Parallel Variable Distribution for Mesh Adaptive Direct Search

Sébastien Le Digabel
Charles Audet
John Dennis

Optimization Days, May 2007

Presentation Outline

Introduction

MADS and PVD Overview

Adaptation of PVD for MADS

Convergence

Preliminary Results

Discussion

Presentation Outline

Presentation Outline

Introduction

MADS and PVD Overview

Adaptation of PVD for MADS

Convergence

Preliminary Results

Discussion

Presentation Outline

Presentation Outline

Introduction

MADS and PVD Overview

Adaptation of PVD for MADS

Convergence

Preliminary Results

Discussion

Presentation Outline

Introduction

- MADS is a direct search algorithm for nonsmooth optimization
- PVD is a generic and parallel framework for optimization
- We propose to apply PVD to MADS in order to solve large problems ($n > 50$)

Target Problem

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

where

- $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$
- function f and constraints defining Ω are black-box functions
  - nonsmooth
  - problematic derivative approximation
  - can possibly fail to evaluate
  - costly to evaluate
  - usually the result of a computer code

MADS Overview

- **M**esh **A**daptive **D**irect **S**earch [Audet, Dennis 2005]
- Extends the **G**eneralized **P**attern **S**earch [Torczon 1997]
- Direct Search method: derivative are not evaluated nor approximated
- Iterative algorithm where the black-box functions are evaluated at some trial points, which are either accepted as new iterates or rejected
- At iteration k: two steps: the Poll and the Search

MADS Mesh

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

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

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

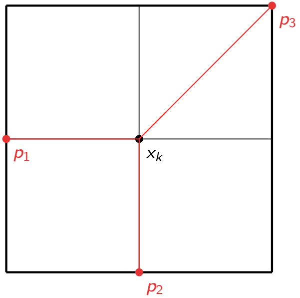
- After each iteration, Δ_k is reduced when no new iterate has been found (iteration fail)

Poll and 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)
- 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)

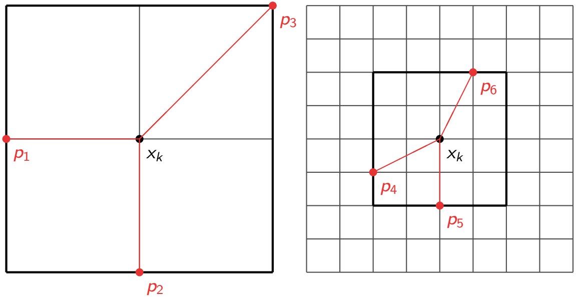
$\Delta_k = 1$



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

Poll illustration (successive fails and mesh shrink)

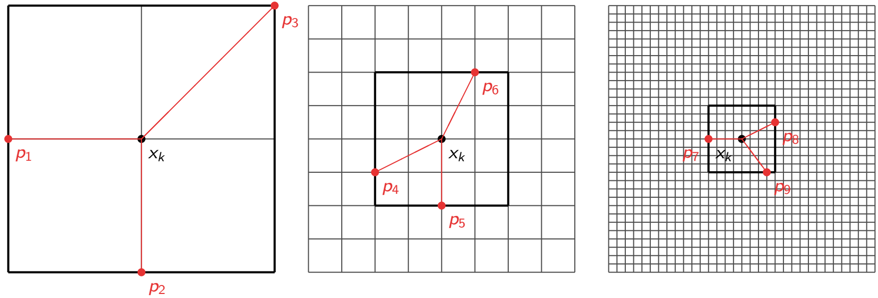
$\Delta_k = 1$ $\Delta_{k+1} = 1/4$



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

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 $\nabla f(\hat{x}) = 0$

PVD algorithm

- **P**arallel **V**ariable **D**istribution [Ferris, Mangasarian 1994]
- Generic and parallel optimization framework
- Parallelism achieved by a master/slaves paradigm
- Idea: each process works on a reduced problem and has responsability 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

PVD algorithm for N processors

1 master, $(N-1)$ slaves

Initializations

x_0, lists of subproblems variables

Iteration k

[1] **Parallel Decomposition [by slave s_i]**

optimizes subproblem P_i from starting point x_k
$y_i \leftarrow$ solution of optimization

[2] **Synchronization [by master]**

$x_{k+1} \leftarrow$ new iterate from solutions y_i's
$k \leftarrow k+1$
goto [1] until a stopping condition is met

