JOPT 2011

Use of Models with the MADS Algorithm for Blackbox Optimization

Sébastien Le Digabel, École Polytechnique de Montréal

2011–05–02

Collaborators:

A.R. Conn (quadratic models)

R.B. Gramacy (TGP surrogates)

Presentation outline

Blackbox optimization problems

Presentation outline

Blackbox optimization problems

The MADS algorithm

Presentation outline

Blackbox optimization problems

The MADS algorithm

MADS and models

Presentation outline

Blackbox optimization problems

The MADS algorithm

MADS and models

Discussion

Blackbox optimization problems

The MADS algorithm

MADS and models

Discussion

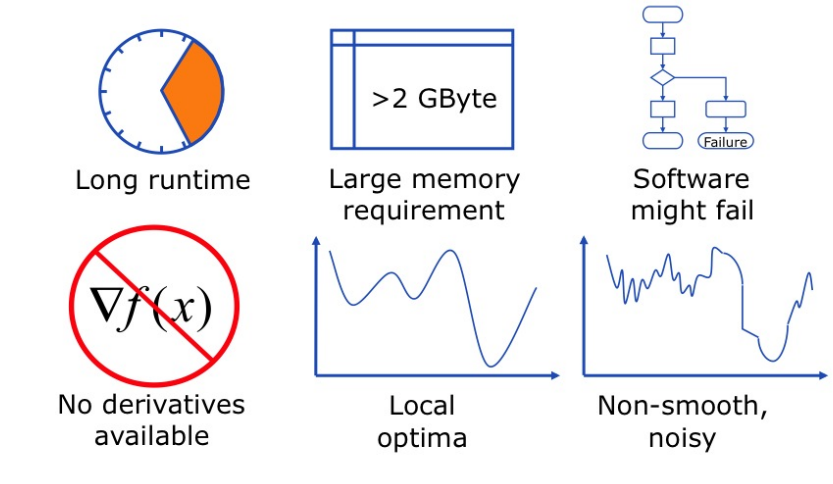
Blackbox optimization problems

We consider the optimization problem:

$$\min_{x \in \Omega} \quad f(x)$$

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

Blackboxes as illustrated by J. Simonis [ISMP 2009]



Long runtime

Large memory requirement

Software might fail

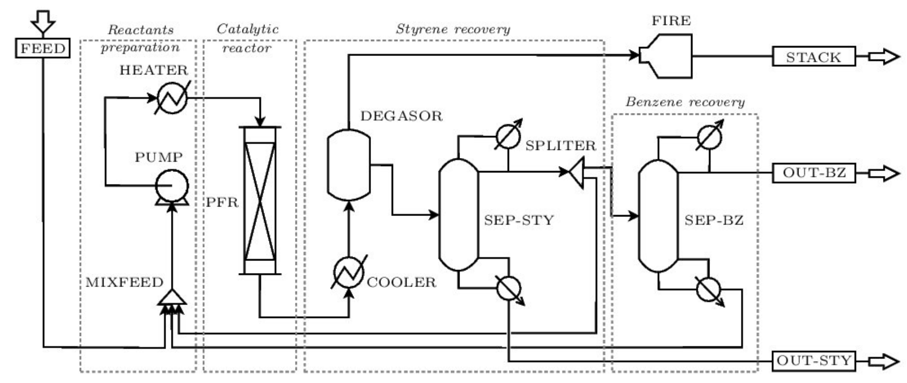
No derivatives available

Local optima

Non-smooth, noisy

Example of an engineering problem

STYRENE problem [Audet, Béchard, Le Digabel, JOGO 2008]



8 variables, 11 constraints, one evaluation $\simeq$ 3s, $\simeq$20% of failures.

