Nonsmooth Optimization by combining MADS and VNS

Charles Audet
Vincent Béchard
Sébastien Le Digabel
École Polytechnique de Montréal

May 2006
Introduction

MADS Algorithm

VNS Metaheuristic

Coupling of MADS and VNS

Preliminary Results

Conclusion
Presentation Outline

Introduction

MADS Algorithm

VNS Metaheuristic

Coupling of MADS and VNS

Preliminary Results

Conclusion
Presentation Outline

Introduction

MADS Algorithm

VNS Metaheuristic

Coupling of MADS and VNS

Preliminary Results

Conclusion
Presentation Outline

Introduction

MADS Algorithm

VNS Metaheuristic

Coupling of MADS and VNS

Preliminary Results

Conclusion
Presentation Outline

Introduction

MADS Algorithm

VNS Metaheuristic

Coupling of MADS and VNS

Preliminary Results

Conclusion
Presentation Outline

Introduction
MADS Algorithm
VNS Metaheuristic
Coupling of MADS and VNS
Preliminary Results
Conclusion
Introduction

MADS is an algorithm for nonsmooth optimization
VNS is a metaheuristic (most of the time) for combinatorial optimization
This work presents a way to incorporate VNS into MADS
This is natural because:
- MADS has a flexible step allowing the introduction of heuristics
- MADS defines a discrete structure of the variable space, easy to use as VNS neighborhoods
- These algorithms have a complementary behaviour (MADS search is more diversified when new solutions are found whereas VNS search is more diversified when no improvement are made)
Problem presentation

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

where

- \( f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\} \)
- objective function \( f \) and functions defining \( \Omega \) are
  - nonsmooth, costly, can possibly fail to evaluate, derivative approximation is problematic
  - viewed as unexploitable black-box functions
MADS Overview

- **MADS**: Mesh Adaptive Direct Search [Audet, Dennis]
- **NOMAD** is the c++ implementation of MADS (freely available at www.gerad.ca/nomad) [Couture]
- MADS generalizes the Generalized Pattern Search (GPS, [Torczon]) Algorithm
- **Main convergence result**: MADS leads to a Clarke-KKT stationary point $\hat{x} \in \Omega$ if $f$ is Lipschitz near $\hat{x}$
MADS Overview

- The black-box functions are evaluated at some trial points, which are either accepted as new iterates or rejected
- Constraints are handled by a filter method determining which new iterates to accept
- All trial points are constructed to lie on a mesh

\[ M(k, \Delta_k) = \{ x_k + \Delta_k Dz : z \in \mathbb{N}^{nD} \} \subset \mathbb{R}^n \]

where \( \Delta_k \in \mathbb{R}^+ \) is the mesh size parameter and \( D \) a fixed set of directions in \( \mathbb{R}^n \)
- After each iteration, the mesh size parameter \( \Delta_k \) is reduced when no new iterate has been found (iteration fail)
- Each MADS iteration has two steps, the **Search** and the **Poll**
MADS Poll

- Local exploration near the best current iterate $x_k$
- A set of direction $D_k$ is randomly chosen. In GPS these directions had to be taken in the global set of directions $D$, but MADS allows a larger choice with the use of a second mesh size parameter $\Delta^p_k$
- The set of poll trial points (the **poll frame**) is then constructed:

$$P_k = \{x_k + \Delta_k d : d \in D_k\} \subseteq M(k, \Delta_k)$$

- The poll is rigidly defined (mesh update, directions used) to ensure convergence results
Poll illustration (successive fails and mesh shrink)

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

\[ P_k = \{p_1, p_2, p_3\} \]
Poll illustration (successive fails and mesh shrink)

\[ \Delta_k = 1 \]
\[ \Delta^p_k = 1 \]
\[ \Delta_{k+1} = \frac{1}{4} \]
\[ \Delta^p_{k+1} = \frac{1}{2} \]

\[ P_k = \{ p_1, p_2, p_3 \} \]
\[ P_{k+1} = \{ p_4, p_5, p_6 \} \]
Poll illustration (successive fails and mesh shrink)

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

\[
P_k = \{p_1, p_2, p_3\} \\
P_{k+1} = \{p_4, p_5, p_6\} \\
P_{k+2} = \{p_7, p_8, p_9\}
\]
MADS Search

- The search is a flexible global search strategy
- A valid search must only generate a finite number of points lying on the mesh
- User can use a problem specific search
- There are also generic searches (Random Search, Latin Hypercube Sampling)
[0] **Initializations**

\[ x_0 \in X, \Delta_0 \in \mathbb{R}^+ \]
\[ k \leftarrow 0 \]

[1] **Poll and search step**

**Search step**

- evaluate the functions on a finite number of points of \( M(k, \Delta_k) \)

