

Extensions

MTH8418

S. Le Digabel, Polytechnique Montréal

Winter 2020

(v2)

Plan

Global optimization

Parallelism

Discrete variables

Other extensions

References

Global optimization

Parallelism

Discrete variables

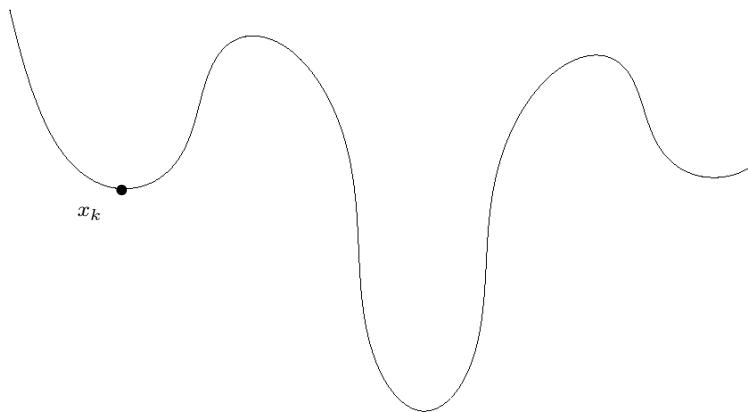
Other extensions

References

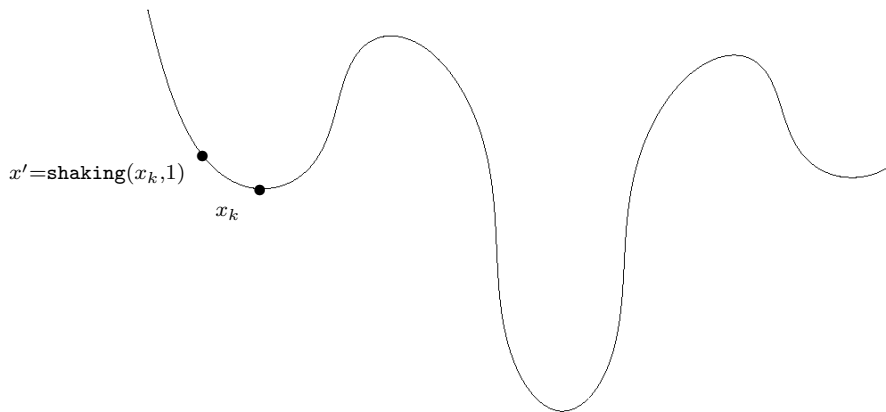
MADS+VNS: VNS overview

- ▶ MADS+VNS [**Audet et al., 2008a**]
- ▶ VNS : **V**ariable **N**eighborhood **S**earch [**Mladenović and Hansen, 1997**]
- ▶ 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