Blackbox optimization problems

The MADS algorithm

MADS and models

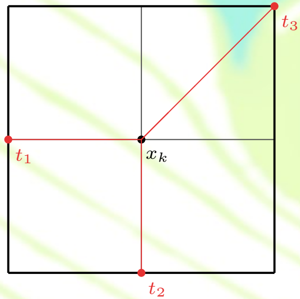
Discussion

Mesh Adaptive Direct Search (MADS)

- Audet and Dennis [SIOPT, 2006]
- Iterative algorithm that evaluates the blackbox at some trial points on a spatial discretization called the mesh.
- One iteration = search and poll.
- The search allows trial points generated anywhere on the mesh.
- The poll consists in generating a list of trial points constructed from poll directions. These directions grow dense.
- At the end of the iteration, the mesh size is reduced if no new iterate is found.
- Algorithm is backed by a convergence analysis based on the Clarke Calculus for nonsmooth functions.

Poll illustration (successive fails and mesh shrink)

$$\Delta_k^m = \Delta_k^p = 1$$



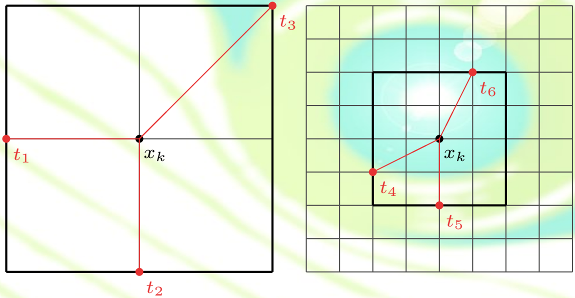
poll trial points=$\{t_1, t_2, t_3\}$

Poll illustration (successive fails and mesh shrink)

$\Delta_k^m = \Delta_k^p = 1$

$\Delta_{k+1}^m = 1/4$

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



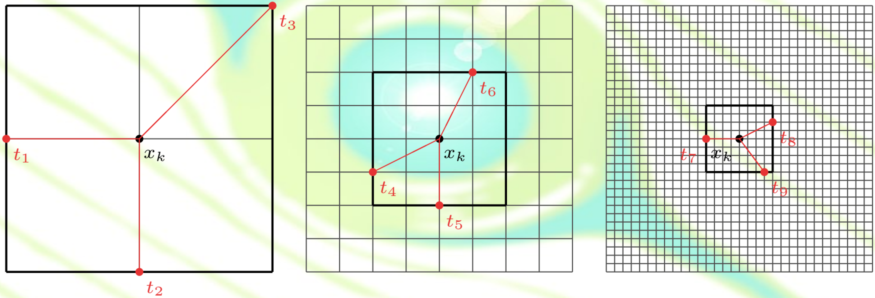
poll trial points=$\{t_1, t_2, t_3\}$ $= \{t_4, t_5, t_6\}$

Poll illustration (successive fails and mesh shrink)

$\Delta_k^m = \Delta_k^p = 1$

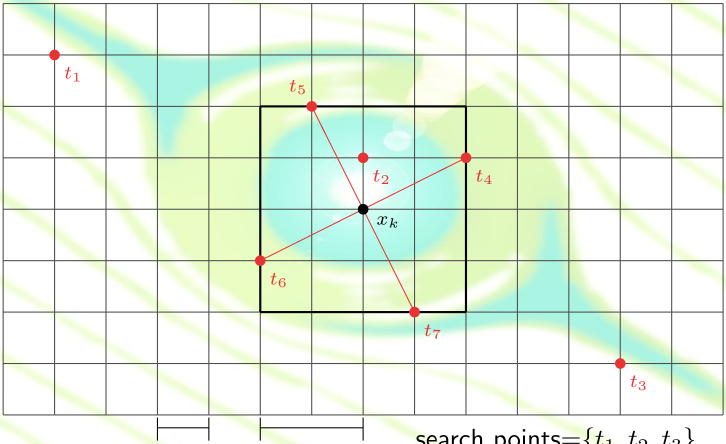
$\Delta_{k+1}^m = 1/4$
$\Delta_{k+1}^p = 1/2$

