LANL 2014

Blackbox Optimization: Algorithm and Applications

Sébastien Le Digabel, École Polytechnique de Montréal

2014–03–04

The projects in this presentation involve:

- Polytechnique: Charles Audet, Christophe Tribes.
- Rice: John Dennis.
- McGill: Michael Kokkolaras.
- Chicago Booth: Bobby Gramacy.
- IBM: Andy Conn.
- Hydro-Québec: Stéphane Alarie, Louis-Alexandre Leclaire, Marie Minville.
- Students: Nadir Amaioua, Dominique Cartier, Vincent Garnier, Bastien Talgorn.

Presentation outline

The MADS algorithm

Models and Surrogates

Snow Water Equivalent estimation

Calibration of a Hydrologic Model

Biobjective optimization of aircraft takeoff trajectories

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).

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.
- MADS is available via the NOMAD free software package at www.gerad.ca/nomad.

[0] Initializations (x_0, Δ_0^m)

[1] Iteration k

- **[1.1] Search**
  - select a finite number of mesh points
  - evaluate candidates opportunistically
- **[1.2] Poll** (if the Search failed)
  - construct poll set $P_k = \{x_k + \Delta_k^m d : d \in D_k\}$
  - sort(P_k)
  - evaluate candidates opportunistically

[2] Updates

- if success
  - $x_{k+1} \leftarrow$ success point
  - increase Δ_k^m
- else
  - $x_{k+1} \leftarrow x_k$
  - decrease Δ_k^m
- $k \leftarrow k+1$, stop or go to **[1]**

The poll (1/2)

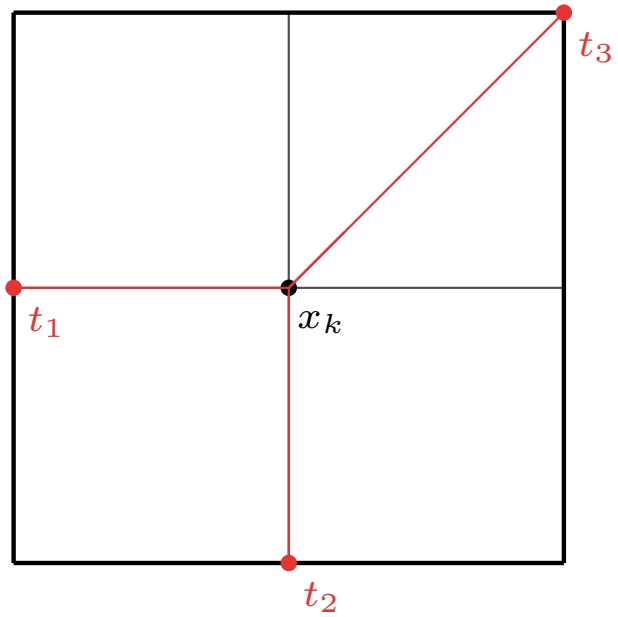
- Mesh at iteration k:
  - $M_k = \bigcup_{x \in V_k} \{x + \Delta_k^m D z : z \in \mathbb{N}^{n_D}\}$.
  - V_k is the "cache".
  - $\Delta_k^m > 0$ is the **mesh size parameter**.
  - D is a fixed set of directions typically set to $[I \ -I]$.
- Poll directions: A positive spanning set $D_k \subset \mathbb{R}^n$ where each direction $d \in D_k$ can be written as a nonnegative integer combination of directions of D.
- Poll set: $P_k = \{x_k + \Delta_k^m d : d \in D_k\}$ where x_k is the current incumbent, or the poll center.
- $\Delta_k^m ||d|| \leq \Delta_k^p$, the **poll size parameter**.

The poll (2/2)

- The directions correspond typically to a minimal positive basis ($n+1$ directions) or a maximal positive basis ($2n$ 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.

Poll illustration (successive fails and mesh shrink)

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



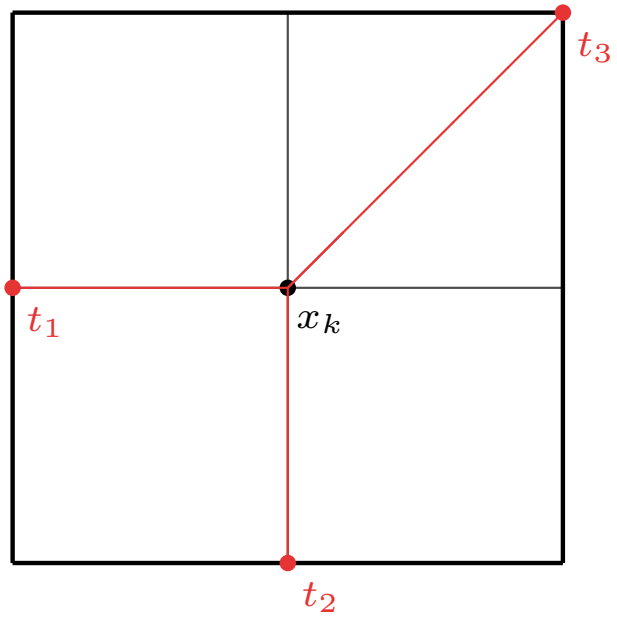
poll trial points$=\{t_1, t_2, t_3\}$

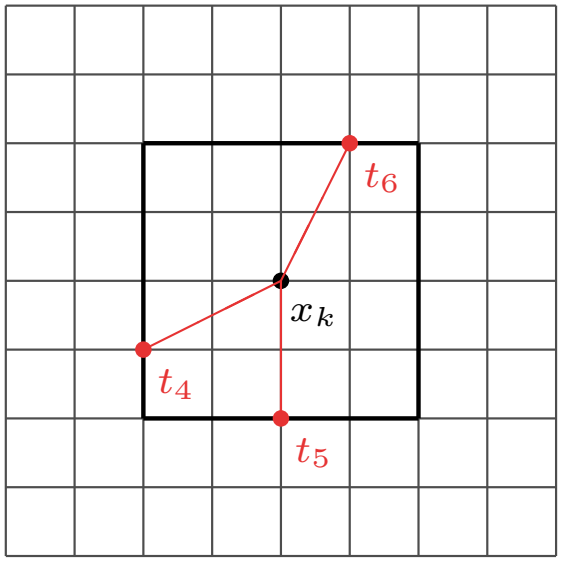
Poll illustration (successive fails and mesh shrink)

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

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

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





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

Poll illustration (successive fails and mesh shrink)

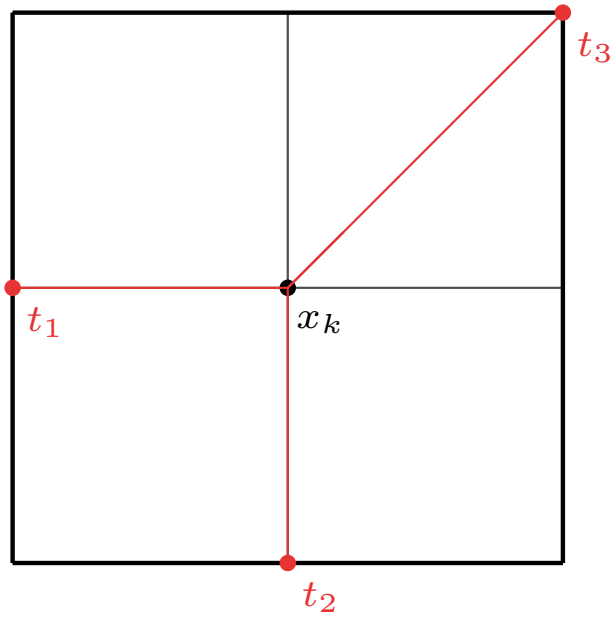
$\Delta_k^m = \Delta_k^p = 1$

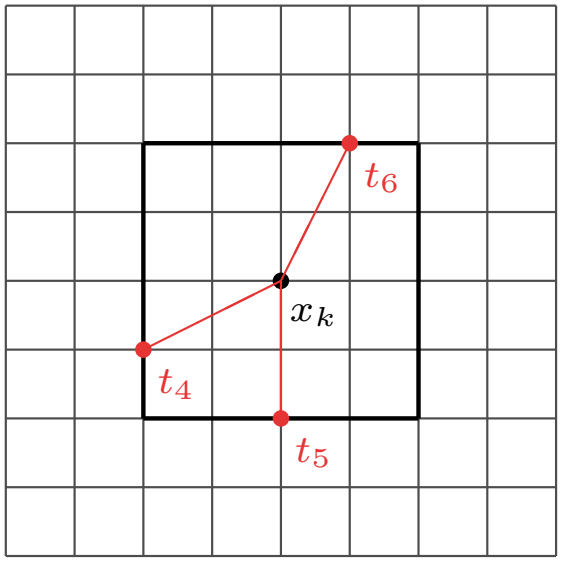
$\Delta_{k+1}^m = 1/4$

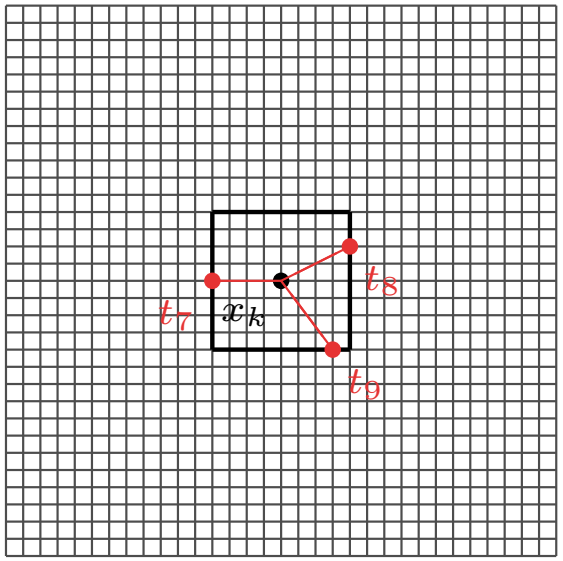
$\Delta_{k+1}^p = 1/2$

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

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







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

Constraints handling

Feasible region: $\Omega = \{x \in \mathcal{X} : c_j(x) \leq 0, j \in J\} \subset \mathbb{R}^n$.