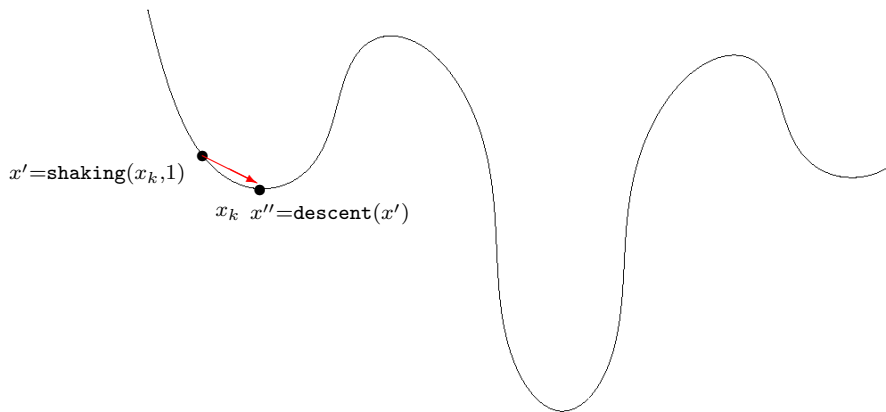
VNS illustration



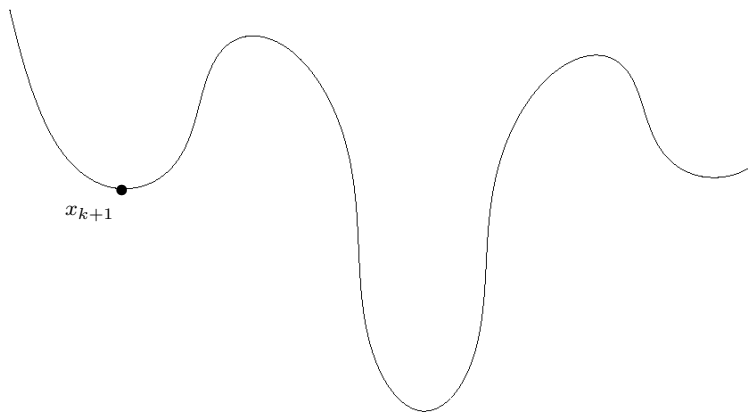
VNS illustration



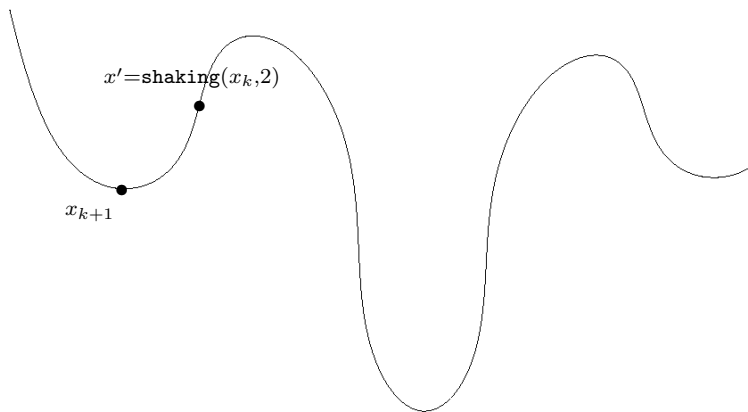
VNS illustration



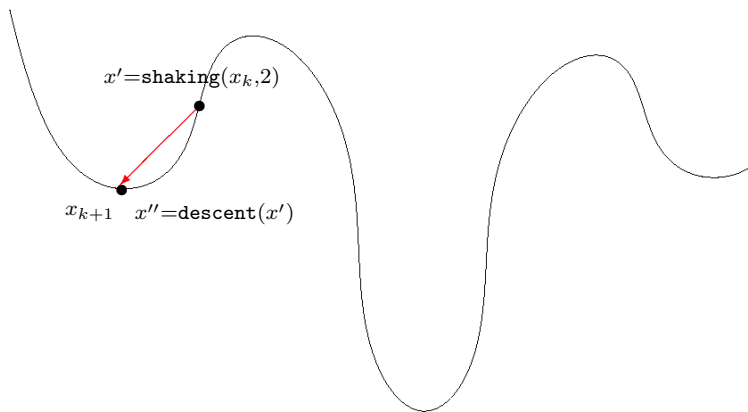
VNS illustration



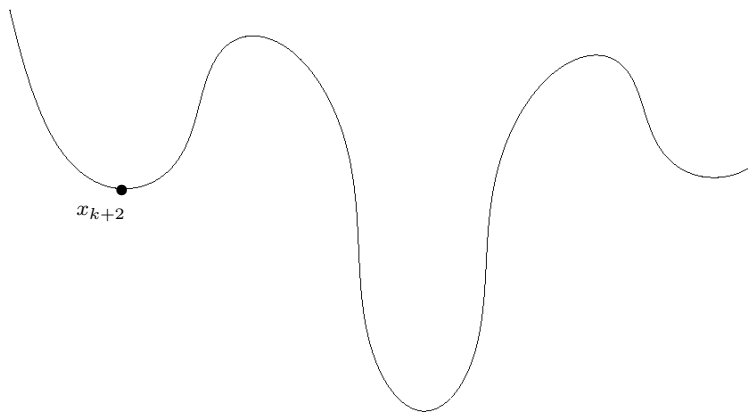
VNS illustration



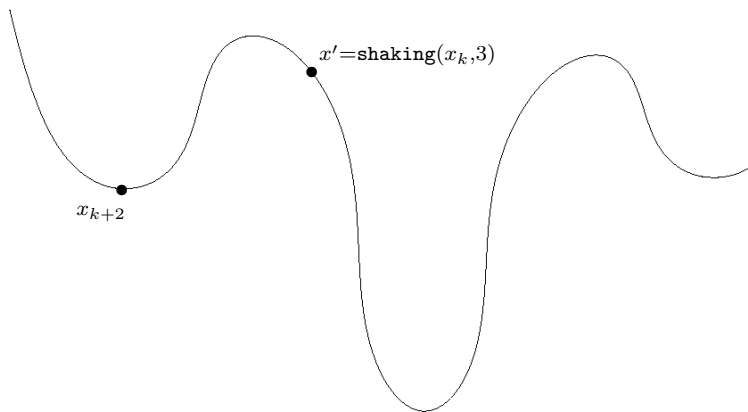
VNS illustration



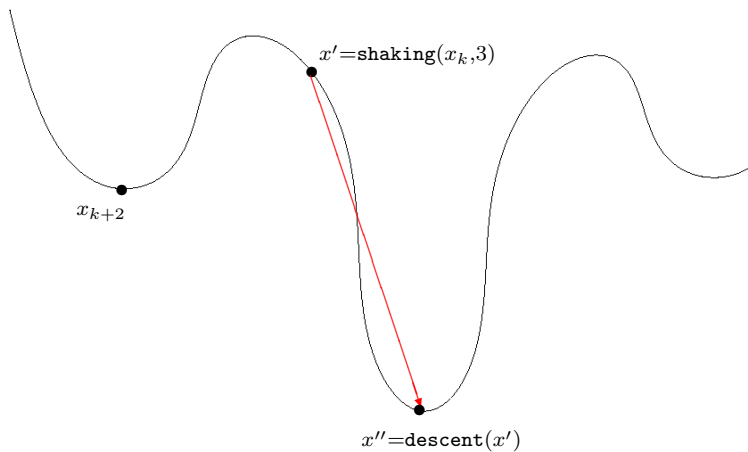
VNS illustration



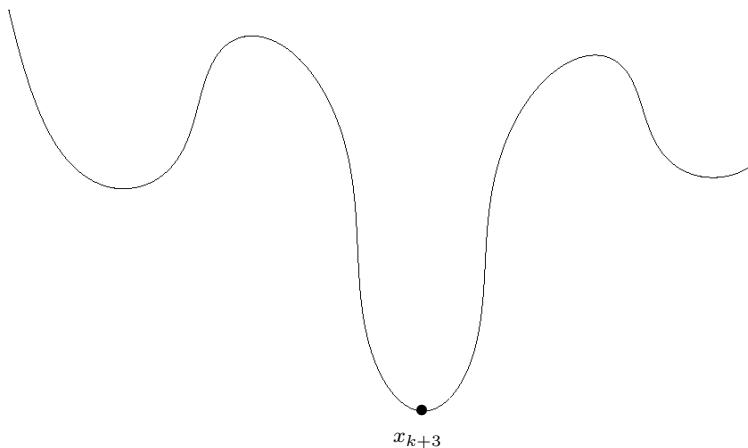
VNS illustration



VNS illustration



VNS illustration



VNS algorithm

[0] Initializations

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

$$k \leftarrow 0$$

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

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

$$x'' \leftarrow \text{descent}(x')$$

if ($f(x'') < f(x_k)$)

