Direct Search methods

MTH8418

S. Le Digabel, Polytechnique Montréal

Winter 2020

(v3)
Plan

Generalized Pattern Search (GPS)

Mesh Adaptive Direct Search (MADS)

Other direct-search methods

References
Blackbox optimization problems

- We consider the optimization problem

\[
\min_{x \in \Omega} f(x)
\]

where the evaluations of \( f \) and the functions defining \( \Omega \) are usually the result of a computer code (a blackbox)

- Extreme Barrier approach: We focus on the optimization of \( f_{\Omega} \) instead of \( f \), with \( f_{\Omega}(x) = \begin{cases} f(x) & \text{if } x \in \Omega \\ \infty & \text{otherwise} \end{cases} \)

- Better treatments of constraints are studied in Lesson #9
Generalized Pattern Search (GPS)

Mesh Adaptive Direct Search (MADS)

Other direct-search methods

References
Introduction

- **Generalized Pattern Search** (GPS) [Torczon, 1997]
- Generalization of the CS and H&J methods

Mesh at iteration \( k \):

\[
M_k = \bigcup_{x \in V_k} \{ x + \Delta_k Dz : z \in \mathbb{N}^{nD} \}
\]

- **Mesh size parameter**: \( \Delta_k \in \mathbb{R}^+ \)
- **Cache**: Set \( V_k \) of the points already evaluated by the start of iteration \( k \)

- **Directions**:
  - Matrix \( D \) of dimension \( n \times n_D \)
  - One column in \( D = \) One direction in \( \mathbb{R}^n \)
  - Typically \( D \) is taken as \([I_n - I_n]\) (as in CS)
[0] **Initializations** \((x_0, \Delta_0)\)

[1] **Iteration** \(k\)

[1.1] (global) **Search**
- select a finite number of mesh points
- sort these points
- evaluate candidates opportunistically

[1.2] (local) **Poll** (if the Search failed)
- construct poll set \(P_k = \{x_k + \Delta_k d : d \in D_k\}\)
- sort\((P_k)\)
- evaluate candidates opportunistically

[2] **Updates**
- if success
  - \(x_{k+1} \leftarrow\) success point
  - possibly increase \(\Delta_k\)
- else
  - \(x_{k+1} \leftarrow x_k\)
  - decrease \(\Delta_k\)
  - \(k \leftarrow k + 1\), stop or go to [1]
The Poll (1/2)

- **Poll set:**
  \[ P_k = \{ x_k + \Delta_k d : d \in D_k \} \]
  where \( x_k \) is the current incumbent, or the poll center

- **Poll directions:** A positive spanning set \( D_k \subset \mathbb{R}^n \) where each direction \( d \in D_k \) is one of the directions of \( D \)

- The directions correspond typically to a **minimal positive basis** \((n + 1 \text{ directions})\) or a **maximal positive basis** \((2n \text{ directions})\)
The trial points in $P_k$ are evaluated following the opportunistic strategy: evaluations are interrupted as soon as a new better solution is found.

Trial points ordering is then crucial in practice. It can be based on:

- Model or surrogate values
- Angle with the gradient of a model
- Angle with the last direction of success
- etc.
The Search

- Flexible and global search strategy
- Is executed prior to the poll step
- Is valid for a finite number of mesh points
- Users can define their own strategy, specific to their problem
- Generic strategies exist (random, LH, speculative, etc.)
Mesh size update

- Update $\Delta_{k+1} \leftarrow \tau^\omega \Delta_k$ with $\tau > 1$ rational and $\omega \in \{\omega^-, \omega^+\}$, with $\omega^-$ integer $\leq -1$, $\omega^+$ integer $\geq 0$

- CS: $\tau = 2$, $\omega^+ = 0$, $\omega^- = -1$

- GPS default: $\tau = 2$, $\omega^+ = 1$, $\omega^- = -1$
Coordinate Search (CS): Polling directions

\[ P_k = \{x_k \pm \Delta_k e_i\}; \ 2n \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

\[ \Delta_k = 1 \]
Coordinate Search (CS): Polling directions

\[ P_k = \{ x_k \pm \Delta_k e_i \}; \text{ } 2n \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

\[ \Delta_k = 1 \]
Coordinate Search (CS): Polling directions

\[ P_k = \{ x_k \pm \Delta_k e_i \}; \quad 2n \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

\[
\Delta_k = 1 \\
\Delta_{k+1} = \frac{1}{2}
\]
Coordinate Search (CS): Polling directions

\[ P_k = \{ x_k \pm \Delta_k e_i \}; \quad 2n \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

\[ \Delta_k = 1 \]