Constraints can be relaxable, unrelaxable or hidden.

- Unrelaxable constraints define $\mathcal{X}$

  Cannot be violated by any trial point.
  For example, logical conditions on the variables indicating if the simulation may be launched.

Constraints handling

Feasible region: $\Omega = \{x \in \mathcal{X} : c_j(x) \leq 0, j \in J\} \subset \mathbb{R}^n$.

Constraints can be relaxable, unrelaxable or hidden.

- Unrelaxable constraints define $\mathcal{X}$
- Relaxable constraints $c_j(x) \leq 0$

  Can be violated, and $c_j(x)$ provides a measure of how much the constraint is violated. A budget for example.

Constraints handling

Feasible region: $\Omega = \{x \in \mathcal{X} : c_j(x) \leq 0, j \in J\} \subset \mathbb{R}^n$.

Constraints can be relaxable, unrelaxable or hidden.

- Unrelaxable constraints define $\mathcal{X}$
- Relaxable constraints $c_j(x) \leq 0$
- Hidden constraints

  A convenient term to denote the set of points in the feasible region for the relaxable or unrelaxable constraints at which the blackbox fails to return a value for one of the problem functions.
  A typical example is when the simulation fails to return a value.

Three strategies to deal with constraints

- Extreme barrier (EB)

Treats the problem as being unconstrained, by replacing the objective function $f(x)$ by

$$f_\Omega(x) := \begin{cases} f(x) & \text{if } x \in \Omega, \\ \infty & \text{otherwise.} \end{cases}$$

The problem

$$\min_{x \in \mathbb{R}^n} f_\Omega(x)$$

is then solved.
Remark : If $x \notin \mathcal{X}$ (the unrelaxable constraints), then the costly evaluation of $f(x)$ is not performed.

Three strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)

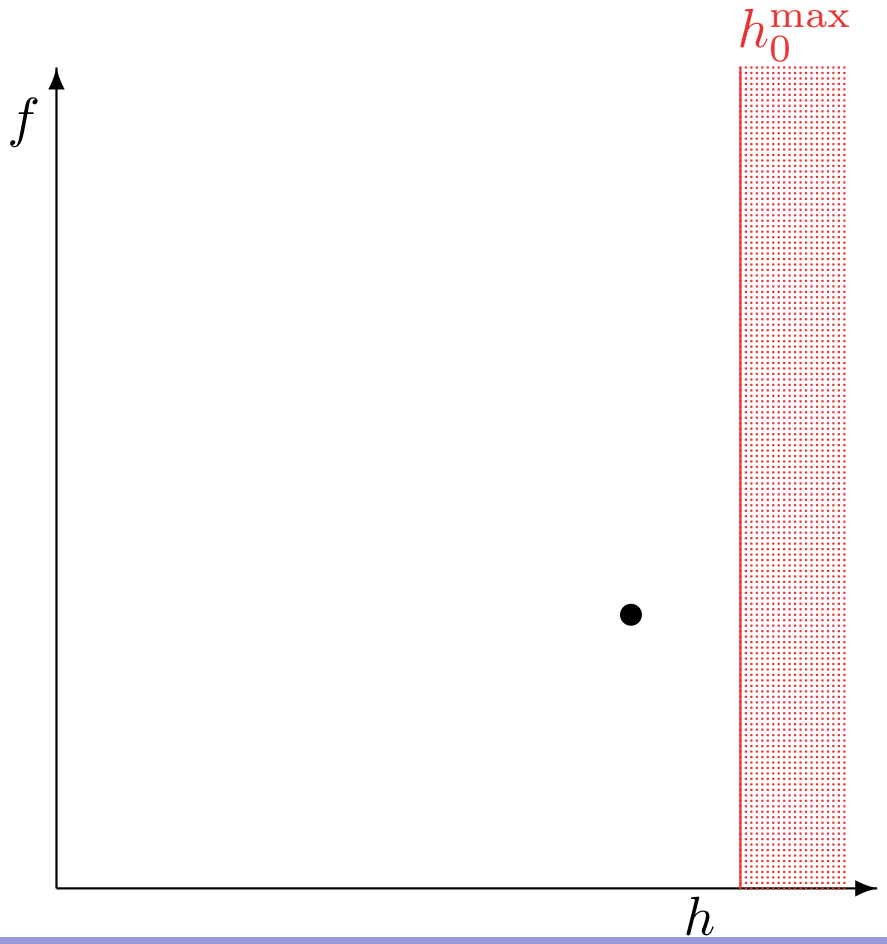
  Defined for the relaxable constraints.
  As in the filter methods of Fletcher and Leyffer, it uses the non-negative constraint violation function $h : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$

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

  At iteration k, points with $h(x) > h_k^{\max}$ are rejected by the algorithm, and $h_k^{\max}$ decreases toward 0 as $k \to \infty$.

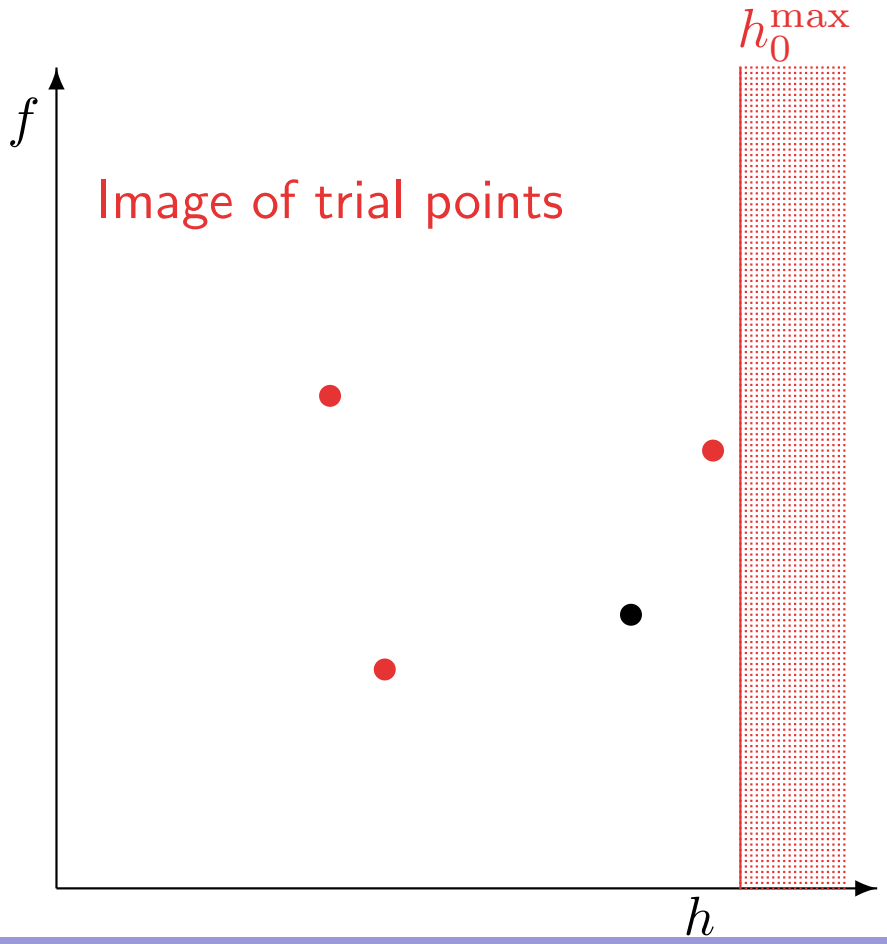
Three strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)