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

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

else

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

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

$$k \leftarrow k + 1$$

Coupling of MADS and VNS

- ▶ Incorporation of VNS into the search step of MADS
- ▶ Mesh redefined with a variable mesh size: $M(\Delta)$
- ▶ 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

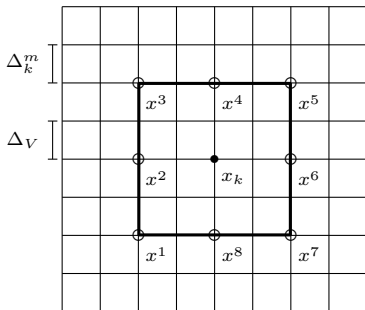
VNS shaking

- ▶ The mesh defines a natural structure for the perturbation method which can be seen as a function
 $\text{shaking} : (M(\Delta_k^m), \mathbb{N}) \rightarrow M(\Delta_V) \subseteq M(\Delta_k^m)$ and
 $x' \leftarrow \text{shaking}(x, \xi_k)$
- ▶ $\xi_k \in \mathbb{N}$ is the **perturbation amplitude**
- ▶ The fixed-size mesh $M(\Delta_V) \subseteq M(\Delta_k^m)$ 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^m
- ▶ Δ_V is called the **VNS trigger** (VNS search only occurs at iteration k when $\Delta_k^m \leq \Delta_V$ and $\Delta_V = \ell \Delta_k^m$ for some $\ell \in \mathbb{N}$)
- ▶ If $D = [I \ -I]^\top$, the perturbed point x' can be chosen so that $\|x_k - x'\|_\infty = \xi_k \Delta_V$

Examples of shaking

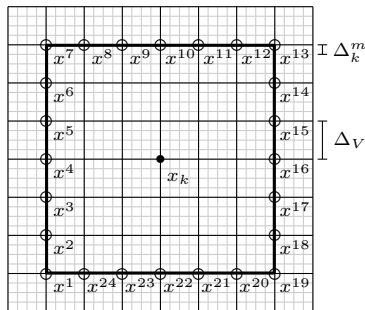
Examples of meshes $M(\Delta_k^m)$ (gray), $M(\Delta_V)$ (black) and possible choices for the perturbation (points x^i on the bold frame at distance $\xi_k \Delta_V$ of x_k)

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



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

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



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

VNS descent

- ▶ Function descent : $M(\Delta_V) \rightarrow M(\Delta_k^m)$ and $x'' \leftarrow \text{descent}(x')$
- ▶ Use of a specific poll step, with its own mesh size parameter and its own filter for the constraints
- ▶ Cannot reduce the current mesh size
- ▶ Use a static surrogate if available

MADS+VNS detailed algorithm

[0] Initializations

$x_0 \in \mathcal{X}, \Delta_0^m \in \mathbb{R}^+, \xi_0, \xi_{max}, \delta, \Delta_V$
 $k \leftarrow 0$

[1] Poll and search steps

Search step

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

Poll step

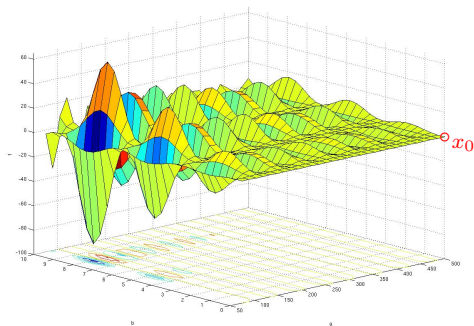
compute p MADS directions $D_k \in \mathbb{R}^{n \times p}$
 construct the frame $P_k \subseteq M(\Delta_k^m)$ with x_k, D_k and Δ_k^m
 evaluate the functions on the p points of P_k

[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 size
 $k \leftarrow k + 1$
 check the stopping conditions, **goto** [1]

An analytic problem with many local optima

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



Results for the analytic problem

Each test consists of 30 executions

| test | parameters | | average | | objective (f) | | <i>neval</i> | |
|------|------------|-----|--------------|--------------|-------------------|----------|--------------|-------|
| | LHS | VNS | obj. (f) | <i>neval</i> | best | worst | best | worst |
| 1 | no | no | -22.099 | 281 | -104.419 | -3.441 | 216 | 528 |
| 2 | 100, 10 | no | -84.896 | 1286 | -105.119 | -53.302 | 884 | 2275 |
| 3 | 100, 10 | 0.1 | -104.801 | 5718 | -105.119 | -102.794 | 2485 | 10000 |

Results for a MDO problem

Each test consists of 30 executions

| test | parameters | | average | | objective (f) | | <i>neval</i> | |
|------|------------|-----|--------------|--------------|-------------------|-----------|--------------|-------|
| | LHS | VNS | obj. (f) | <i>neval</i> | 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 |

Global optimization

Parallelism

Discrete variables

Other extensions

References

pMADS

- ▶ Idea: simply evaluate the trial points in parallel
- ▶ Synchronous version **pMADS-S**:
 - ▶ The iteration is over only when all the evaluations in progress are terminated
 - ▶ Processes can be idle between two evaluations
 - ▶ The algorithm is identical to the scalar version (without opportunism)
- ▶ Asynchronous version **pMADS-A**:
 - ▶ If a new best point is found, the iteration is terminated even if there are evaluations in progress. New trial points are then generated
 - ▶ Processes never wait between two evaluations
 - ▶ “Old” evaluations are considered when they are finished
 - ▶ The algorithm is slightly reorganized

