” for regular slaves
Choice of the sets $N_p$

- $N_p$ are the sets of variables for subproblem $\mathcal{P}_p$ optimized by regular slave $s_p$

- Original PSD method:
  - fixed sets $N_p$ for all iterations
  - sets $N_p$ had to form a partition of $\{1, 2, ..., n\}$

- PSD-MADS:
  - sets are not required to form a partition of $\{1, 2, ..., n\}$
  - sets may change for a same slave between two optimizations
  - Implementation: sets are randomly and uniformly chosen
  - Implementation: all sets have the same size $ns$
Convergence from the pollster’s perspective

- The pollster runs a complete MADS algorithm on the original problem:
  - The poll contains a single direction
  - Search
    - consists in obtaining the best cache point
    - by construction, slaves generate a finite number of points on the pollster mesh
  - Global mesh size $\Delta_P$: link between pollster and slaves:
    $$\Delta_{\text{pollster}} \leq \Delta_P \leq \Delta_{\text{reg.slaves}}$$

- All MADS convergence conditions are verified: the MADS theoretical convergence analysis holds
\[ \Delta P = 1 \]
\[ x_0 = x^* = [10, 10, 10, 10] \]
\[ f(x_0) = 10 \]
Introduction

MADS and PSD Overview

PSD-MADS

Convergence analysis

Preliminary Results

Discussion

**Master**

\[ \Delta P = 1 \]

\[ x_0 = x* = [10, 10, 10, 10] \]

\[ f(x_0) = 10 \]

**Pollster**

\[ \Delta = 1 \]

\[ x_0 = [10, 10, 10, 10] \]

\[ f(x_0) = 10 \]

\[ y_1 = [11, 10, 10, 10] \]

\[ f(y_1) = 14 \]

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

**Slave s2**

\[ \Delta = 1 \]

\[ x_0 = [10, 10, 10, 10] \]

\[ f(x_0) = 10 \]

\[ \Delta \text{min} = 1 \]

\[ N_2 = \{3, 4\} \]

\[ y_1 = [10, 10, 11, 10] \]

**Slave s3**

\[ \Delta = 1 \]

\[ x_0 = [10, 10, 10, 10] \]

\[ f(x_0) = 10 \]

\[ \Delta \text{min} = 1 \]

\[ N_3 = \{2, 3\} \]

\[ y_1 = [10, 11, 10, 10] \]
Introduction
MADS and PSD Overview
PSD-MADS
Convergence analysis
Preliminary Results
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

\[ \Delta = 1 \]
\[ x_0 = [10\ 10\ 10\ 10] \]
\[ f(x_0) = 10 \]
\[ \Delta_{\text{min}} = 1 \]
\[ N_2 = \{3,4\} \]
\[ y_1 = [10\ 10\ 11\ 10] \]
\[ f(y_1) = 12 \]
\[ y_2 = [10\ 10\ 9\ 10] \]
\[ f(y_2) = 9 \]

\[ \text{it. succeed} \]

Slave s3

\[ \Delta = 1 \]
\[ x_0 = [10\ 10\ 10\ 10] \]
\[ f(x_0) = 10 \]
\[ \Delta_{\text{min}} = 1 \]
\[ N_3 = \{2,3\} \]
\[ y_1 = [10\ 11\ 10\ 10] \]
\[ f(y_1) = 16 \]
\[ y_2 = [10\ 10\ 11\ 10] \]
\[ f(y_2) = 11 \]
\[ y_3 = [\_\_\_\_\_\_] \]

\[ \text{time} \]
Introduction
MADS and PSD Overview
PSD-MADS
Convergence analysis
Preliminary Results
Discussion

Master

\[ \Delta p = 1 \]
\[ x_0 = x^* = \begin{bmatrix} 10 & 10 & 10 & 10 \end{bmatrix} \]
\[ f(x_0) = 10 \]

\[ \Delta p = 1 \]
\[ x^* = \begin{bmatrix} 10 & 10 & 9 & 10 \end{bmatrix} \]
\[ f(x^*) = 9 \]

\[ \Delta p = 1 \]

Pollster

\[ \Delta = 1 \quad x_0 = \begin{bmatrix} 10 & 10 & 10 & 10 \end{bmatrix} \]
\[ f(x_0) = 10 \]
\[ y_1 = \begin{bmatrix} 11 & 10 & 10 & 10 \end{bmatrix} \]
\[ f(y_1) = 14 \]
stop (1 it.) it. fail

\[ \Delta = \frac{1}{4} \]
\[ x_0 = \begin{bmatrix} 10 & 10 & 10 & 10 \end{bmatrix} \]
\[ f(x_0) = 10 \]
\[ y_1 = \begin{bmatrix} 10 & 10 & 9.75 & 10 \end{bmatrix} \]

\[ \Delta = 1 \quad x_0 = \begin{bmatrix} 10 & 10 & 9 & 10 \end{bmatrix} \]
\[ f(x_0) = 9 \]
\[ y_1 = \begin{bmatrix} 10 & 9 & 9 & 10 \end{bmatrix} \]