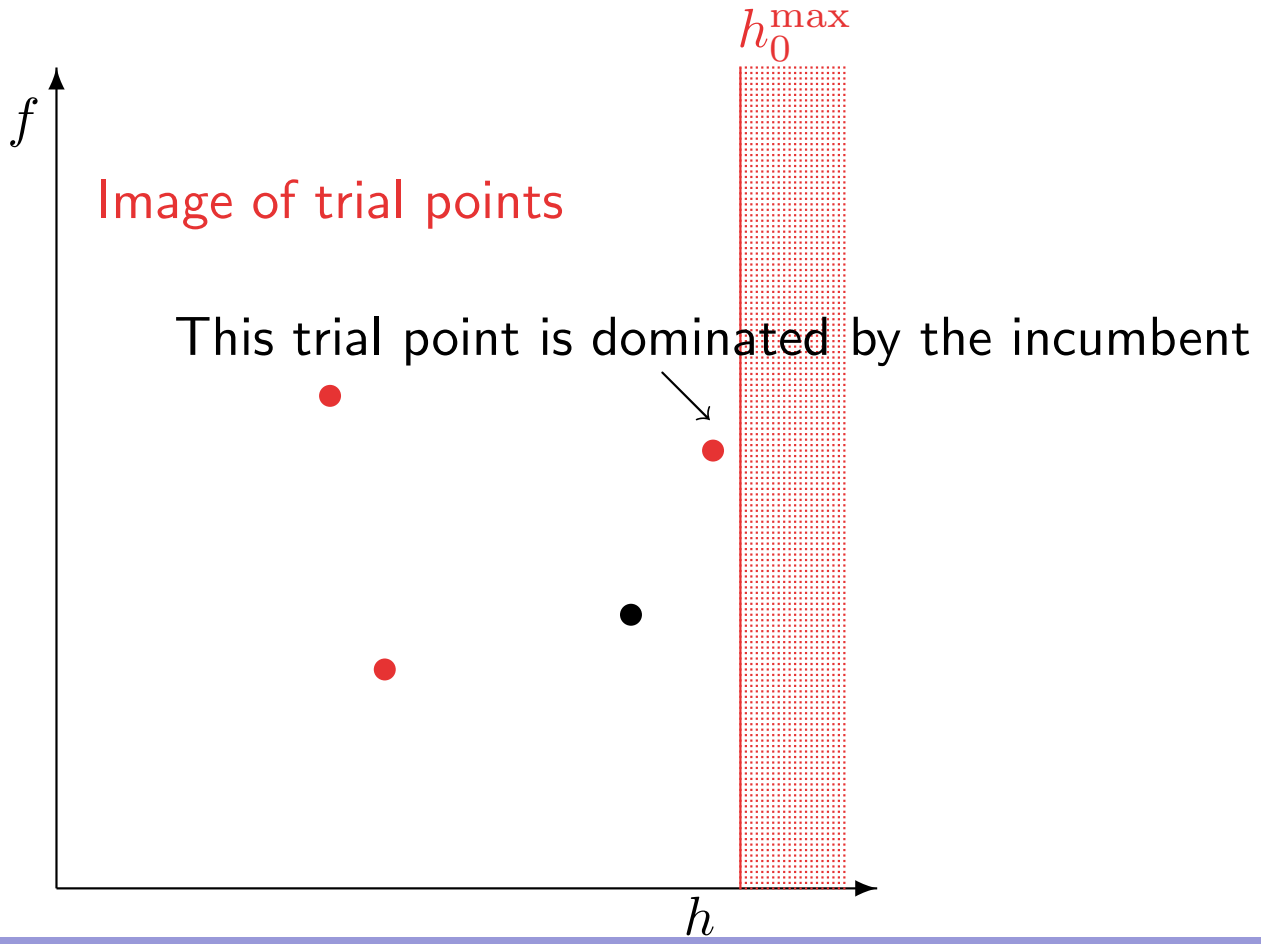
Three strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)



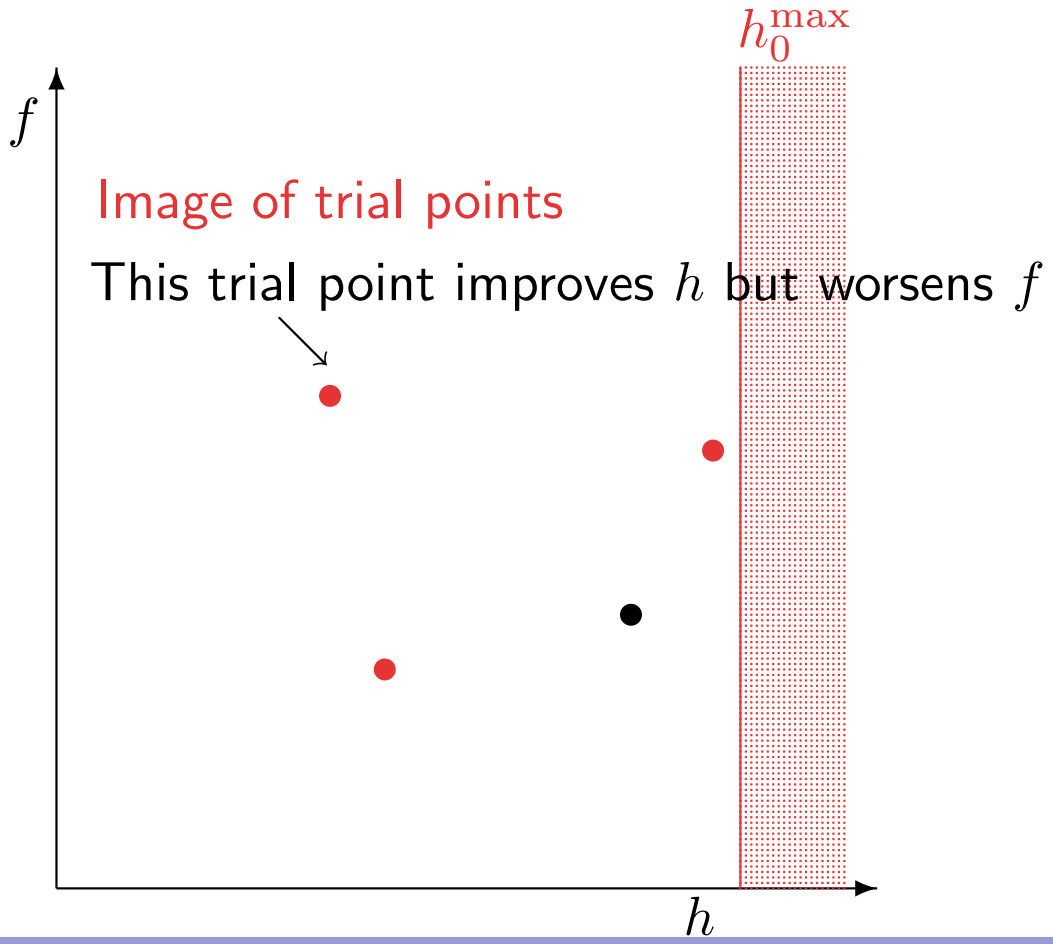
Three strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)



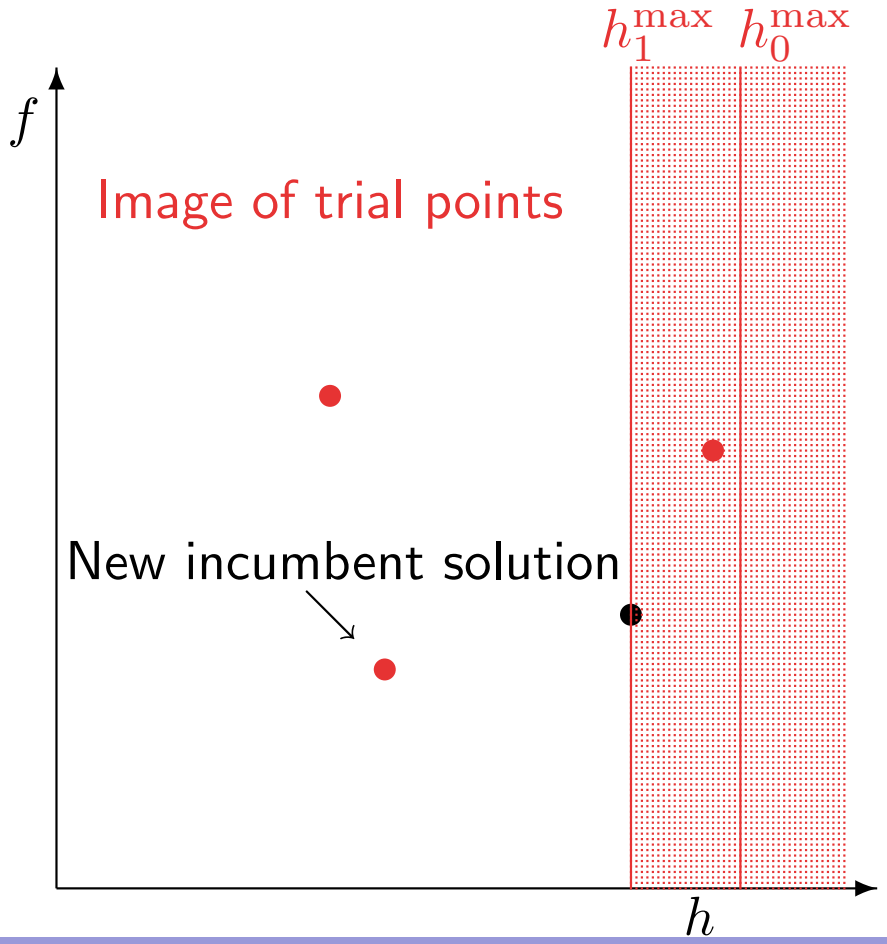
Three strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)



Three strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)

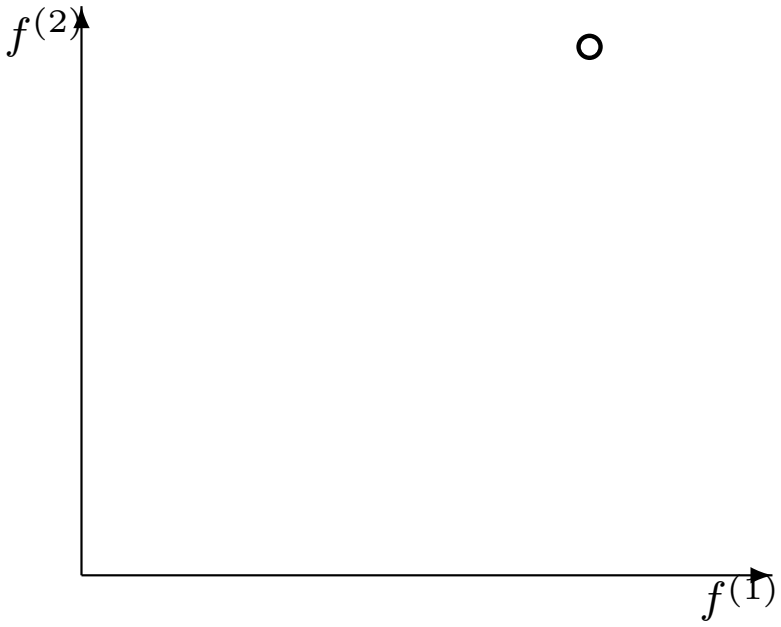


Three strategies to deal with constraints

- Extreme barrier (EB)
- Progressive barrier (PB)
- Progressive-to-Extreme Barrier (PEB)