PSD-MADS

- ▶ **PSD: P**arallel **S**pace **D**ecomposition [**Audet et al., 2008b**]
- ▶ Idea: each process executes a MADS algorithm on a subproblem and has responsibility of small groups of variables
- ▶ Based on the block-Jacobi method [**Bertsekas and Tsitsiklis, 1989**] and on the Parallel Variable Distribution [**Ferris and Mangasarian, 1994**]
- ▶ Objective: solve larger problems ($\simeq 50 - 500$ instead of $\simeq 10 - 20$)
- ▶ Asynchronous method
- ▶ Convergence analysis

PSD-MADS: processes

▶ Master

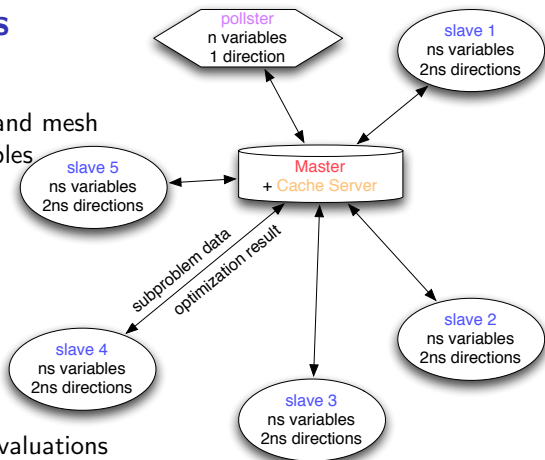
- ▶ Receives all slave signals
- ▶ Updates current solution and mesh
- ▶ Decides subproblem variables
- ▶ Sends subproblem data

▶ Slaves

- ▶ Receive subproblem data
- ▶ Optimize subproblem
- ▶ Send optimization data

▶ Cache server

- ▶ Memorizes all black-box evaluations
- ▶ Allows the “cache search” in the pollster

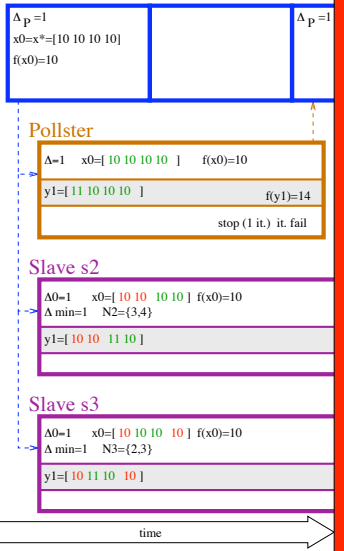


Master

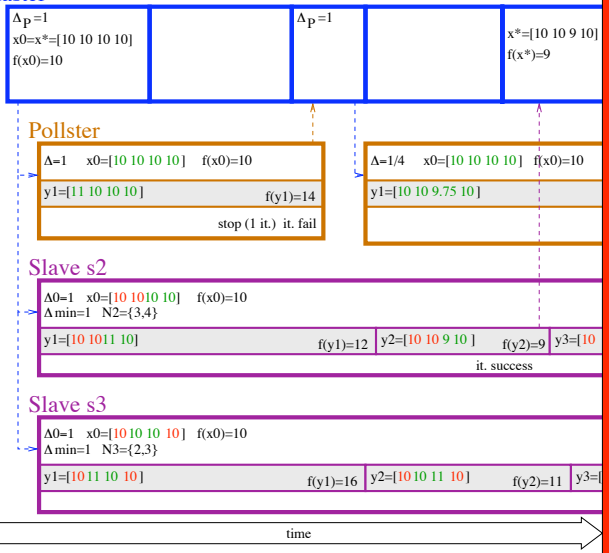
```
Δp = 1
x0 = x* = [10 10 10 10]
f(x0) = 10
```

time

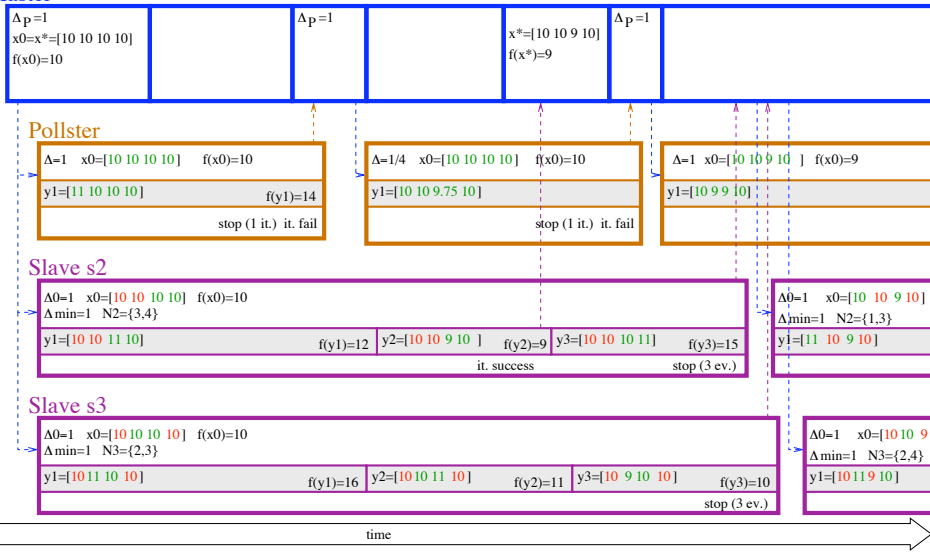
Master



Master



Master



New PSD-MADS

- ▶ PSD-MADS from 2008: Groups of variables are randomly decided
- ▶ New version: [Alarie et al., 2018]
- ▶ Identify natural clusters as the groups of variables. k -means clustering is used on a matrix of sensitivities
- ▶ Larger problems can be solved efficiently: Up to 4000 variables so far
- ▶ Future work: Use PCA in order to construct the groups