$\Delta_{k+2}^m = 1/16$
$\Delta_{k+2}^p = 1/4$



poll trial points$=\{t_1, t_2, t_3\}$ $=\{t_4, t_5, t_6\}$ $=\{t_7, t_8, t_9\}$

Poll and Search trial points



$\Delta_k^m = \frac{1}{4}$ $\Delta_k^p = \frac{1}{2}$

search points=$\{t_1, t_2, t_3\}$

poll points=$\{t_4, t_5, t_6, t_7\}$

Blackbox optimization problems

The MADS algorithm

MADS and models

Discussion

General framework

Only the additions to the MADS algorithm are reported.

```
[0] Initializations
[1] Iteration $k$
        [1.1] Model Search
                select data points from cache
                construct one model for each output (obj + cstrs)
                select points for model improvement
                optimize model to determine oracle points
                project candidates to the mesh
                evaluate candidates opportunistically
        [1.2] Poll (if the Search failed)
                Model Ordering: use models to sort trial points
[2] Updates
```

Constraints handling (Filter-type method)

- Constraint set $\Omega = \{x \in R : c_j(x) \leq 0, j \in J = \{1, \ldots, m\}\}$.
- Constraint violation function:

$$h(x) = \begin{cases} \sum_{j \in J} (\max(c_j(x), 0))^2 & \text{if } x \in R, \\ \infty & \text{otherwise.} \end{cases} \tag{1}$$

- x dominates y if $f(x) \leq f(y)$ and $h(x) \leq h(y)$ with at least one strict inequality.
- One model for each output: $f^s \simeq f$ and $c_j^s \simeq c_j$, $j \in J$.
- A model h^s for the violation is computed as in (1).
- No model for R (yet).

Model Ordering

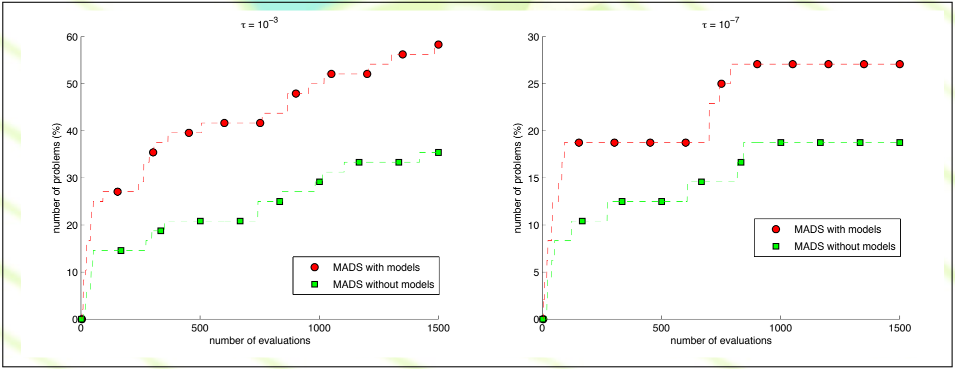
- Exploits the opportunistic strategy which consists to interrupt a series of evaluation as soon as a success is made.
- Points with $h^s = 0$ are given the highest priority.
- Predicted infeasible points are sorted accordingly to the dominance relation. Priority is given to smallest h^s values.
- Not limited to the Poll step.

Quadratic Models

- Local: data points are collected inside the ball of radius $\rho\Delta_k^p$ centered at the current solution (typically $\rho = 2$).
- The more smooth the functions are, the better the models.
- Cheap to construct (because in general $n \leq 20$).
- Under and over-determined cases:
  - If the number of data points is larger than the number of points necessary for exact quadratic interpolation, regression in the least square sense is used.
  - Otherwise (most likely), Minimum Frobenius Norm (MFN) interpolation is chosen.
- Well-poisedness (quality of the geometry of the data set) seems not to be an issue in the MADS context.

Results for quadratic models

Data profiles on a set of 48 problems (mostly academic).