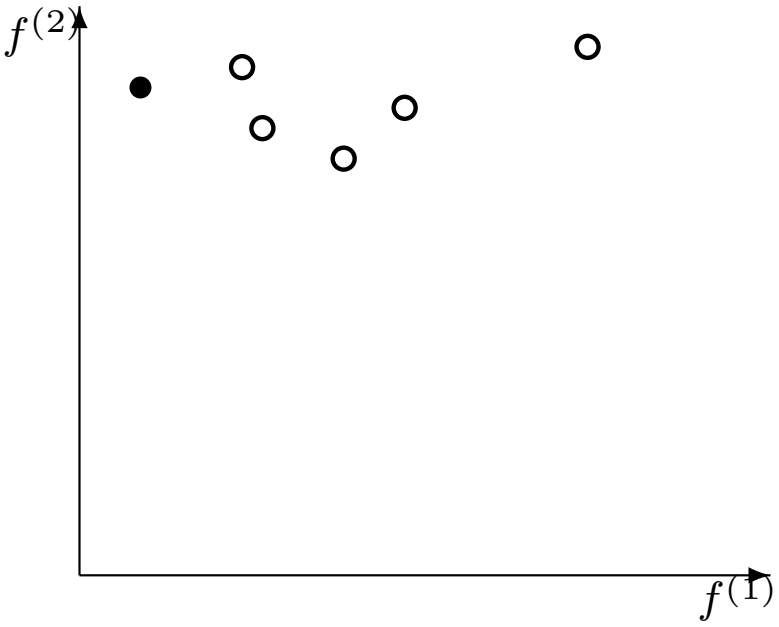
  Initially treats a relaxable constraint by the progressive barrier. Then, if polling around the infeasible poll center generates a new infeasible incumbent that satisfies a constraint violated by the poll center, then that constraint moves from being treated by the progressive barrier to the extreme barrier.

Biobjective optimization: successive MADS runs



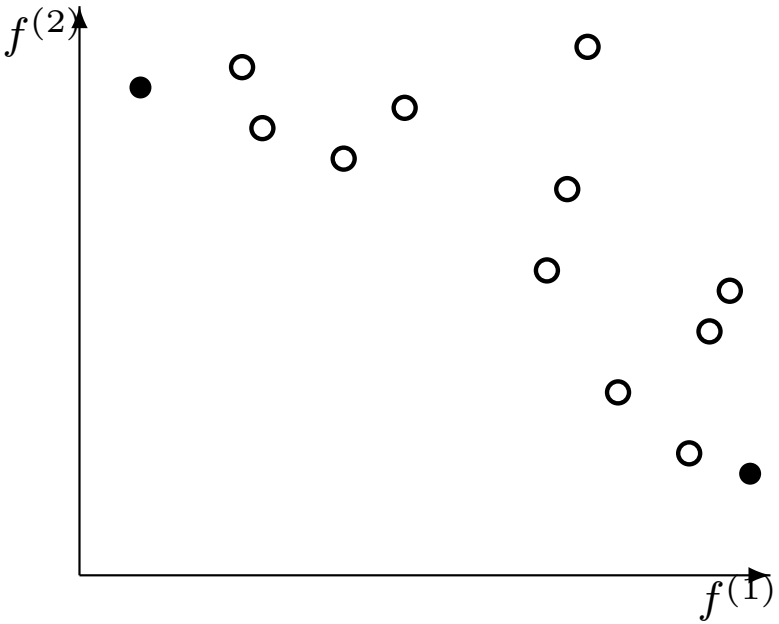
- INITIALIZATION:
  Solve $\min_{x \in \Omega} f^{(q)}(x)$ for $q \in \{1, 2\}$.

Biobjective optimization: successive MADS runs



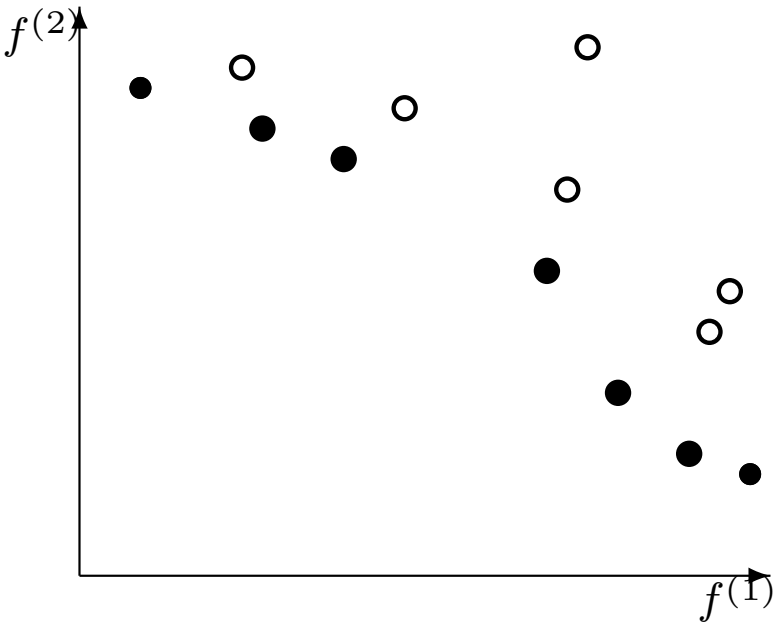
- INITIALIZATION:
  Solve $\min_{x \in \Omega} f^{(q)}(x)$ for $q \in \{1, 2\}$.

Biobjective optimization: successive MADS runs



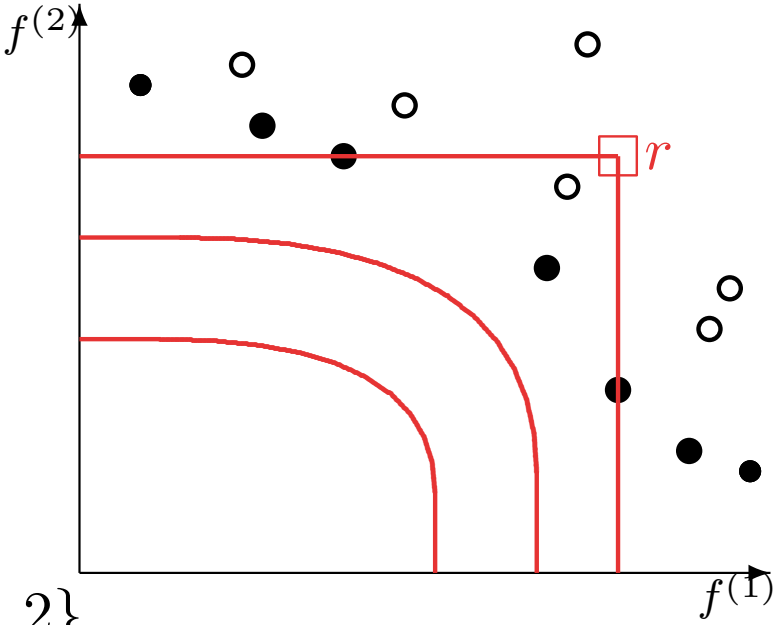
- INITIALIZATION:
  Solve $\min_{x \in \Omega} f^{(q)}(x)$ for $q \in \{1, 2\}$.

Biobjective optimization: successive MADS runs



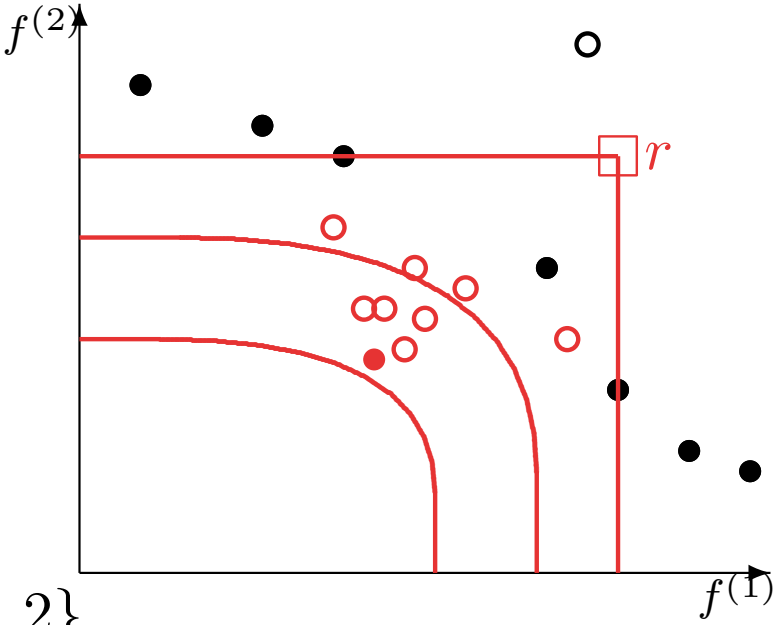
- INITIALIZATION:
  Solve $\min_{x \in \Omega} f^{(q)}(x)$ for $q \in \{1, 2\}$.
- MAIN ITERATIONS:
  - REFERENCE POINT DETERMINATION:
    Use the set of feasible ordered undominated points generated so far to generate a reference point r.