COOP-MADS

- ▶ Uses a simplified version of the PSD-MADS parallel framework
- ▶ Processes run in parallel on the original problem with different seeds in order to produce different behaviours
- ▶ The cache server allows to share evaluations and the cache search is performed by all processes
- ▶ Asynchronous method

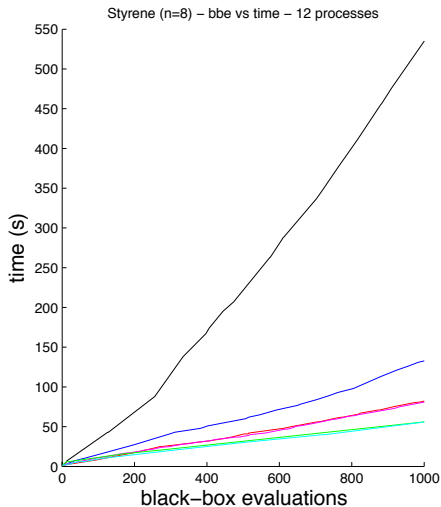
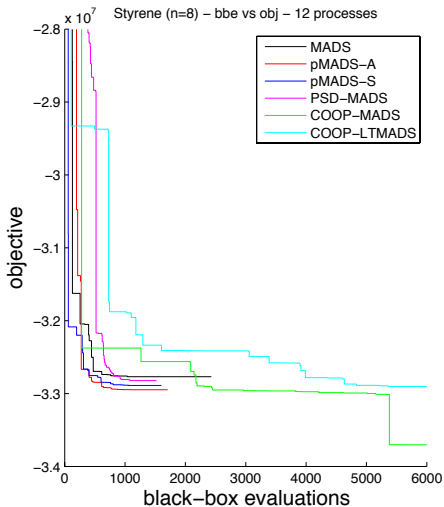
Computational tests

- ▶ NOMAD version 3.4, May 2010
- ▶ MADS, pMADS-A/S, COOP-MADS and PSD-MADS (2008 version)
- ▶ Parallelism with MPI
- ▶ Computer: 6 cores with Hyper-Threading \simeq 12 processors

Algorithms parameters

- ▶ MADS and pMADS-A/S: default parameters, $np=13$ (1 master+12 slaves)
- ▶ PSD-MADS with the SIOPT paper settings:
 - ▶ $np=14$: 1 master + 1 cache server + 1 pollster slave + 11 regular slaves
 - ▶ Max. number of evaluations for each regular slave: 10
 - ▶ Number of free variables for each regular slave: 2
 - ▶ Groups of variables: randomly chosen
- ▶ COOP-MADS, $np=13$ (1 cache server+12 workers)
 - ▶ COOP-MADS: each process uses OrthoMADS polling directions with different Halton seeds
 - ▶ COOP-LTMADS: LTMADS with different random seeds
- ▶ Stopping criteria: maximum number of blackbox evaluations or a minimal mesh size

Styrene, 6000 evaluations



Test problem G2

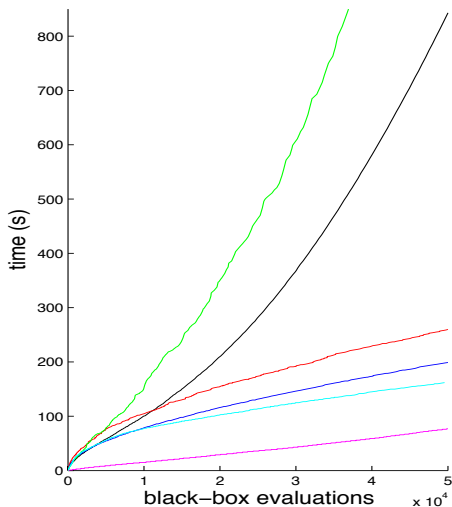
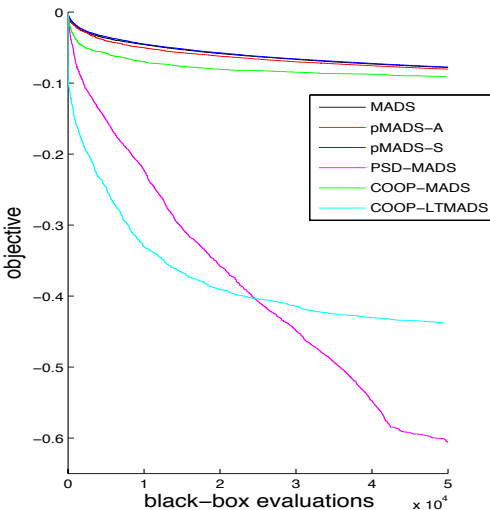
Taken from [Hedar and Fukushima, 2006]

