Nonsmooth Optimization by combining MADS and VNS

Charles Audet
Vincent Béchard
Sébastien Le Digabel
École Polytechnique de Montréal

May 2006

Presentation Outline

Presentation Outline

Presentation Outline

Presentation Outline

Presentation Outline

Presentation Outline

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 Ω are
  - nonsmooth, costly, can possibly fail to evaluate, derivative approximation is problematic
  - viewed as unexploitable black-box functions

MADS Overview

- MADS : **M**esh **A**daptive **D**irect **S**earch [Audet, Dennis]
- **NOMAD** is the c++ implemation of MADS (freely available at `www.gerad.ca/nomad`) [Couture]
- MADS generalizes the **G**eneralized **P**attern **S**earch (GPS, [Torczon]) Algorithm
- **Main convergence result** : MADS leads to a Clarke-KKT stationnary 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

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

  where $\Delta_k \in \mathbb{R}^+$ is the mesh size parameter and D a fixed set of directions in $\mathbb{R}^n$
- After each iteration, the mesh size parameter Δ_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 Δ_k^p
- The set of poll trial points (the **poll frame**) is then constructed :

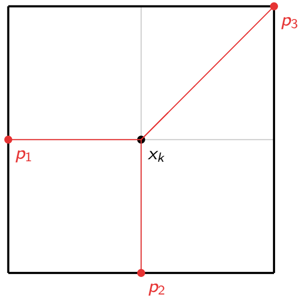
$$P_k = \left\{ x_k + \Delta_k d : d \in D_k \right\} \subseteq M(k, \Delta_k)$$

- The poll is rigidly defined (mesh update, directions used) to ensure convergence results

Poll illustration (successive fails and mesh shrink)

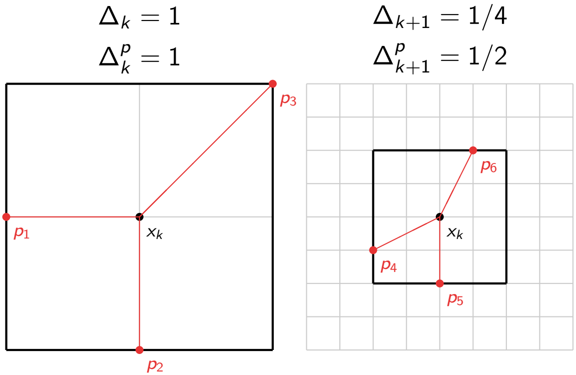
$$\Delta_k = 1$$

$$\Delta_k^p = 1$$



$$P_k = \{p_1, p_2, p_3\}$$

Poll illustration (successive fails and mesh shrink)

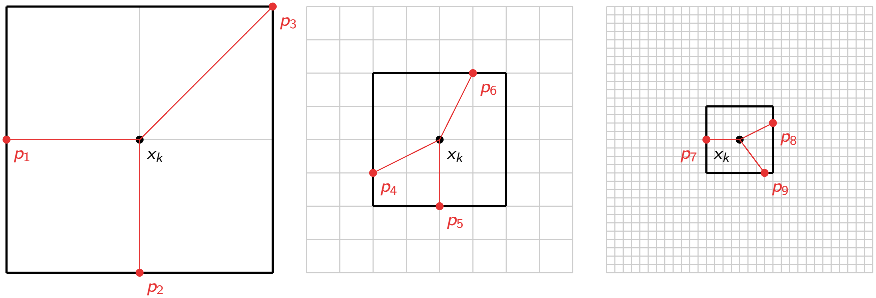


$P_k = \{p_1, p_2, p_3\}$ $\quad$ $P_{k+1} = \{p_4, p_5, p_6\}$

Poll illustration (successive fails and mesh shrink)

$$\Delta_k = 1 \qquad \Delta_{k+1} = 1/4 \qquad \Delta_{k+2} = 1/16$$

$$\Delta_k^p = 1 \qquad \Delta_{k+1}^p = 1/2 \qquad \Delta_{k+2}^p = 1/4$$



$$P_k = \{p_1, p_2, p_3\} \qquad P_{k+1} = \{p_4, p_5, p_6\} \qquad P_{k+2} = \{p_7, p_8, p_9\}$$

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 Δ_k
evaluate the functions on the p points of P_k