Biobjective optimization: successive MADS runs



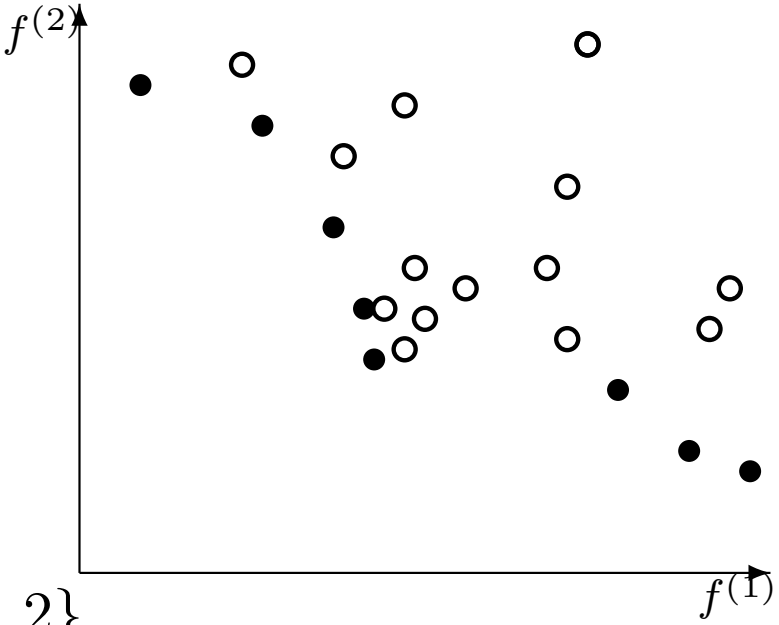
- INITIALIZATION:
  Solve $\min_{x \in \Omega} f^{(q)}(x)$ for $q \in \{1, 2\}$.
- MAIN ITERATIONS:
  - REFERENCE POINT DETERMINATION:
    Use the set of feasible ordered undominated points generated so far to generate a reference point r.
  - SINGLE-OBJECTIVE MINIMIZATION:
    Solve the problem $\max_{x \in \Omega} (r_1 - f^{(1)}(x))_+^2 (r_2 - f^{(2)}(x))_+^2$.

Biobjective optimization: successive MADS runs



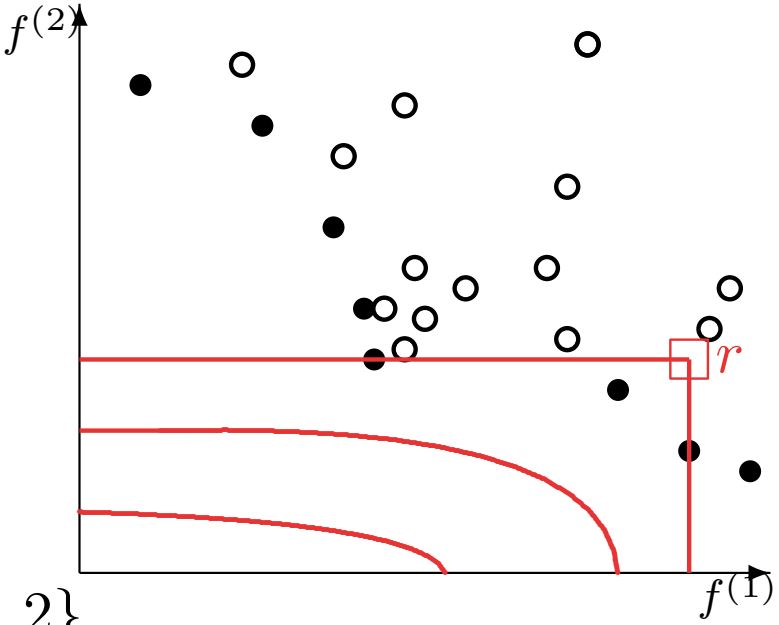
- INITIALIZATION:
  Solve $\min_{x \in \Omega} f^{(q)}(x)$ for $q \in \{1, 2\}$.
- MAIN ITERATIONS:
  - REFERENCE POINT DETERMINATION:
    Use the set of feasible ordered undominated points generated so far to generate a reference point r.
  - SINGLE-OBJECTIVE MINIMIZATION:
    Solve the problem $\max_{x \in \Omega} (r_1 - f^{(1)}(x))_+^2 (r_2 - f^{(2)}(x))_+^2$.

Biobjective optimization: successive MADS runs



- INITIALIZATION:
  Solve $\min_{x \in \Omega} f^{(q)}(x)$ for $q \in \{1, 2\}$.
- MAIN ITERATIONS:
  - REFERENCE POINT DETERMINATION:
    Use the set of feasible ordered undominated points generated so far to generate a reference point r.
  - SINGLE-OBJECTIVE MINIMIZATION:
    Solve the problem $\max_{x \in \Omega} (r_1 - f^{(1)}(x))_+^2 (r_2 - f^{(2)}(x))_+^2$.

Biobjective optimization: successive MADS runs



- INITIALIZATION:
  Solve $\min_{x \in \Omega} f^{(q)}(x)$ for $q \in \{1, 2\}$.
- MAIN ITERATIONS:
  - REFERENCE POINT DETERMINATION:
    Use the set of feasible ordered undominated points generated so far to generate a reference point r.
  - SINGLE-OBJECTIVE MINIMIZATION:
    Solve the problem $\max_{x \in \Omega} (r_1 - f^{(1)}(x))_+^2 (r_2 - f^{(2)}(x))_+^2$.

The MADS algorithm

Models and Surrogates

Snow Water Equivalent estimation

Calibration of a Hydrologic Model

Biobjective optimization of aircraft takeoff trajectories

Static versus dynamic models/surrogates

- Static surrogate: A cheaper model defined a priori by the user. It is used as a blackbox too. Typically a simplified physics model. Variable precision is not yet considered.
- Dynamic surrogate: Model managed by the algorithm, based on past evaluations. It can be periodically updated.

In the remaining of this presentation, we focus on dynamic surrogates based on quadratic models and on the `dynaTree` library.

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
  - *(→ the subproblem, or surrogate problem)*
  - 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

Model Ordering

- Exploits the opportunistic strategy which consists to interrupt a series of evaluation as soon as a success is made.
- Predicted feasible points are given the highest priority.
- Predicted infeasible points are sorted accordingly to the dominance relation defined by the couples (f, h). Priority is given to smallest predicted h 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.

The subproblem with quadratic models

- At each Search step, the following subproblem is solved:

$$\begin{cases} \min\limits_{x \in \mathcal{X}} & \hat{f}(x) \\ \text{s.t.} & \hat{c}_j(x) \leq 0 \;\; \forall j \in J \; . \end{cases}$$

- $\hat{f}$ and $\hat{c}_j$, $j \in J$, are the models of f and c_j, $j \in J$, respectively.
- Currently solved with MADS.
- Dedicated solvers are currently tested.

The `dynaTree` library

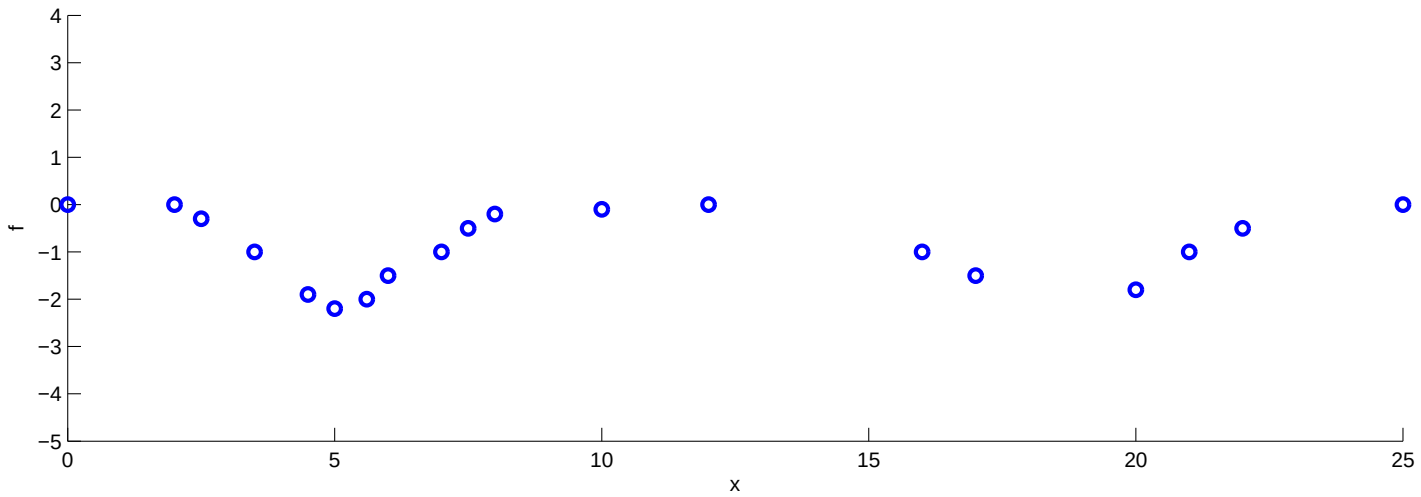
- Developed for parameter-free regression by Taddy, Gramacy, and Polson.
- Based on Bayesian inference.
- `R` package available on CRAN.