**Poll step**

- compute \( p \) MADS directions \( D_k \in \mathbb{R}^{n \times p} \)
- construct the frame \( P_k \subseteq M(k, \Delta_k) \) with \( x_k, D_k \) and \( \Delta_k \)
- evaluate the functions on the \( p \) points of \( P_k \)

[2] **Updates**

- determine the type of success of iteration \( k \)
- solution update \((x_{k+1})\)
- mesh update \((\Delta_{k+1})\)
- \( k \leftarrow k + 1 \)
- check the stopping conditions
- goto [1]
VNS Overview

- **VNS**: Variable Neighborhood Search [Hansen, Mladenović]
- More often used in combinatorial optimization but can be applied in the continuous case
- It is based on a local search (descent) and on a perturbation method (shaking) allowing to get away from local optima
- The perturbation method is parametrized by $\xi_k$ and increasingly changes the current solution when $\xi_k$ grows
- The search is more and more global when no improvements are made
VNS illustration

$X_k$
VNS illustration

\[ x' = \text{shaking}(x_k, 1) \]

\[ x_k \]
VNS illustration

\[ x' = shaking(x_k, 1) \]

\[ x_k \quad x'' = descent(x') \]
VNS illustration

\[ x_{k+1} \]
VNS illustration

\[ x' = \text{shaking}(x_k, 2) \]

\[ x_{k+1} \]
VNS illustration

\[ x' = \text{shaking}(x_k, 2) \]

\[ x_{k+1} \ x'' = \text{descent}(x') \]
VNS illustration

$x_{k+2}$
VNS illustration

\[ x' = \text{shaking}(x_k, 3) \]

\[ x_{k+2} \]
VNS illustration

\[ x' = \text{shaking}(x_k, 3) \]

\[ x'' = \text{descent}(x') \]
VNS Illustration

$X_{k+3}$
[0] **Initializations**

\[ \xi_{\text{max}}, \xi_0, \delta \in \mathbb{N}^+, \ x_0 \in X \]

\[ k \leftarrow 0 \]

[1] **while** \( (\xi_k \leq \xi_{\text{max}}) \)

\[ x' \leftarrow \text{shaking}(x_k, \xi_k) \]

\[ x'' \leftarrow \text{descent}(x') \]

**if** \( (f(x'') < f(x_k)) \)

\[ x_{k+1} \leftarrow x'' \]

\[ \xi_{k+1} \leftarrow \xi_0 \]

**else**

\[ x_{k+1} \leftarrow x_k \]

\[ \xi_{k+1} \leftarrow \xi_k + \delta \]

\[ k \leftarrow k + 1 \]
The main contribution of this work is the incorporation of VNS into the search step of MADS. This new VNS search only has to generate a finite number of mesh points in order to keep the convergence properties of MADS. The two VNS components (descent and shaking) are defined using the mesh of MADS.
VNS shaking

- The mesh defines a natural structure for the perturbation method which can be seen as a function:
  \[
  \text{shaking} : (M(k, \Delta_k), \mathbb{N}) \rightarrow M(k, \Delta_V) \subseteq M(k, \Delta_k) \quad \text{and} \quad x' \leftarrow \text{shaking}(x, \xi_k)
  \]

- \(\xi_k \in \mathbb{N}\) is the **perturbation amplitude**

- The fixed-size mesh \(M(k, \Delta_V) \subseteq M(k, \Delta_k)\) allows the perturbation to be based only on the amplitude \(\xi_k\) in order to remain independent of the current mesh size parameter \(\Delta_k\)

- \(\Delta_V\) is called the **VNS trigger** (VNS search only occurs at iteration \(k\) when \(\Delta_k \leq \Delta_V\) and \(\Delta_V = \ell \Delta_k\) for some \(\ell \in \mathbb{N}\))

- If \(D = [I - I]^T\), the perturbed point \(x'\) can be chosen so that
  \[
  \|x_k - x'\|_\infty = \xi_k \Delta_V
  \]
Examples of meshes $M(k, \Delta_k)$ (gray), $M(k, \Delta_V)$ (black) and possible choices for the perturbation (points $x^i$ on the bold frame at distance $\xi_k \Delta_V$ of $x_k$)

$\text{shaking}(x_k, 2) \in \{x^1, \ldots, x^8\}$

$\text{shaking}(x_k, 3) \in \{x^1, \ldots, x^{24}\}$

$\Delta_V = \Delta_k$, $\xi_k \Delta_V = 2\Delta_k$

$\Delta_V = 4\Delta_k$, $\xi_k \Delta_V = 12\Delta_k$
VNS descent

- Function \( descent : M(k, \Delta_V) \rightarrow M(k, \Delta_k) \) and \( x'' \leftarrow descent(x') \)
- Use of a specific poll step, with its own mesh size parameter and its own filter
- Cannot reduce the current mesh size
- Strategies to reduce the evaluations cost of the descent:
  - Uses surrogate functions if available
  - Compare the descent trial points with the points in cache (DS strategy)
[0] **Initializations**

\[ x_0 \in X, \Delta_0 \in \mathbb{R}^+, \xi_0, \xi_{\text{max}}, \delta, \Delta_V \]

\[ k \leftarrow 0 \]