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

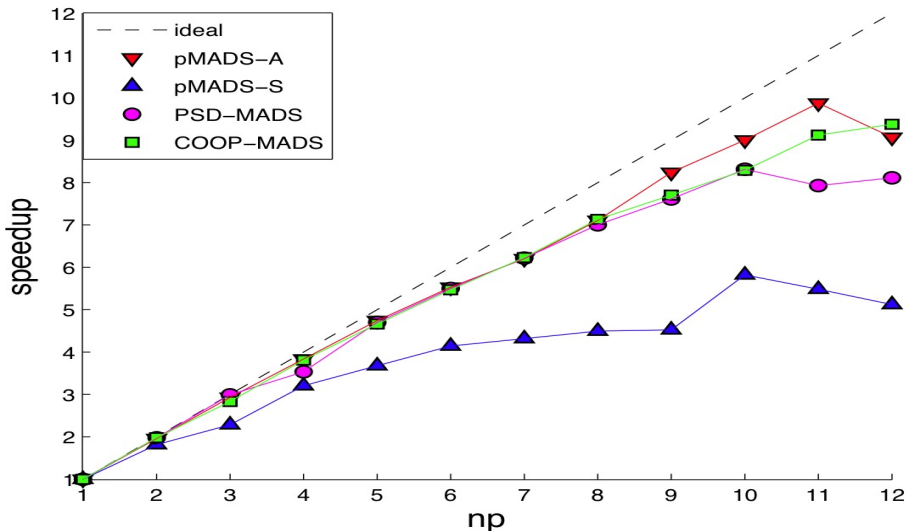
$$s.t. \begin{cases} g_1(x) = -\prod_{i=1}^n x_i + 0.75 \leq 0 \\ g_2(x) = \sum_{i=1}^n x_i - 7.5n \leq 0 \end{cases}$$

$$n = 500, 0 \leq x \leq 10, x_0 = [5 \ 5 \ \dots \ 5]^\top$$

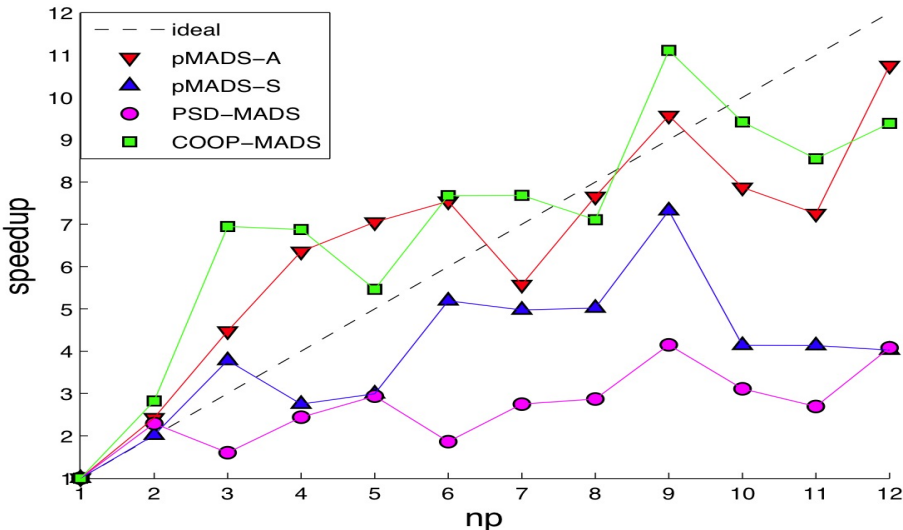
Problem G2, 50,000 evaluations



Speedup for 100 evaluations of G2, $n=10$



Speedup for 1000 evaluations on Styrene



Results analysis

- ▶ PSD-MADS is much more efficient on the large problem
- ▶ COOP-MADS gives the best result on the Styrene problem, and COOP-MADS is better than COOP-LTMADS
- ▶ The bad behaviour of COOP-MADS on the large problem is due to the fact that processes never go to small meshes where OrthoMADS directions are different even for different Halton seeds: all processes are evaluating almost exactly the same points and the cache server makes them wait
- ▶ MADS and pMADS seem equivalent. In fact, many other tests suggest that MADS gives better solutions than pMADS. This is due to the **opportunistic strategy** that the scalar version exploits better

Global optimization

Parallelism

Discrete variables

Other extensions

References

Classification of variables

- ▶ Continuous \mathbb{R}
- ▶ Integer \mathbb{Z} (orderable) and granular
- ▶ Binary $\{0, 1\}$
- ▶ Categorical

Classification of variables

- ▶ Continuous \mathbb{R}
Requires no special attention
- ▶ Integer \mathbb{Z} (orderable) and granular
- ▶ Binary $\{0, 1\}$
- ▶ Categorical

Classification of variables

- ▶ Continuous \mathbb{R}
Requires no special attention
- ▶ Integer \mathbb{Z} (orderable) and granular
GMESH strategy [Audet et al., 2019b]: Redefinition of MADS
- ▶ Binary $\{0, 1\}$
- ▶ Categorical

Classification of variables

- ▶ Continuous \mathbb{R}
Requires no special attention
- ▶ Integer \mathbb{Z} (orderable) and granular
GMESH strategy [Audet et al., 2019b]: Redefinition of MADS
- ▶ Binary $\{0, 1\}$
An integer variable in $[0; 1]$
- ▶ Categorical

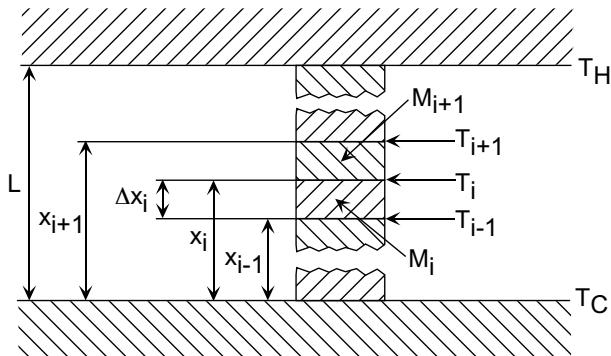
Classification of variables

- ▶ Continuous \mathbb{R}
Requires no special attention
- ▶ Integer \mathbb{Z} (orderable) and granular
GMESH strategy [Audet et al., 2019b]: Redefinition of MADS
- ▶ Binary $\{0, 1\}$
An integer variable in $[0; 1]$
- ▶ Categorical
Non-orderable integer variable
Unrelaxable discrete variable

Categorical variables: Example 1

- ▶ Buy pump from manufacturer A or B. Pump A has 3 parameters. Pump B has only 1.
- ▶ The number of variables changes depending on the value of the categorical variables.
- ▶ The idea is that the user provides the poll set associated to the categorical variables (the neighborhood structure).
- ▶ We still need to work on the integration with the poll step.