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

Always the same 2 directions
Coordinate Search (CS): Polling directions

\[ P_k = \{x_k \pm \Delta_k e_i\}; \text{ } 2n \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

\[ \Delta_k = 1 \]  \hspace{1cm}  \[ \Delta_{k+1} = \frac{1}{2} \]  \hspace{1cm}  \[ \Delta_{k+2} = \frac{1}{4} \]
Coordinate Search (CS): Polling directions

\[ P_k = \{ x_k \pm \Delta_k e_i \}; \ 2n \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

\[ \Delta_k = 1 \quad \Delta_{k+1} = \frac{1}{2} \quad \Delta_{k+2} = \frac{1}{4} \]
Coordinate Search (CS): Polling directions

\[ P_k = \{ x_k \pm \Delta_k e_i \}; \quad 2n \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

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

Always the same \(2n = 4\) directions
GPS: Example of poll directions

\[ P_k = \{ x_k + \Delta_k d : d \in D_k \}; \ n + 1 \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

\[ \Delta_k = 1 \]
**GPS: Example of poll directions**

\[ P_k = \{ x_k + \Delta_k d : d \in D_k \} ; \ n + 1 \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

\[ \Delta_k = 1 \]
GPS: Example of poll directions

\[ P_k = \{ x_k + \Delta_k d : d \in D_k \} ; \quad n + 1 \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

\[ \Delta_k = 1 \quad \Delta_{k+1} = \frac{1}{2} \]
GPS: Example of poll directions

\[ P_k = \{x_k + \Delta_k d : d \in D_k\}; \ n + 1 \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

\[ \Delta_k = 1 \]
\[ \Delta_{k+1} = \frac{1}{2} \]
GPS: Example of poll directions

\[ P_k = \{ x_k + \Delta_k d : d \in D_k \} \]; \( n + 1 \) mesh points at distance \( \Delta_k \) from \( x_k \)

\[ \Delta_k = 1 \quad \Delta_{k+1} = \frac{1}{2} \quad \Delta_{k+2} = \frac{1}{4} \]
**GPS: Example of poll directions**

\[ P_k = \{ x_k + \Delta_k d : d \in D_k \}; \; n + 1 \text{ mesh points at distance } \Delta_k \text{ from } x_k \]

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

14 different ways of defining \( D_k \) on this mesh
GPS: Poll and Search
GPS: Poll and Search

Search points
GPS: Poll and Search

Search points

Poll points
GPS: Poll and Search

Search points
Poll points

Success

$x_{k+1} = p^2$
GPS: Poll and Search

Search points

Poll points

Success

\[ x_{k+1} = p^2 \]
**GPS: Poll and Search**

*Search points*

*Poll points*

---

**Failure**

\[ x_{k+1} = x_k \]
GPS: Poll and Search

Search points

Poll points

Failure

\[ x_{k+1} = x_k \]
Convergence analysis

If the series of iterates \( \{x_k\} \) belongs to a bounded set, then

- \( \lim_{k} \Delta_k = 0 \) for any infinite subset of indices
- There exists \( \hat{x} \in \mathbb{R}^n \) the limit of a subsequence of mesh local minimizers for meshes that get infinitely fine: \( x_k \to \hat{x} \), with \( f(x_k + \Delta_k d) \geq f(x_k) \) for all \( d \in D_k \), and \( k \in K \)
Convergence analysis

If the series of iterates \( \{x_k\} \) belongs to a bounded set, then

1. \( \lim_{k} \Delta_k = 0 \) for any infinite subset of indices
2. There exists \( \hat{x} \in \mathbb{R}^n \) the limit of a subsequence of mesh local minimizers for meshes that get infinitely fine: \( x_k \to \hat{x} \), with \( f(x_k + \Delta_k d) \geq f(x_k) \) for all \( d \in D_k \), and \( k \in K \)

**Theorem**

*If \( f \) Lipschitz near \( \hat{x} \), then \( f^\circ(\hat{x}; d) \geq 0 \) for all directions \( d \in D \) used infinitely many times*
Convergence analysis

If the series of iterates \( \{x_k\} \) belongs to a bounded set, then

- \( \lim_{k} \Delta_k = 0 \) for any infinite subset of indices
- There exists \( \hat{x} \in \mathbb{R}^n \) the limit of a subsequence of mesh local minimizers for meshes that get infinitely fine: \( x_k \to \hat{x} \), with \( f(x_k + \Delta_k d) \geq f(x_k) \) for all \( d \in D_k \), and \( k \in K \)