Adaptation of PVD for MADS

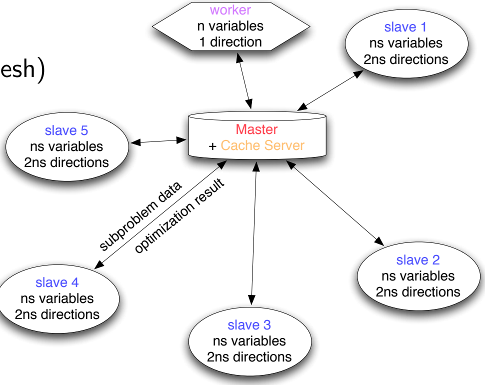
- MADS is used to optimize subproblems
- The synchronization barrier step is removed
- New algorithm most important parameters:
  - *bbe* : maximum number of black-box evaluations for each MADS optimization (does not include cache hits)
  - *ns* : number of variables in subproblems

Slaves and Worker

- Slaves: solve subproblems with standard MADS method, with a reduced number of variables
- Worker
  - is one of the slaves with special status
  - all n variables are considered
  - polls with only one MADS special direction: this direction is defined in the MADS method as the minimal direction allowing convergence

New method: master/slaves occupation

- Master
  - look for a slave signal
  - get optimization data
  - updates (current solution, mesh)
  - decide subproblem variables
  - send subproblem data
- Slave or Worker
  - receive subproblem data
  - optimize subproblem
  - send optimization data



Choice of sets L_i

- L_i are the sets of variables for subproblem P_i optimized by slave s_i
- Original PVD method:
  - fixed sets L_i for all iterations
  - sets L_i had to be a partition of $\{1, 2, ..., n\}$
- New method:
  - $L_i \longrightarrow L_i^k$ for variables of slave s_i at iteration k
  - sets have not to form a partition of $\{1, 2, ..., n\}$
  - sets are randomly and uniformly chosen
  - all sets have the same size ns

Mesh Update

- For all MADS optimizations, an initial mesh $M(\Delta_0)$ and a minimal mesh $M(\Delta_{min})$ are to be defined
- Initially: all slaves begin optimizations with user specified Δ_0
- Worker
  - $\Delta_0 \leftarrow$ last Δ_k of worker optimization
  - if the current solution has been updated by another slave, initial mesh is expanded (search success)
  - $\Delta_{min} \leftarrow$ user specified small value
  - $\Delta_{PVD} \leftarrow \Delta_0$
- Slaves
  - $\Delta_0 \leftarrow$ last Δ_k of slave optimization
  - $\Delta_{min} \leftarrow \Delta_{PVD}$: all trial points lie on mesh $M(\Delta_{PVD})$

Convergence

- **From the point of view of the Worker process**
- The Worker runs a complete MADS algorithm on the original problem :
  - A single direction Poll
  - Search
    - is performed after an overal *bbe* trial poll points
    - consists in obtaining the best iterate from slaves
    - by construction, slaves generate a finite number of points on the worker mesh
- All MADS convergence conditions are met: MADS theoretical convergence analysis holds

Test Problem [JOGO; Hedar, Fukushima]

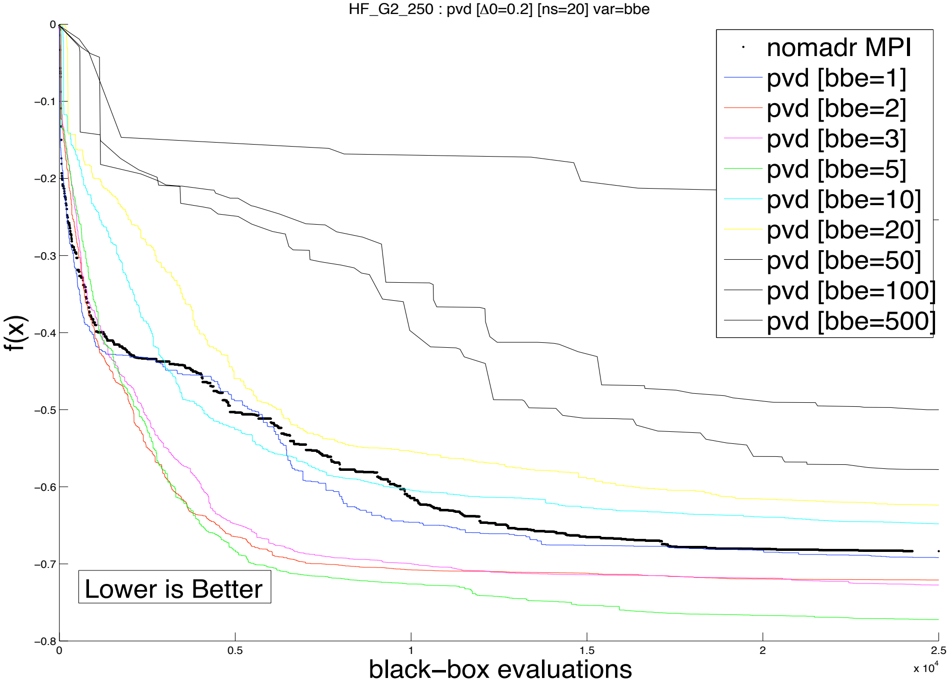
$$\min_{x \in \mathbb{R}^n} f(x) = \left| \frac{\sum_{i=1}^{n} \cos^4 x_i - 2 \prod_{i=1}^{n} \cos^2 x_i}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$$