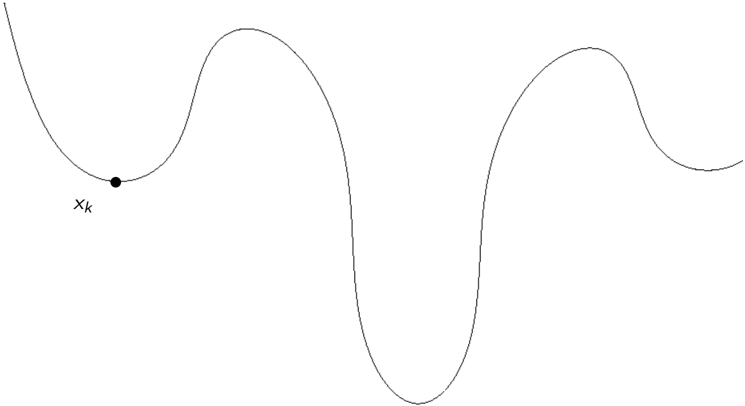
[2] Updates

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

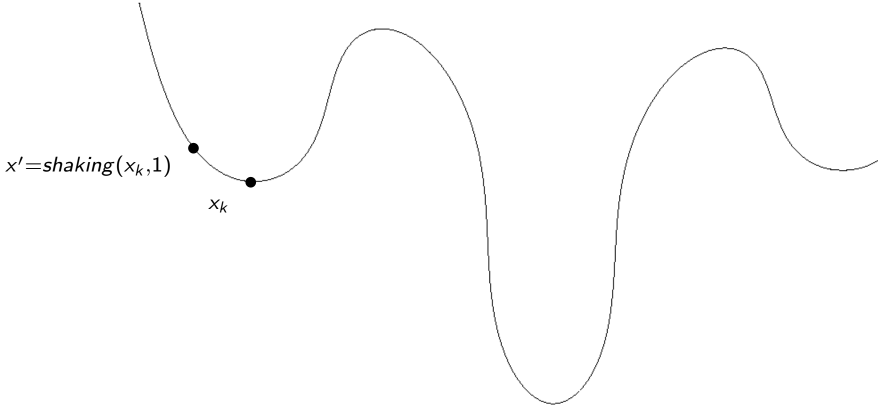
VNS Overview

- VNS : **V**ariable **N**eighborhood **S**earch [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 ξ_k and increasingly changes the current solution when ξ_k grows
- The search is more and more global when no improvements are made