**Theorem**

If \( f \) Lipschitz near \( \hat{x} \), then \( f^\circ(\hat{x}; d) \geq 0 \) for all directions \( d \in D \) used infinitely many times

**Proof:**

\[
f^\circ(\hat{x}; d) := \limsup_{y \to \hat{x}, \ t \downarrow 0} \frac{f(y + td) - f(y)}{t}
\]
Convergence analysis

If the series of iterates \( \{x_k\} \) belongs to a bounded set, then

\[ \lim_{k} \Delta_k = 0 \]

for any infinite subset of indices

There exists \( \hat{x} \in \mathbb{R}^n \) the limit of a subsequence of mesh local minimizers for meshes that get infinitely fine: \( x_k \rightarrow \hat{x} \), with

\[ f(x_k + \Delta_k d) \geq f(x_k) \]

for all \( d \in D_k \), and \( k \in K \)

Theorem

If \( f \) Lipschitz near \( \hat{x} \), then \( f^\circ(\hat{x}; d) \geq 0 \) for all directions \( d \in D \) used infinitely many times

Proof:

\[ f^\circ(\hat{x}; d) := \limsup_{y \rightarrow \hat{x}, \ t \downarrow 0} \frac{f(y + td) - f(y)}{t} \geq \limsup_{k \in K} \frac{f(x_k + \Delta_k d) - f(x_k)}{\Delta_k} \]
Convergence analysis

If the series of iterates \( \{x_k\} \) belongs to a bounded set, then

1. \( \lim_{k} \Delta_k = 0 \) for any infinite subset of indices

2. There exists \( \hat{x} \in \mathbb{R}^n \) the limit of a subsequence of mesh local minimizers for meshes that get infinitely fine: \( x_k \to \hat{x} \), with \( f(x_k + \Delta_k d) \geq f(x_k) \) for all \( d \in D_k \), and \( k \in K \)

**Theorem**

*If \( f \) Lipschitz near \( \hat{x} \), then \( f^\circ(\hat{x}; d) \geq 0 \) for all directions \( d \in D \) used infinitely many times*

**Proof:**

\[
\begin{align*}
    f^\circ(\hat{x}; d) := \limsup_{y \to \hat{x}, \ t \downarrow 0} \frac{f(y + td) - f(y)}{t} \\
    \geq \limsup_{k \in K} \frac{f(x_k + \Delta_k d) - f(x_k)}{\Delta_k} \geq 0
\end{align*}
\]

**Note:** These directions form a positive spanning set
Convergence analysis

If the series of iterates \( \{x_k\} \) belongs to a bounded set, then

- \( \lim_{k} \Delta_k = 0 \) for any infinite subset of indices
- There exists \( \hat{x} \in \mathbb{R}^n \) the limit of a subsequence of mesh local minimizers for meshes that get infinitely fine: \( x_k \to \hat{x} \), with \( f(x_k + \Delta_k d) \geq f(x_k) \) for all \( d \in D_k \), and \( k \in K \)

**Theorem**

- If \( f \) Lipschitz near \( \hat{x} \), then \( f^\circ(\hat{x}; d) \geq 0 \) for all directions \( d \in D \) used infinitely many times
- If \( f \) regular near \( \hat{x} \), then \( f'(\hat{x}; d) \geq 0 \) for all directions \( d \in D \) used infinitely many times
- If \( f \) strictly differentiable near \( \hat{x} \), then \( \nabla f(\hat{x}) = 0 \)
Limitations

- In order to move from a non-optimal point, the poll step has to generate descent directions inside the tangent cone.

- With general constraints, it is impossible to identify all the tangent cone generators.

- The biggest limitation is then the fixed and limited number of possible directions.
Generalized Pattern Search (GPS)

Mesh Adaptive Direct Search (MADS)

Other direct-search methods

References
The MADS acronyms

- **MADS** (Mesh Adaptive Direct Search)
  → The algorithmic framework without the definition of the polling directions

- **LT-MADS**: Original MADS implementation

- **OrthoMADS**: Second MADS implementation

- **QR-MADS**: Van Dyke and Asaki MADS implementation

- **NOMAD** (Nonlinear Optimization with the MADS algorithm):
  → The software package. Includes LT-MADS and OrthoMADS

- NOMADS does not exist
Mesh Adaptive Direct Search (MADS)

- [Audet and Dennis, Jr., 2006]
- Generalization of GPS
- Better convergence result: If $f$ Lipschitz near $\hat{x}$, it is a Clarke stationary point:

$$f^\circ(\hat{x}; d) \geq 0 \text{ for all } d \in T^C_\Omega(\hat{x})$$

($T^C_\Omega$ is a generalization of the tangent cone)
Directions

- The directions $D_k \subset \mathbb{R}^n$ are not taken in $D$. But each direction $d \in D_k$ can be written as a nonnegative integer combination of directions of $D$.