Slave s2

\[ \Delta_0 = 1 \quad x_0 = \begin{bmatrix} 10 & 10 & 10 & 10 \end{bmatrix} \quad f(x_0) = 10 \]
\[ \Delta \min = 1 \quad N_2 = \{3, 4\} \]
\[ y_1 = \begin{bmatrix} 10 & 10 & 11 & 10 \end{bmatrix} \]
\[ f(y_1) = 12 \]
\[ y_2 = \begin{bmatrix} 10 & 10 & 9 & 10 \end{bmatrix} \]
\[ f(y_2) = 9 \]
\[ y_3 = \begin{bmatrix} 10 & 10 & 10 & 11 \end{bmatrix} \]
\[ f(y_3) = 15 \]
\[ \text{it. success} \]
stop (1 it.) it. fail

Slave s3

\[ \Delta_0 = 1 \quad x_0 = \begin{bmatrix} 10 & 10 & 10 & 10 \end{bmatrix} \quad f(x_0) = 10 \]
\[ \Delta \min = 1 \quad N_3 = \{2, 3\} \]
\[ y_1 = \begin{bmatrix} 10 & 11 & 10 & 10 \end{bmatrix} \]
\[ f(y_1) = 16 \]
\[ y_2 = \begin{bmatrix} 10 & 10 & 11 & 10 \end{bmatrix} \]
\[ f(y_2) = 11 \]
\[ y_3 = \begin{bmatrix} 10 & 9 & 10 & 10 \end{bmatrix} \]
\[ f(y_3) = 10 \]
\[ \text{stop (3 ev.)} \]

\[ \Delta_0 = 1 \quad x_0 = \begin{bmatrix} 10 & 10 & 9 \end{bmatrix} \]
\[ \Delta \min = 1 \quad N_3 = \{2, 4\} \]
\[ y_1 = \begin{bmatrix} 10 & 11 & 9 & 10 \end{bmatrix} \]
\[ f(y_1) = 16 \]
\[ y_2 = \begin{bmatrix} 10 & 10 & 11 & 10 \end{bmatrix} \]
\[ f(y_2) = 11 \]
\[ y_3 = \begin{bmatrix} 10 & 9 & 10 & 10 \end{bmatrix} \]
\[ f(y_3) = 10 \]
\[ \text{stop (3 ev.)} \]
Test Problem G2 from [Hedar, Fukushima 2006]

\[
\min_{x \in \mathbb{R}^n} f(x) = \left| \sum_{i=1}^{n} \cos^4 x_i - 2 \prod_{i=1}^{n} \cos^2 x_i \right| \sqrt{\sum_{i=1}^{n} ix_i^2} \\
\begin{align*}
g_1(x) &= - \prod_{i=1}^{n} x_i + 0.75 \leq 0 \\
g_2(x) &= \sum_{i=1}^{n} x_i - 7.5n \leq 0
\end{align*}
\]

\[n = 250, \ 0 \leq x_i \leq 10, \ x_0 = [5 \ 5 \ ... \ 5]^T\]
Testing protocols

- Graphs showing the number of evaluations v.s the objective function value
- Each plot is an average of 30 runs
- Different PSD-MADS runs are compared to the basic parallel version of MADS
- PSD-MADS parameters tested: \( bbe \) and \( ns \)
- Budget of 25000 evaluations
- 14 processes
## Introduction

MADS and PSD Overview

PSD-MADS

Convergence analysis

Preliminary Results

Discussion

<table>
<thead>
<tr>
<th>Introduction</th>
<th>MADS and PSD Overview</th>
<th>PSD-MADS</th>
<th>Convergence analysis</th>
<th>Preliminary Results</th>
<th>Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Preliminary Results

**HF_G2_250 : PSD–MADS [ns=20]**

- **MADS**
- **bbe=1**
- **bbe=2**
- **bbe=3**
- **bbe=5**
- **bbe=10**
- **bbe=100**

- Lower is better

**f(x)** vs. **black–box evals x10^4**

![Graph](image-url)
HF_G2_250 : PSD–MADS [bbe=5]

- Lower is better

f(x)

black–box evals x10^4

0 0.5 1 1.5 2 2.5

x 10^4

0 0.5 1 1.5 2 2.5

x 10^4

0 0.5 1 1.5 2 2.5

x 10^4

0 0.5 1 1.5 2 2.5

x 10^4

0 0.5 1 1.5 2 2.5

x 10^4
Discussion

- PSD-MADS: new algorithm applying the PSD parallel framework to MADS
- Promising results for a large problem
- Convergence results of MADS still hold
- Work in progress:
  - PVD synchronization → new PSD-MADS recomposition
  - include the PVD “forget-me-not” terms
  - compare results with APPS: Asynchronous Parallel Pattern Search [Hough, Kolda, Torczon 2001]

Questions?
Discussion

- PSD-MADS: new algorithm applying the PSD parallel framework to MADS
- Promising results for a large problem
- Convergence results of MADS still hold
- Work in progress:
  - PVD synchronization → new PSD-MADS recomposition
  - include the PVD “forget-me-not” terms
  - compare results with APPS: Asynchronous Parallel Pattern Search [Hough, Kolda, Torczon 2001]
- Questions?