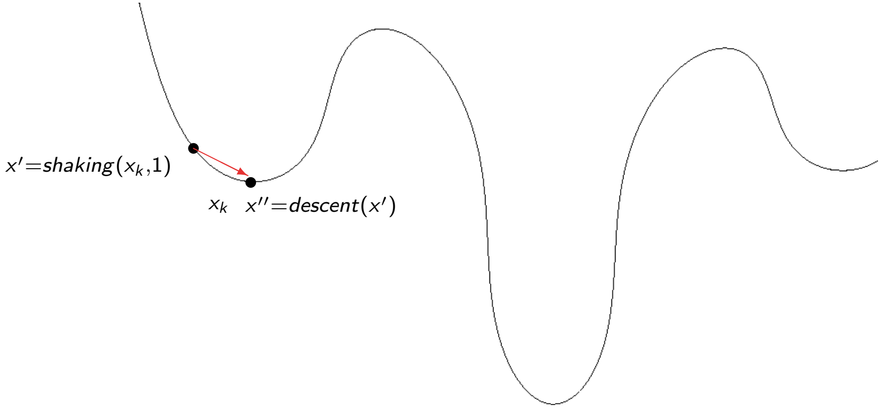
VNS illustration



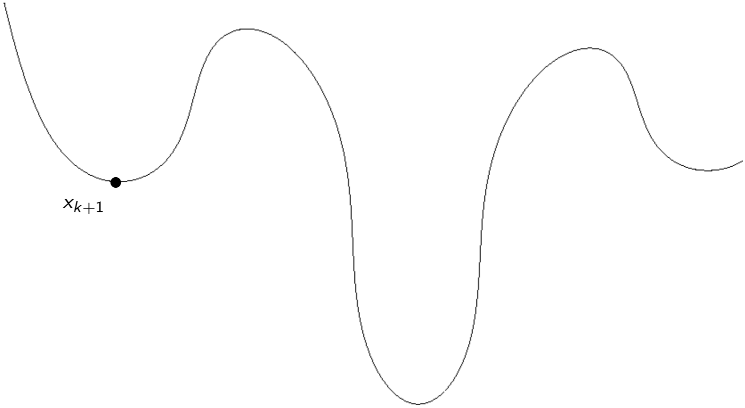
VNS illustration



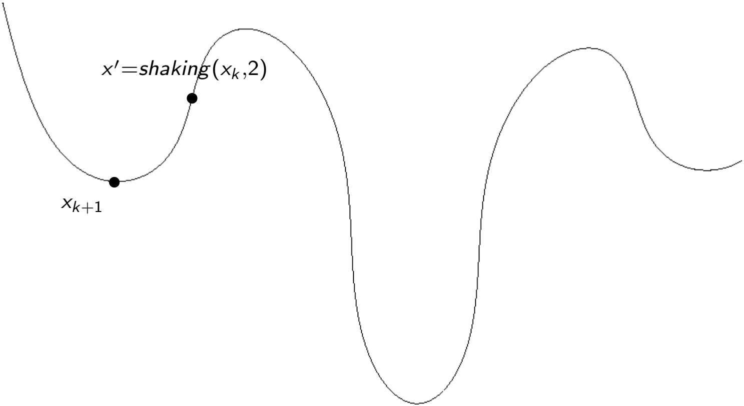
VNS illustration



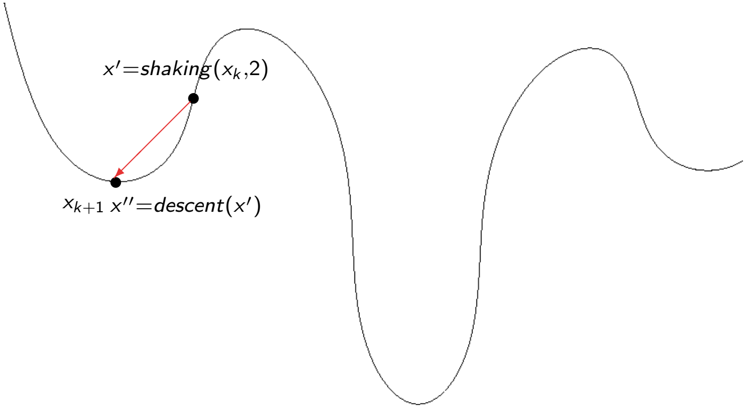
VNS illustration



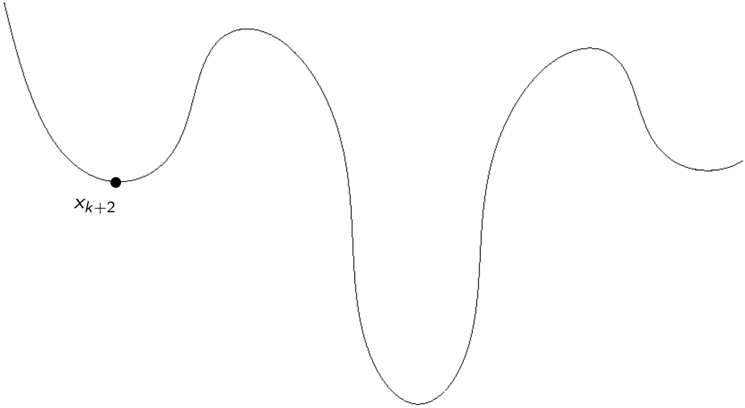
VNS illustration



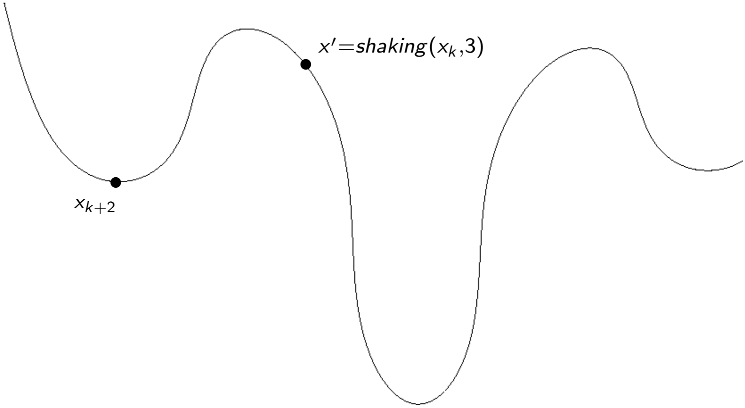
VNS illustration



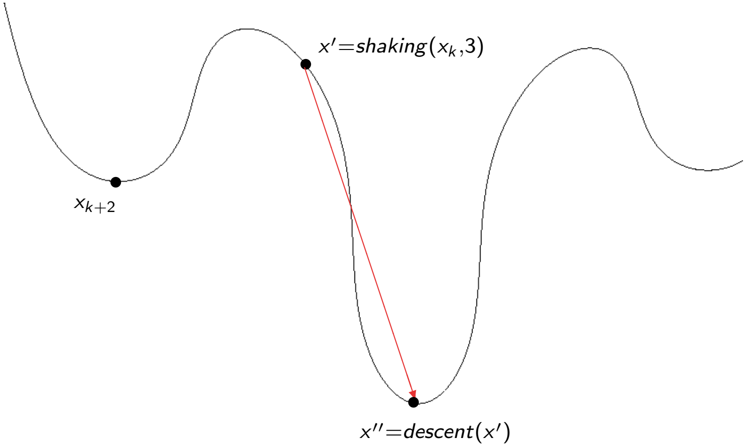
VNS illustration



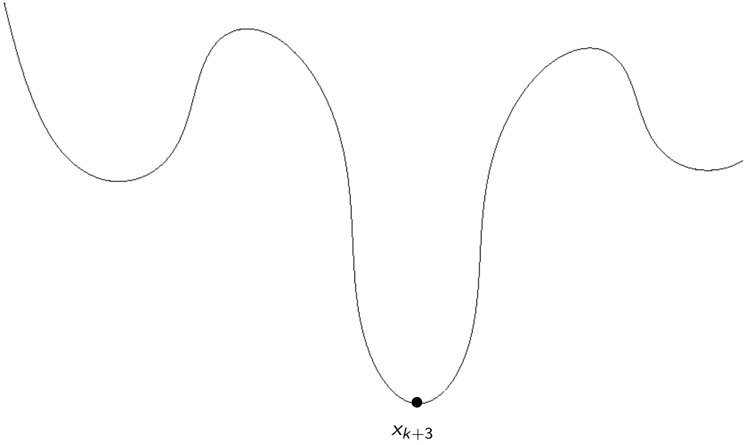
VNS illustration



VNS illustration



VNS illustration



[0] **Initializations**

$$\xi_{max}, \xi_0, \delta \in \mathbb{N}^+ \;,\; x_0 \in X$$

$$k \leftarrow 0$$

[1] **while** $(\xi_k \le \xi_{max})$

$$x' \leftarrow shaking(x_k, \xi_k)$$

$$x'' \leftarrow descent(x')$$

if $\left(f(x'') < f(x_k)\right)$

$$x_{k+1} \leftarrow x''$$

$$\xi_{k+1} \leftarrow \xi_0$$

else

$$x_{k+1} \leftarrow x_k$$

$$\xi_{k+1} \leftarrow \xi_k + \delta$$

$$k \leftarrow k + 1$$

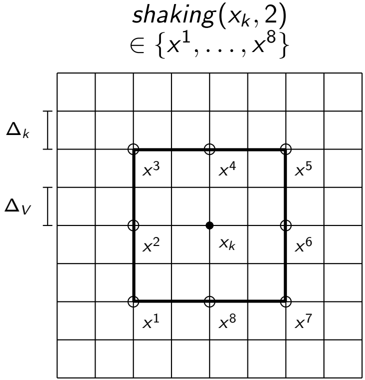
Coupling of MADS and VNS

- 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