- The set of normalized directions grows dense in the unit sphere, i.e.:

  For all $\varepsilon > 0$, and for all $d \in \mathbb{R}^n$, there exists one MADS direction $d_k$ such that $\left\| \frac{d_k}{\|d_k\|} - \frac{d}{\|d\|} \right\| < \varepsilon$.
Mesh

- The GPS mesh size parameter $\Delta_k$ is replaced by the MADS mesh size parameter $\Delta^m_k$, and the new poll size parameter $\Delta^p_k$ is introduced.

- The mesh and the poll set remain the same with
  
  $M_k = \bigcup_{x \in V_k} \{ x + \Delta^m_k D z : z \in \mathbb{N}^{nD} \}$ and
  
  $P_k = \{ x_k + \Delta^m_k d : d \in D_k \}$, but now we have $\|\Delta^m_k d\| \approx \Delta^p_k$.

- $\Delta^m_k \leq \Delta^p_k$ at each iteration $k$, and $\Delta^m_k$ is reduced faster than $\Delta^p_k$.

- Typically:
  
  - $\Delta^m_{k+1} \leftarrow 4 \Delta^m_k$ or $\Delta^m_k / 4$
  
  - $\Delta^p_{k+1} \leftarrow 2 \Delta^p_k$ or $\Delta^p_k / 2$
  
  - $\Delta^p_k = \sqrt{\Delta^m_k}$, with $\Delta^m_k, \Delta^p_k \leq 1$
Poll illustration (successive fails and mesh shrinks)

\[ \Delta^m_k = \Delta^p_k = 1 \]

poll trial points = \{t_1, t_2, t_3\}
Poll illustration (successive fails and mesh shrinks)

\[ \Delta^m_k = \Delta^p_k = 1 \]

\[ \Delta^m_{k+1} = \frac{1}{4} \]
\[ \Delta^p_{k+1} = \frac{1}{2} \]

poll trial points = \{t_1, t_2, t_3\} = \{t_4, t_5, t_6\}
Poll illustration (successive fails and mesh shrinks)

\[ \Delta_{k}^{m} = \Delta_{k}^{p} = 1 \]
\[ \Delta_{k+1}^{m} = \frac{1}{4} \]
\[ \Delta_{k+1}^{p} = \frac{1}{2} \]
\[ \Delta_{k+2}^{m} = \frac{1}{16} \]
\[ \Delta_{k+2}^{p} = \frac{1}{4} \]

Poll trial points:
\[ \{t_1, t_2, t_3\} \]
\[ \{t_4, t_5, t_6\} \]
\[ \{t_7, t_8, t_9\} \]
MADS implementations

- MADS is a general framework. It defines the conditions on the directions, but do not define the direction themselves.

- There are several implementations:
  - LT-MADS: Based on Lower-Triangular random matrices [Audet and Dennis, Jr., 2006]
  - QR-MADS: Based on the QR decomposition and on normally distributed directions [Van Dyke and Asaki, 2013]
  - OrthoMADS: Quasi-random, deterministic, and orthogonal directions. Current default in NOMAD [Abramson et al., 2009]
OrthoMADS

$x_k$
OrthoMADS
OrthoMADS

\[ x_k \]
OrthoMADS
OrthoMADS
OrthoMADS
OrthoMADS

Set of normalized directions is dense in the unit sphere

infinite number of directions
OrthoMADS

Set of normalized directions is dense in the unit sphere

- OrthoMADS is deterministic
- At each iteration, directions are orthogonal

infinite number of directions
OrthoMADS is based on the quasi-random Halton sequence [Halton, 1960] in order to generate a sequence of vectors \( \{u_t\}_{t=1}^{\infty} \) dense in \([0; 1]^n\).

- \( u_t \) is transformed into \( \frac{2u_t - e}{\|2u_t - e\|} \) on the unit sphere, with \( e = (1, 1, \ldots, 1) \), and scaled.

- The latter is projected to the current mesh (\( \rightarrow q_{t,\ell} \)).
The Householder transformation is applied:

\[ H = \| q_{t,\ell} \|^2 I_n - 2q_{t,\ell}q_{t,\ell}^\top \]

By construction, \( H \) is an integer orthogonal basis of \( \mathbb{R}^n \).