Example 2: A thermal insulation system



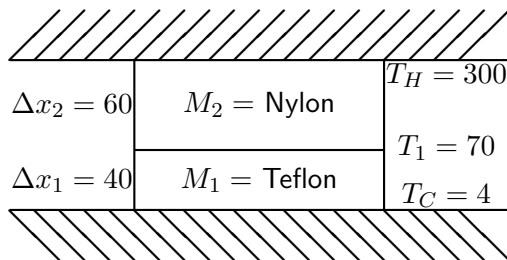
$$\begin{aligned}
 & \min_{\Delta \mathbf{x}, \mathbf{T}, \mathbf{n}, \mathbf{M}} && \text{power}(\Delta \mathbf{x}, \mathbf{T}, \mathbf{n}, \mathbf{M}) \\
 & \text{s.t.} && \Delta \mathbf{x} \geq \mathbf{0}, \quad T_C \leq \mathbf{T} \leq T_H, \\
 & && \mathbf{n} \in \mathbb{N}, \quad \mathbf{M} \in \text{Materials}
 \end{aligned}$$

MADS with categorical variables

- ▶ **[Abramson et al., 2009]**
- ▶ The search is still a finite search on the mesh, free of any rules
- ▶ The poll is the failsafe step that evaluates function values at mesh neighbors for the continuous variables, and in a user-defined set of neighbors $\mathcal{N}(x_k)$
- ▶ This set of neighbors defines a notion of *local optimality*

Illustration of a set of neighbors

Heat shield example:



Mesh neighbors: Vary either Δx_1 or Δx_2 or T_1 by $\pm \Delta_k^m$

Set of neighbors \mathcal{N} :

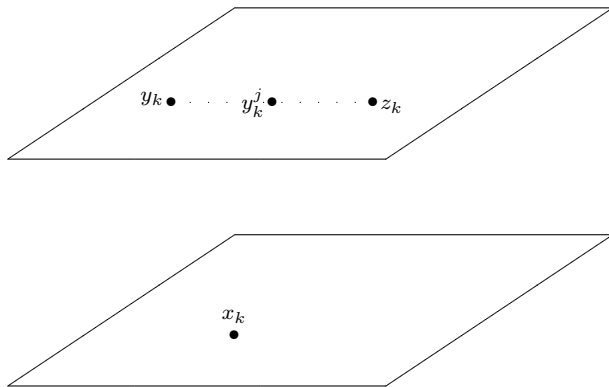
- Change M_1 to Nylon or to Fiberglass
- Change M_2 to Teflon or to Fiberglass
- Add a shield and a material
- Remove a shield and a material

MADS with categorical variables

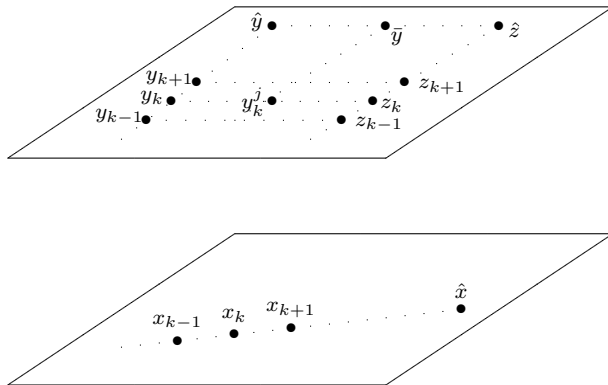
Given Δ_0^m , $x_0 \in M(\Delta_0^m)$ for $k = 0, 1, \dots$, do

1. Search on the mesh $M(\Delta_k^m)$
2. Poll on the set of mesh neighbors and on the set of neighbors $\mathcal{N}(x_k)$
3. **Extended poll step:** If $f(y)$ is within ξ of $f(x_k)$ for some y in $\mathcal{N}(x_k)$, then fix the discrete variables at y^d and poll until a local mesh minimizer is found or until a solution whose objective function value is less than $f(x_k)$ is found

Extended poll



Extended poll



User-controlled “knobs”

- ▶ Search: Use heuristics, knowledge of the problem, surrogates of functions, interpolation models, etc.
- ▶ Set of neighbors \mathcal{N} : Use information about the problem to define it
- ▶ Extended polling trigger ξ : Key control in finding a good local solution

Global optimization

Parallelism

Discrete variables

Other extensions

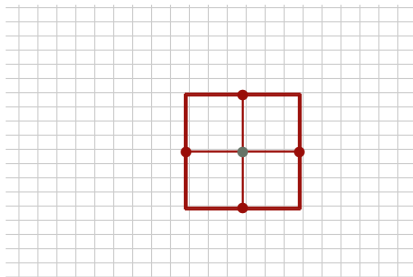
References

Dynamic scaling

- ▶ Before 2014: All MADS methods used an **isotropic** mesh (i.e. one mesh size parameter)
- ▶ Initial scaling was crucial, since weights for each variable are fixed

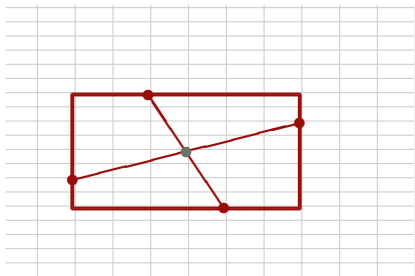
Dynamic scaling

- ▶ Before 2014: All MADS methods used an **isotropic** mesh (i.e. one mesh size parameter)
- ▶ Initial scaling was crucial, since weights for each variable are fixed
- ▶ New strategy: **Anisotropic mesh**:



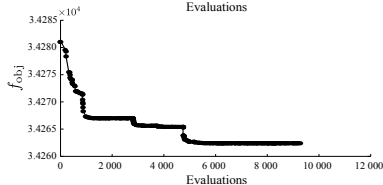
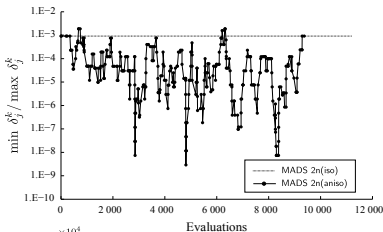
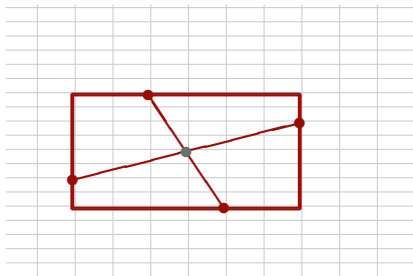
Dynamic scaling

- ▶ Before 2014: All MADS methods used an **isotropic** mesh (i.e. one mesh size parameter)
- ▶ Initial scaling was crucial, since weights for each variable are fixed
- ▶ New strategy: **Anisotropic mesh**:



Dynamic scaling

- ▶ Before 2014: All MADS methods used an **isotropic** mesh (i.e. one mesh size parameter)
- ▶ Initial scaling was crucial, since weights for each variable are fixed
- ▶ New strategy: **Anisotropic mesh**:



Recent research

- ▶ MADS + Nelder-Mead [Audet and Tribes, 2018]
- ▶ Robust optimization:
 - ▶ Reduce Implementation errors [Audet et al., 2018]
 - ▶ Stochastic blackboxes [Audet et al., 2019a]
- ▶ Gray boxes [Audet et al., 2020]
- ▶ Variable precision surrogates [Alarie et al., 2019]

Other topics

- ▶ Find multiple local optima [Larson and Wild, 2016]
- ▶ Important variables
- ▶ Exploit structure (gray boxes or other)
- ▶ etc.

Global optimization

Parallelism

Discrete variables

Other extensions

References

References I

Abramson, M. (2004). Mixed variable optimization of a Load-Bearing thermal insulation system using a filter pattern search algorithm. *Optimization and Engineering*, 5(2):157–177.

Abramson, M., Audet, C., Chrissis, J., and Walston, J. (2009). Mesh Adaptive Direct Search Algorithms for Mixed Variable Optimization. *Optimization Letters*, 3(1):35–47.

Alarie, S., Amaioua, N., Audet, C., Le Digabel, S., and Leclaire, L.-A. (2018). Selection of variables in parallel space decomposition for the mesh adaptive direct search algorithm. Technical Report G-2018-38, Les cahiers du GERAD.

Alarie, S., Audet, C., Bouchet, P.-Y., and Digabel, S. L. (2019). Optimization of noisy blackboxes with adaptive precision. Technical Report G-2019-84, Les cahiers du GERAD.

Audet, C., Béchard, V., and Le Digabel, S. (2008a). Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search. *Journal of Global Optimization*, 41(2):299–318.

Audet, C., Côté, P., Poissant, C., and Tribes, C. (2020). Monotonic grey box optimization. *Optimization Letters*, 14(1):3–18.

Audet, C., Dennis, Jr., J., and Le Digabel, S. (2008b). Parallel Space Decomposition of the Mesh Adaptive Direct Search Algorithm. *SIAM Journal on Optimization*, 19(3):1150–1170.

Audet, C., Dzahini, K., Kokkolaras, M., and Le Digabel, S. (2019a). StoMADS: Stochastic blackbox optimization using probabilistic estimates. Technical Report G-2019-30, Les cahiers du GERAD.

Audet, C., Ihaddadene, A., Le Digabel, S., and Tribes, C. (2018). Robust optimization of noisy blackbox problems using the Mesh Adaptive Direct Search algorithm. *Optimization Letters*, 12(4):675–689.

References II

- Audet, C., Le Digabel, S., and Tribes, C. (2016). Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization. *Optimization and Engineering*, 17(2):333–358.
- Audet, C., Le Digabel, S., and Tribes, C. (2019b). The Mesh Adaptive Direct Search Algorithm for Granular and Discrete Variables. *SIAM Journal on Optimization*, 29(2):1164–1189.
- Audet, C. and Tribes, C. (2018). Mesh-based Nelder-Mead algorithm for inequality constrained optimization. *Computational Optimization and Applications*, 71(2):331–352.
- Bertsekas, D. and Tsitsiklis, J. (1989). *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Ferris, M. and Mangasarian, O. (1994). Parallel variable distribution. *SIAM Journal on Optimization*, 4(4):815–832.
- Hedar, A.-R. and Fukushima, M. (2006). Tabu search directed by direct search methods for nonlinear global optimization. *European Journal of Operational Research*, 170(2):329–349.
- Larson, J. and Wild, S. (2016). A batch, derivative-free algorithm for finding multiple local minima. *Optimization and Engineering*, 17(1):205–228.
- Le Digabel, S., Abramson, M., Audet, C., and Dennis, Jr., J. (2010). Parallel Versions of the MADS Algorithm for Black-Box Optimization. In *Optimization days, Montreal*. GERAD. Slides available at https://www.gerad.ca/Sebastien.Le.Digabel/talks/2010_JOPT_25mins.pdf.
- Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research*, 24(11):1097–1100.