Main idea:

- **Hypothesis**: the function is piecewise linear with gaussian noise.
- Partitioning of the design space in p parts:
$$\mathbb{R}^n = \bigcup_{i=1}^{p} \eta_i \qquad \text{with } \eta_i \cap \eta_j = \varnothing \;\; \forall i \neq j.$$
- In each part η_i, linear predictive mean:
$$x \in \eta_i \Rightarrow \hat{f}(x) = \mathbb{E}[f(x)] = \alpha_i + \beta_i x, \;\; \alpha_i \in \mathbb{R}, \; \beta_i \in \mathbb{R}^n.$$

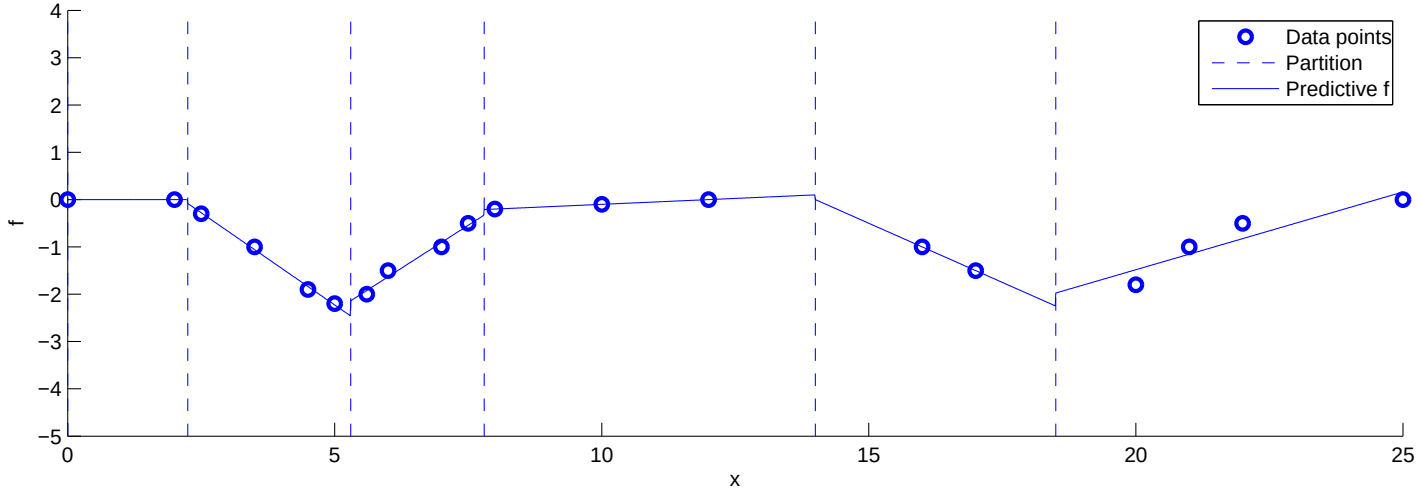
dynaTree example

Data points: $\{x_i, f(x_i)\}_{i=1,\ldots,p}$

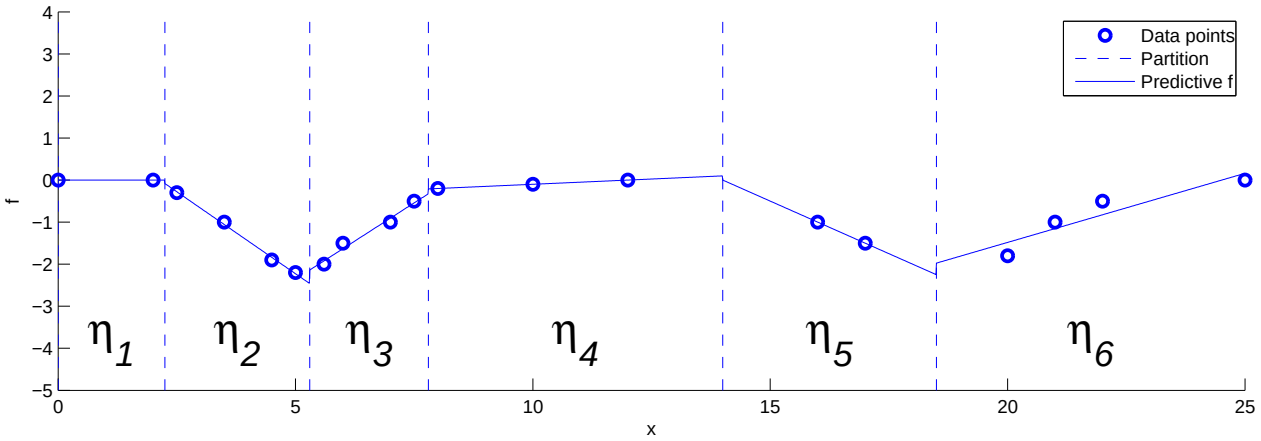


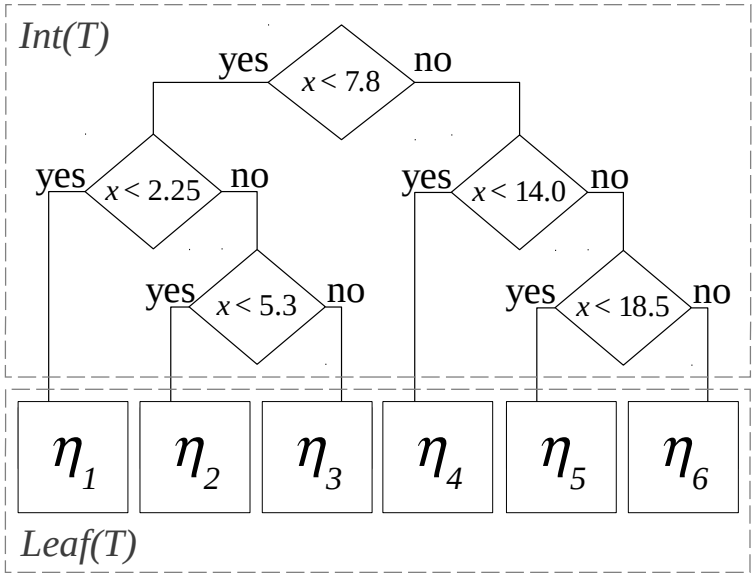
dynaTree example

Data points: $\{x_i, f(x_i)\}_{i=1,\ldots,p}$



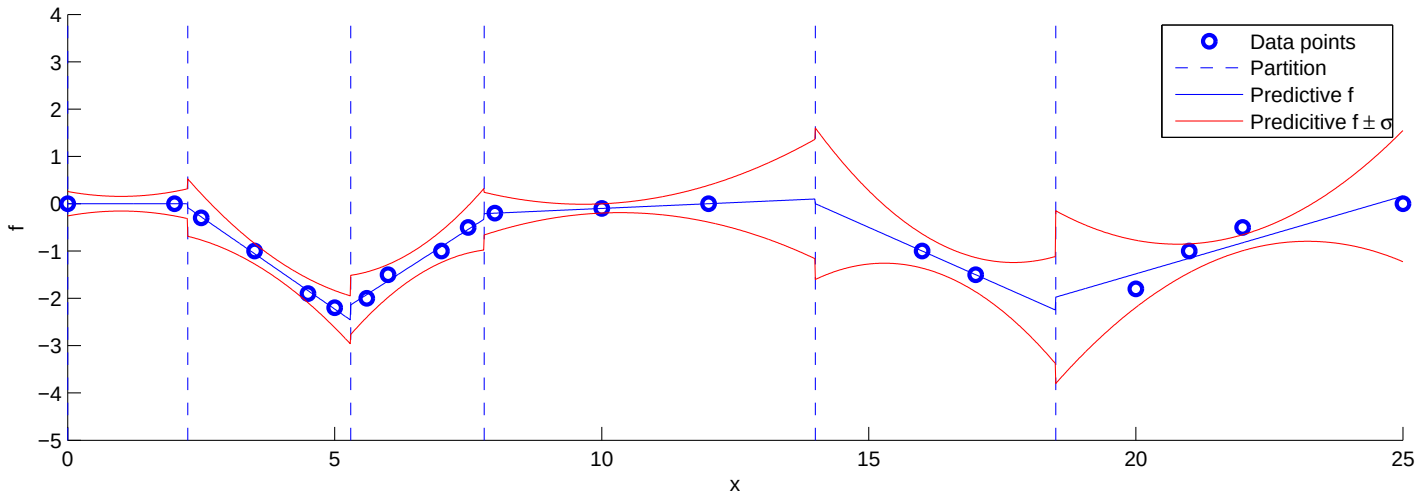
dynaTree example





dynaTree outputs

- The predictive mean $\hat{f}(x)$.
- The predictive standard deviation $\hat{\sigma}_f(x)$.



- The predictive cumulative distribution $\mathbb{P}[\hat{f}(x) \leq f_0]$.
- Likewise for the constraints: $\hat{c}_j(x)$, $\hat{\sigma}_j(x)$, $\mathbb{P}[\hat{c}_j(x) \leq c_0]$.