The poll directions are the columns of \( H \) and \(-H\).
OrthoMADS: Dense directions
OrthoMADS $n + 1$

- [Audet et al., 2014]

- For various reasons, OrthoMADS is preferred to LT-MADS. LT-MADS defines $n + 1$ and $2n$ types of directions, and OrthoMADS had only the $2n$ variant

<table>
<thead>
<tr>
<th></th>
<th>LT-MADS</th>
<th>OrthoMADS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n + 1$</td>
<td>2006</td>
<td>2014</td>
</tr>
<tr>
<td>$2n$</td>
<td>2006</td>
<td>2009</td>
</tr>
</tbody>
</table>

- Some tests suggested that the LT-MADS implementation was more efficient with $n + 1$ directions

- This more recent OrthoMADS variant uses $n + 1$ directions as well
General framework #1

Idea: Given a poll set of $2n$ trial points, prune it to $n$ points and add a direction to obtain $n + 1$ points

Poll at iteration $k$

\[
P_k^o = \{x_k + \Delta_k d : d \in D_k^o\} \text{ (original poll set)}
\]

extract $D'_k \subset D_k^o$

compute new direction $d_k$

\[
D_k = D'_k \cup \{d_k\}
\]

construct $P_k = \{x_k + \Delta_k d : d \in D_k\} \text{ (reduced poll set)}$

sort($P_k$)

evaluate($P_k$) (opportunistically)
OrthoMads $n + 1$ with framework #1

- $D_k^o = [H_k - H_k]$ is the original OrthoMADS spanning set with $2n$ directions and $H_k \in \mathbb{Z}^{n \times n}$ an orthogonal basis with integer coefficients.

- The selection of $n$ columns of $D_k^o$ to obtain $D_k'$ is based on a target direction $w \in \mathbb{R}^n$.

- The target direction is taken as the last direction of success.

- The $(n + 1)^{th}$ direction is $d_k = - \sum_{d \in D_k'} d$. 

\[ D_k^o = [H_k - H_k] \]
OrthoMads $n + 1$ with framework #1: Idea

2n directions $D_k^0$

$n + 1$ directions $D_k = D'_k \cup \{d_k\}$
OrthoMads $n + 1$ with framework #1: Idea

2n directions $D_k^0$

$n + 1$ directions $D_k = D_k' \cup \{d_k\}$
Completion using function values

- Second and more general framework

- This version is not limited to OrthoMADS and may be applied to any poll sets. For example hybrid versions with more than $2n$ points

- The first framework is decomposed allowing to evaluate $n$ trial points in a first step and possibly one last $(n + 1)^{th}$ point

\[ y_k = x_k + d_k \Delta_k \]

- $y_k$ is constructed by exploiting the function values at the first $n$ points

- $y_k$ must lie on the mesh and $d_k$ must be inside the cone of the negative directions of $D'_k$ so that the poll directions remain a positive spanning set
General framework #2

Poll at iteration $k$

\[ P_k^o = \{ x_k + \Delta_k d : d \in D_k^o \} \] (original poll set)

extract $D_k' \subset D_k^o$ and construct $P_k'$

sort $(P_k')$

evaluate $(P_k')$ (opportunistically)

Success

interrupt iteration

Failure

compute new direction $d_k$

evaluate $(x_k + \Delta_k d_k)$
Use of quadratic models

Quadratic models may be used at two different levels:

1. In Framework #1: The simplex gradient is taken as the target direction $w$

2. In Framework #2: Optimize a model to determine the last trial point $y_k = x_k + \Delta_k d_k$
Generalized Pattern Search (GPS)

Mesh Adaptive Direct Search (MADS)

Other direct-search methods

References
Other direct-search methods (1/2)

- **Hooke and Jeeves:**
  - [Hooke and Jeeves, 1961]
  - Original Pattern Search method
  - CS directions
  - Precursor of the Search step: “Exploratory moves”
  - Introduced the term “Direct-Search”

- **Implicit filtering:**
  - [Winslow et al., 1991]
  - Simplex gradients using second-order approximations
  - Line search
  - Quasi-Newton update
  - IFFCO, IMFIL software packages
Other direct-search methods (2/2)

- **DIRECT:**
  - [Jones et al., 1993]
  - DIviding RECTangles
  - Global optimization
  - The space is divided into hyperrectangles, and the most promising ones are divided again into smaller hyperrectangles
  - The blackbox is evaluated at the center of the hyperrectangles
  - DIRECT software

- **GSS:**
  - [Kolda et al., 2003]
  - Generating Set Search
  - GPS with Search and Poll steps
  - Additional directions conforming to bounds and linear constraints
  - HOPSPACK software
Generalized Pattern Search (GPS)

Mesh Adaptive Direct Search (MADS)

Other direct-search methods

References
References I


References II


References III