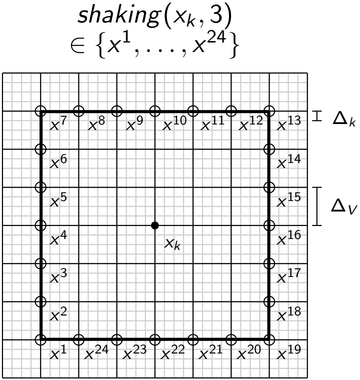
VNS shaking

- The mesh defines a natural structure for the perturbation method which can be seen as a function $shaking : (M(k, \Delta_k), \mathbb{N}) \to M(k, \Delta_V) \subseteq M(k, \Delta_k)$ and $x' \leftarrow shaking(x, \xi_k)$
- $\xi_k \in \mathbb{N}$ 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 ξ_k in order to remain independent of the current mesh size parameter Δ_k
- Δ_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)



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



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

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

$x_0 \in X$, $\Delta_0 \in \mathbb{R}^+$, ξ_0 , ξ_{max}, δ, Δ_V
$k \leftarrow 0$

[1] Poll and search step

Search step (optional)

$x' \leftarrow$ *shaking* (x_k, ξ_k) (perturb. of ampl. ξ_k)
$x'' \leftarrow$ *descent* (x') (descent on $M(k, \Delta_V) \subseteq M(k, \Delta_k)$)
$S_k \leftarrow$ finite number of points of $M(k, \Delta_k)$ (possibly empty)
evaluate the functions on $S_k \cup \{x''\}$