Surrogate problem formulations

- At each iteration, find the most promising candidates by solving the surrogate problem.
- The most basic formulation is:

$$\begin{cases} \min\limits_{x \in \mathcal{X}} & \hat{f}(x) \\ \text{s.t.} & \hat{c}_j(x) \leq 0 \;\; \forall j \in J \; . \end{cases}$$

- We tested other formulations in the following submitted manuscript: B. Talgorn, S. Le Digabel, and M. Kokkolaras: *Problem Formulations for Simulation-based Design Optimization using Statistical Surrogates and Direct Search* [Optimization Online].

Diversification term

The standard deviation is added to the blackbox outputs:

$$(F\sigma) \quad \begin{cases} \min\limits_{x \in \mathcal{X}} & \hat{f}(x) - \lambda.\hat{\sigma}_f(x) \\ \text{s.t.} & \hat{c}_j(x) - \lambda.\hat{\sigma}_j(x) \leq 0 \;\; \forall j \in J \; . \end{cases}$$

$\lambda > 0 \Rightarrow$ diversification: focus more on exploration than on a particular region. The feasible domain is extended and poor values of $\hat{f}$ may be considered if $\hat{\sigma}_f$ is large.

Probability of feasibility of a point

One continuous `dynaTree` model is built for each constraint.

$$\mathbb{P}[x] = \mathbb{P}[x \text{ is feasible}] = \prod_{j \in J} \mathbb{P}[c_j(x) \leq 0] \ .$$

$\Rightarrow$ one scalar statistical measure to handle the constraints.

Chance constraint

$$(F\sigma P) \quad \begin{cases} \min\limits_{x \in \mathcal{X}} & \hat{f}(x) - \lambda.\hat{\sigma}_f(x) \\ \text{s.t.} & \mathbb{P}[x \text{ is feasible}] \geq p_c \ . \end{cases}$$

Diversification is possible (with parameter λ), but only candidates which are likely to be feasible are evaluated.

Generally, $p_c = \frac{1}{2}$, but it can be tailored according to the number of constraints.

Improvement

Improvement: $I(x) = \max\{f_{min} - f(x), 0\}$:

- M. Schonlau, D.R. Jones, and W.J. Welch [JOGO, 1998].
- f_{min}: current best known solution value.
- $I(x) > 0$ if x is better than the incumbent solution.
- $I(x) = 0$ otherwise.

Two statistical measurements:

- Probability of improvement: $PI(x) = \mathbb{P}\left[I(x) > 0\right]$.
- Expected improvement: $EI(x) = \mathbb{E}\left[I(x)\right]$.

What to do with constraints?

Probability of feasible improvement

Probabilities on:

- The objective: $PI(x)$
- The feasibility: $\mathbb{P}[x]$

$\Rightarrow$ Probability of Feasible Improvement:

$$PFI(x) = \mathbb{P}[x \text{ is feasible}] \times PI(x)$$

$$(PFI) \quad \left\{ \min_{x \in \mathcal{X}} \; -PFI(x) \right.$$

Expected improvement subject to constraints

Maximization of the expected improvement under constraints:

$$(EI\sigma) \quad \begin{cases} \min\limits_{x \in \mathcal{X}} & -EI(x) - \lambda.\hat{\sigma}_f(x) \\ \text{s.t.} & \hat{c}_j(x) - \lambda.\hat{\sigma}_j(x) \leq 0 \;\; \forall j \in J \end{cases}$$

Expected feasible improvement (EFI)

Statistical measurement of

- The objective: $EI(x)$
- The feasibility: $\mathbb{P}[x]$

$\Rightarrow$ expected feasible improvement:

$$EFI(x) = \mathbb{P}[x \text{ is feasible}] \times \mathbb{E}\left[I(x)\right]$$

The EFI represents what a candidate will statistically yield, in regard to the optimization problem.

Expected feasible improvement (EFI)

Maximization of the expected feasible improvement:

$$(EFI) \quad \left\{ \min_{x \in \mathcal{X}} \; -EFI(x) \right.$$

EFI with a diversification term:

$$(EFI\sigma) \quad \left\{ \min_{x \in \mathcal{X}} \; -EFI(x) - \lambda . \hat{\sigma}_f(x) \right.$$

List of surrogate problem formulations

$(F\sigma)$	$\begin{cases} \min\limits_{x\in\mathcal{X}} & \hat{f}(x) - \lambda.\hat{\sigma}_f(x) \\ st: & \hat{c}_j(x) - \lambda.\hat{\sigma}_j(x) \leq 0 \end{cases}$	$\lambda \in \left\{0, \frac{1}{100}, \frac{1}{10}, 1\right\}$
$(F\sigma P)$	$\begin{cases} \min\limits_{x\in\mathcal{X}} & \hat{f}(x) - \lambda.\hat{\sigma}_f(x) \\ st: & \mathbb{P}[x \text{ is feasible}] \geq p_c \end{cases}$	
$(EI\sigma)$	$\begin{cases} \min\limits_{x\in\mathcal{X}} & EI(x) - \lambda.\hat{\sigma}_f(x) \\ st: & \hat{c}_j(x) - \lambda.\hat{\sigma}_j(x) \leq 0 \end{cases}$	
$(EFI\sigma)$	$\min\limits_{x\in\mathcal{X}} -EFI(x) - \lambda.\hat{\sigma}_f(x)$	
(PFI)	$\min\limits_{x\in\mathcal{X}} -PFI(x)$	

Current conclusions

- Set of 20 analytic problems: $(F\sigma)$ with $\lambda = 1$ performs better.
- For two realistic MDO applications, $(EI\sigma)$ with $\lambda = 1/100$ and 1, and $(EFI\sigma)$ with $\lambda = 1/10$, gave the best results.
- `NOMAD` with `dynaTree` should be available in the next `NOMAD` release.

The MADS algorithm

Models and Surrogates

Snow Water Equivalent estimation

Calibration of a Hydrologic Model

Biobjective optimization of aircraft takeoff trajectories

Importance of the Snow Water Equivalent (SWE)

- Accurate estimate of water stored in snow is crucial to optimize hydroelectric plants management.
- Exact snow measurement is impossible.
- SWE is measured at specific sites and next interpolated over the territory.
- Territory is huge: Hydro-Québec (HQ) operates 565 dams, 75 reservoirs, and 56 hydroelectric power plants, located over 90 watersheds and covering more than 550,000 km^2.



SWE estimation

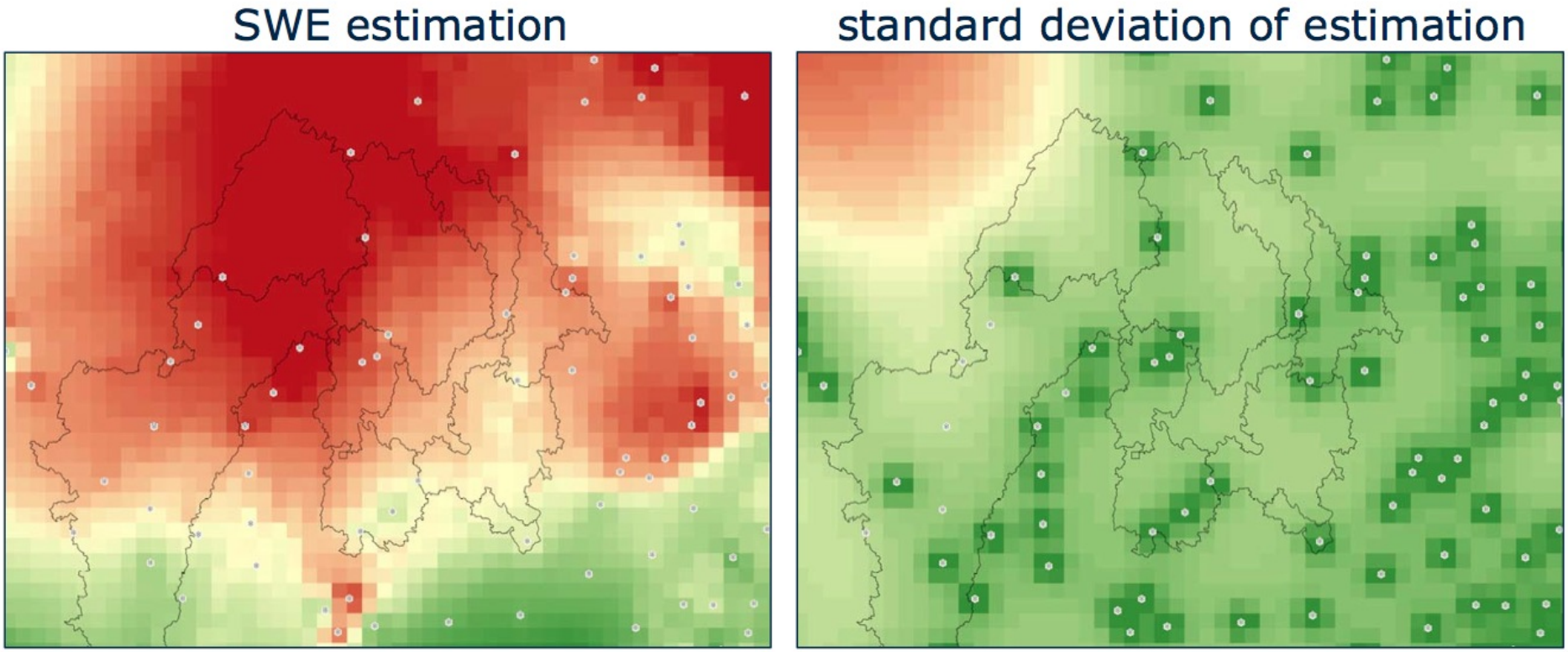
- Presently, done manually by weighing snow cores at specific sites.
- Each measurement campaign requires 2 weeks.
- Missing measurements due to adverse meteorological conditions.