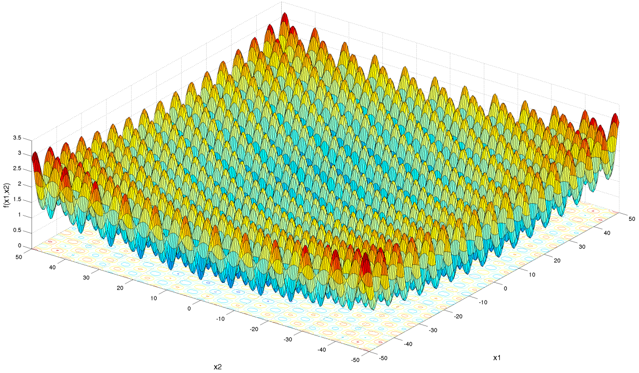


TGP Surrogates

- TGP: Treed Gaussian Processes [Gramacy and Lee, 2008]
- Space is partitioned as a tree. A Gaussian Process model is implemented at each leaf of this tree.
- Global model.
- More computationally costly.
- In addition to predictions for each output, TGP provides:
  - Expected improvement.
  - Expected reduction of variance.
- This gives other candidates in addition to the model optimization solutions.

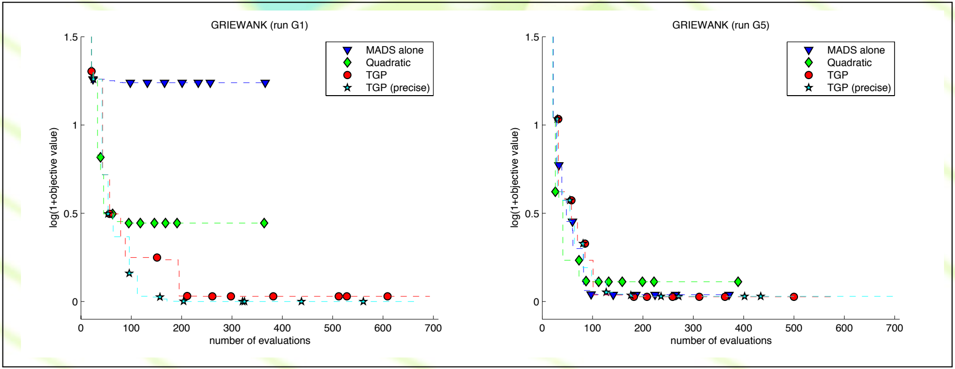
Results for TGP (1/3)

Griewank function, $n = 2$, $m = 0$, many local optima.



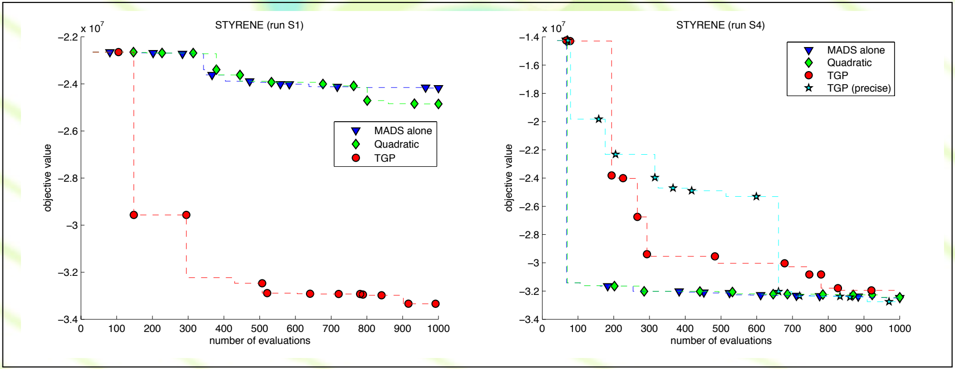
Results for TGP (2/3)

Best and worse TGP executions out of 10 runs with different seeds.
The same 20 LH points are chosen as starting solutions.



Results for TGP (3/3)

On the STYRENE problem. Out of 5 runs, the worse solution given by TGP has a 32% improvement versus 6% with quadratic models.



Blackbox optimization problems

The MADS algorithm

MADS and models

Discussion

Discussion

- Improvement of the MADS algorithm for constrained blackbox optimization with the addition of models.
- Two types of models: quadratic and TGP.
- One or the other seems more adapted depending on the situation:
  - TGP for costly and real applications with hidden constraints.
  - TGP for simple problems with many local optima.
  - Quadratic models for all other situations (default in NOMAD).