[1] **Poll and search step**

**Search step (optional)**

\[ x' \leftarrow \text{shaking} \left( x_k, \xi_k \right) \]

\[ x'' \leftarrow \text{descent} \left( x' \right) \]

\[ S_k \leftarrow \text{finite number of points of } M(k, \Delta_k) \text{ (possibly empty)} \]

**Poll step**

compute \( p \) MADS directions \( D_k \in \mathbb{R}^{n \times p} \)

construct the frame \( P_k \subseteq M(k, \Delta_k) \) with \( x_k, D_k \) and \( \Delta_k \)

evaluate the functions on the \( p \) points of \( P_k \)

[2] **Updates**

update of VNS amplitude \( (\xi_{k+1} \leftarrow \xi_0 \text{ or } \xi_{k+1} \leftarrow \xi_k + \delta) \)

updates of solution and mesh

\[ k \leftarrow k + 1 \]

check the stopping conditions, \textit{goto [1]}
An analytic problem with many local optima

\[
\min f(a, b) = \frac{1000 \ b \ \sin^2 b \ \sin 300a}{a} \quad \text{s.t.} \quad \begin{cases}
75 \leq a \leq 500 \\
0 \leq b \leq 10
\end{cases}
\]
Results for the analytic problem

<table>
<thead>
<tr>
<th>test</th>
<th>parameters</th>
<th>average</th>
<th>objective (f)</th>
<th>neval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LH</td>
<td>obj. (f)</td>
<td>best</td>
<td>worst</td>
</tr>
<tr>
<td></td>
<td>VNS</td>
<td>neval</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>no</td>
<td>-22.099</td>
<td>-104.419</td>
<td>-3.441</td>
</tr>
<tr>
<td>2</td>
<td>100, 10</td>
<td>-84.896</td>
<td>-105.119</td>
<td>-53.302</td>
</tr>
<tr>
<td>3</td>
<td>100, 10</td>
<td>0.1</td>
<td>-104.801</td>
<td>-102.794</td>
</tr>
<tr>
<td>4</td>
<td>100, 10</td>
<td>0.1</td>
<td>-103.304</td>
<td>-87.627</td>
</tr>
</tbody>
</table>
Results for the analytic problem

1: MADS poll only

2: + LH

3: + LH + VNS

4: + LH + VNS + DS

summary: average values

Audet, Béchard and Le Digabel

Nonsmooth Optimization by combining MADS and VNS 21/25
A MDO problem

- MDO: **MultiDisciplinary Optimization**
- Simplified aircraft model with 10 variables, 10 constraints and 3 disciplines, one for each main model component: structure, aerodynamics and propulsion
- Convergence of the model with a fixed point method through the 3 disciplines
- Surrogate obtained by relaxing the fixed point method stopping criteria (**warning**: only the number of “true” functions evaluations are counted in these tests)
## Results for the MDO problem

<table>
<thead>
<tr>
<th>test</th>
<th>parameters</th>
<th>average</th>
<th>objective (f)</th>
<th>neval</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LH  VNS</td>
<td>obj. (f)</td>
<td>best</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>neval</td>
<td>worst</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5000, 0</td>
<td>-1623.416</td>
<td>-2273.648</td>
<td>5000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>no</td>
<td>5000</td>
<td>-1315.849</td>
<td>5000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>no</td>
<td>-3101.393</td>
<td>-3964.199</td>
<td>1178</td>
<td></td>
</tr>
<tr>
<td></td>
<td>no</td>
<td>2567</td>
<td>-1588.350</td>
<td>5165</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>100, 10</td>
<td>-3443.092</td>
<td>-3964.200</td>
<td>1204</td>
<td></td>
</tr>
<tr>
<td></td>
<td>no</td>
<td>5690</td>
<td>-1355.656</td>
<td>10000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>100, 10</td>
<td>-3961.385</td>
<td>-3964.198</td>
<td>1374</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>3060</td>
<td>-3881.935</td>
<td>5806</td>
<td></td>
</tr>
</tbody>
</table>
Results for the MDO problem

1: LH alone

2: MADS poll only

3: + LH

4: + LH + VNS + SRGTE

summary: average values
Conclusion

- MADS and VNS are two complementary algorithms (MADS mesh and VNS neighborhoods, diversification when successes or failures occur), so it was natural to combine the 2
- Preliminary results show an improvement in term of quality of the solution (the random component of MADS is less critical)
- Use of surrogates and DS strategy reduce the number of function evaluations.