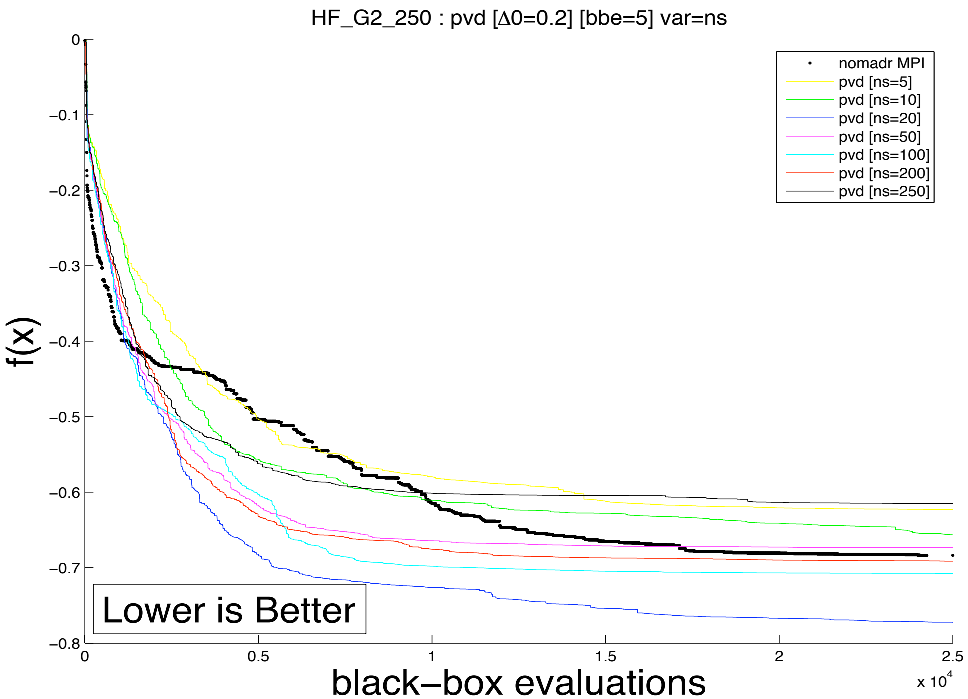
$$s.t. \begin{cases} g_1(x) = -\prod_{i=1}^{n} x_i + 0.75 \leq 0 \\ g_2(x) = \sum_{i=1}^{n} x_i - 7.5n \leq 0 \end{cases}$$

$n = 250,\ 0 \leq x_i \leq 10,\ x_0 = [5\ 5\ \ldots\ 5]^T$

Testing protocols

- Graphs showing the number of evaluations *v.s* the objective function value
- Each plot is an average of 5 runs
- Different pvd runs are compared to a synchronous parallel MADS algorithm
- PVD parameters tested: *bbe* and *ns*
- Budget of 25000 evaluations
- 12 slaves


HF_G2_250 : pvd [Δ0=0.2] [ns=20] var=bbe
nomadr MPI
pvd [bbe=1]
pvd [bbe=2]
pvd [bbe=3]
pvd [bbe=5]
pvd [bbe=10]
pvd [bbe=20]
pvd [bbe=50]
pvd [bbe=100]
pvd [bbe=500]
Lower is Better
f(x)
black−box evaluations
x 10^4
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
−0.7
−0.8
0
0.5
1
1.5
2
2.5


HF_G2_250 : pvd [Δ0=0.2] [bbe=5] var=ns
nomadr MPI
pvd [ns=5]
pvd [ns=10]
pvd [ns=20]
pvd [ns=50]
pvd [ns=100]
pvd [ns=200]
pvd [ns=250]
Lower is Better
f(x)
black–box evaluations
x 10^4
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
−0.7
−0.8
0
0.5
1
1.5
2
2.5

First observations

- Promising preliminary results
- Efficient runs are obtained with small values of parameter *bbe* (best value is a maximum of 5 black-box true evaluations for each subproblem)

Discussion

- New algorithm applying the PVD parallel framework to MADS
- Promising results for large problems
- Convergence results of MADS still hold
- Work in progress:
  - original PVD synchronization $\rightarrow$ new PVD recomposition
  - compare results with APPS (Asynchronous Parallel Pattern Search [Kolda 2005])
- Questions ?

Discussion

- New algorithm applying the PVD parallel framework to MADS
- Promising results for large problems
- Convergence results of MADS still hold
- Work in progress:
  - original PVD synchronization $\rightarrow$ new PVD recomposition
  - compare results with APPS (Asynchronous Parallel Pattern Search [Kolda 2005])
- Questions ?