Poll step

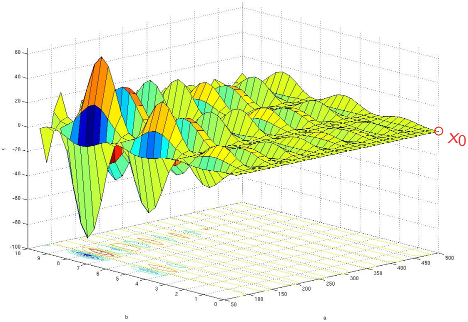
compute p MADS directions $D_k \in \mathbb{R}^{n \times p}$
construct the frame $P_k \subseteq M(k, \Delta_k)$ with x_k, D_k and Δ_k
evaluate the functions on the p points of P_k

[2] Updates

update of VNS amplitude ($\xi_{k+1} \leftarrow \xi_0$ or $\xi_{k+1} \leftarrow \xi_k + \delta$)
updates of solution and mesh
$k \leftarrow k + 1$
check the stopping conditions, **goto** [1]

An analytic problem with many local optima

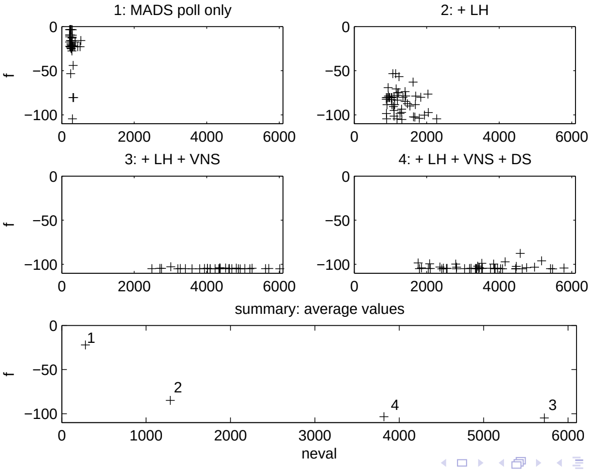
$$\min f(a,b) = \frac{1000\ b\ \sin^2 b\ \sin\ 300a}{a} \text{ s.t. } \begin{cases} 75 & \leq a \leq & 500 \\ 0 & \leq b \leq & 10 \end{cases}$$



Results for the analytic problem

test	parameters			average		objective (f)		*neval*	
	LH	VNS	DS	obj. (f)	*neval*	best	worst	best	worst
1	no	no	no	-22.099	281	-104.419	-3.441	216	528
2	100, 10	no	no	-84.896	1286	-105.119	-53.302	884	2275
3	100, 10	0.1	no	-104.801	5718	-105.119	-102.794	2485	10000
4	100, 10	0.1	10^{-4}	-103.304	3818	-105.119	-87.627	1764	8065

Results for the analytic problem



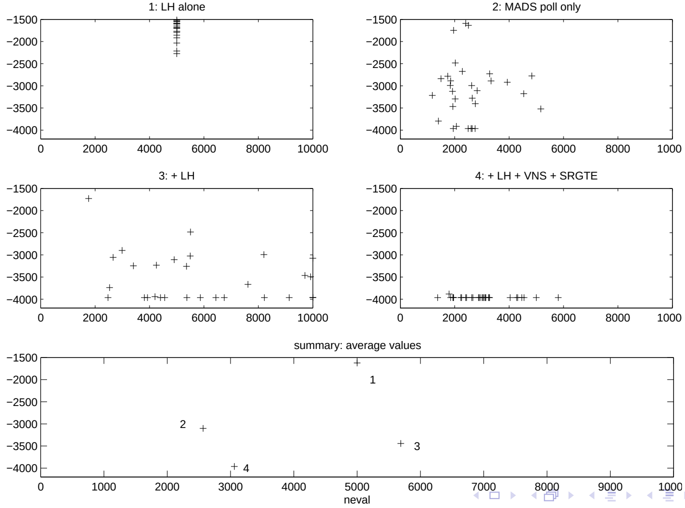
A MDO problem

- MDO : **M**ulti**D**isciplinary **O**ptimization
- 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

test	parameters		average		objective (f)		*neval*	
	LH	VNS	obj. (f)	*neval*	best	worst	best	worst
1	5000, 0	no	-1623.416	5000	-2273.648	-1315.849	5000	5000
2	no	no	-3101.393	2567	-3964.199	-1588.350	1178	5165
3	100, 10	no	-3443.092	5690	-3964.200	-1355.656	1204	10000
4	100, 10	0.1	-3961.385	3060	-3964.198	-3881.935	1374	5806

Results for the MDO problem



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.