GMON device

- A new measuring instrument that provides daily automatic SWE.
- GMON for Gamma-MONitoring device: it measures the natural Gamma radiation emitted from the soil.
- Communicates via satellites.

SWE estimation from GMON measures

- Kriging interpolation is used to obtain SWE estimation together with an error map.
- How to find the device locations that minimize the kriging interpolation error of the SWE?

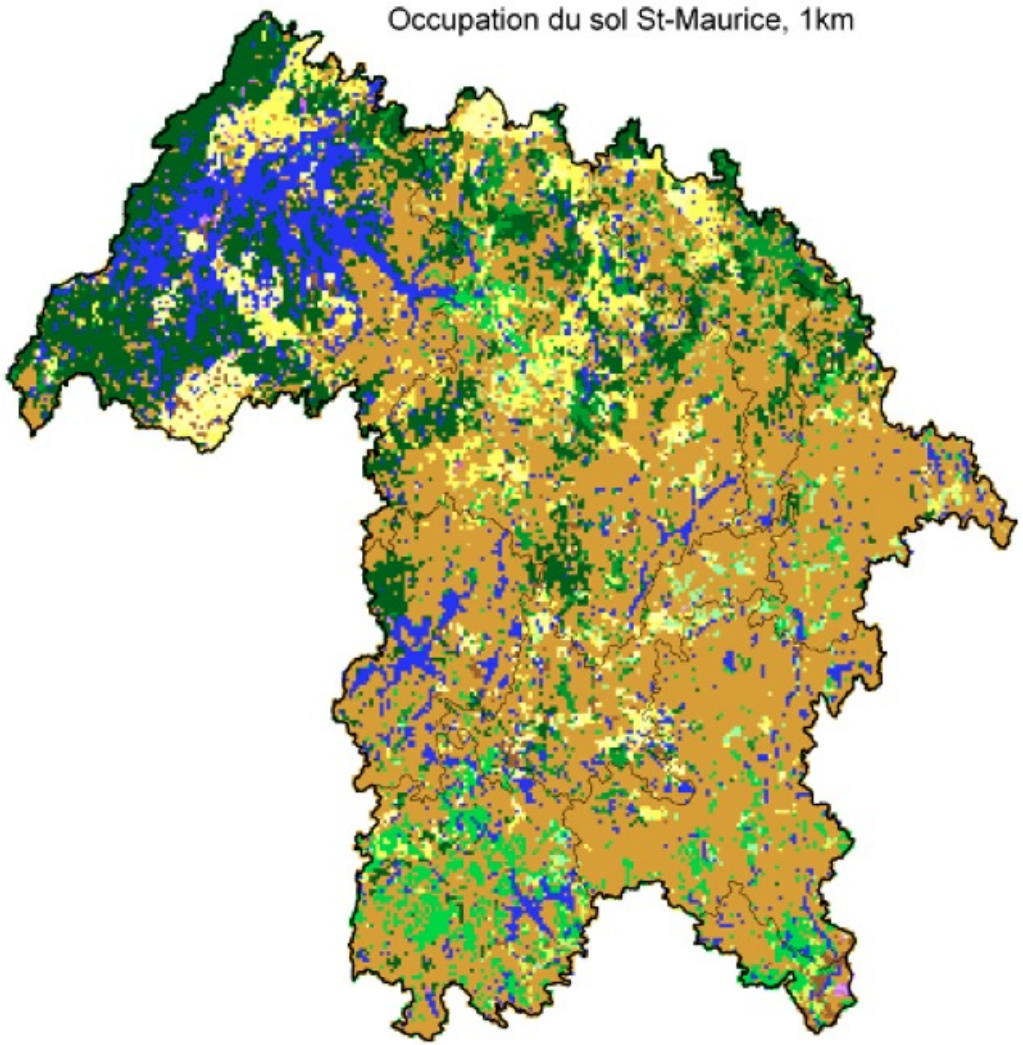


Problem formulation

- $x \in \mathbb{R}^{2N}$ are the locations of N stations.
- Typically, $N \leq 10$, so we do not consider it as a variable.
- $\Omega \subseteq \mathbb{R}^2$ is the feasible domain where the stations can be located.
- $f(x)$ is a score based on the standard deviation map obtained by the kriging simulation and is considered as a blackbox.
- Each simulation requires $\simeq 2$ seconds, and can only be launched within the Hydro-Québec research center.

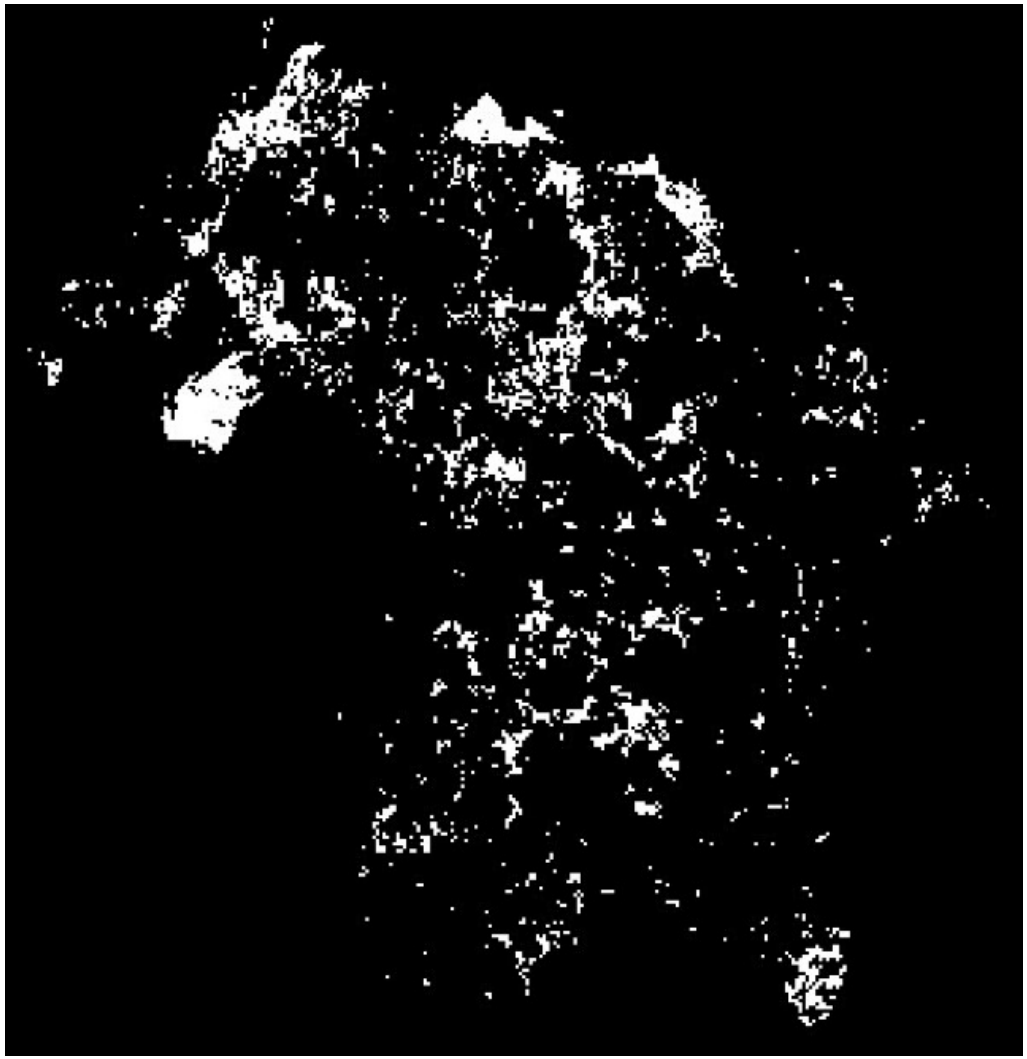
Constraints

- GMON stations cannot be located anywhere.
- Restrictions on:
  - subsoil properties,
  - slope,
  - vegetation,
  - exploitation,
  - etc.



Constraints

- GMON stations cannot be located anywhere.
- Restrictions on:
  - subsoil properties,
  - slope,
  - vegetation,
  - exploitation,
  - etc.
- Highly fragmented domain.



Special features

- Fragmented domain: Heuristic directly integrated in the simulator to identify the closest feasible location.
- Groups of variables:
  - Variables represent 2D locations.
  - Makes sense to simultaneously move both GMON coordinates.
  - Different grouping strategies are developed.
  - Some are dynamic: groups are changed during the optimization.
- Static surrogate:
  - Cheap replacement of the true function.
  - Simple analytic expression of the objective.
  - Allowed the algorithm design outside of Hydro-Québec.
  - Parameters defining the surrogate were chosen in collaboration with Hydro-Québec experts, by comparing corresponding error maps.

Results

- Three maps: Gatineau, Saint-Maurice and La Grande.
- The number of GMON stations varies from $N = 5$ to 10, for a total of 18 test instances.
- Dynamically regrouping the variables is preferable than either moving individual variables, or moving all variables simultaneously.
- Some strategies developed in this work are specific to positioning problems, other are generic.

The MADS algorithm

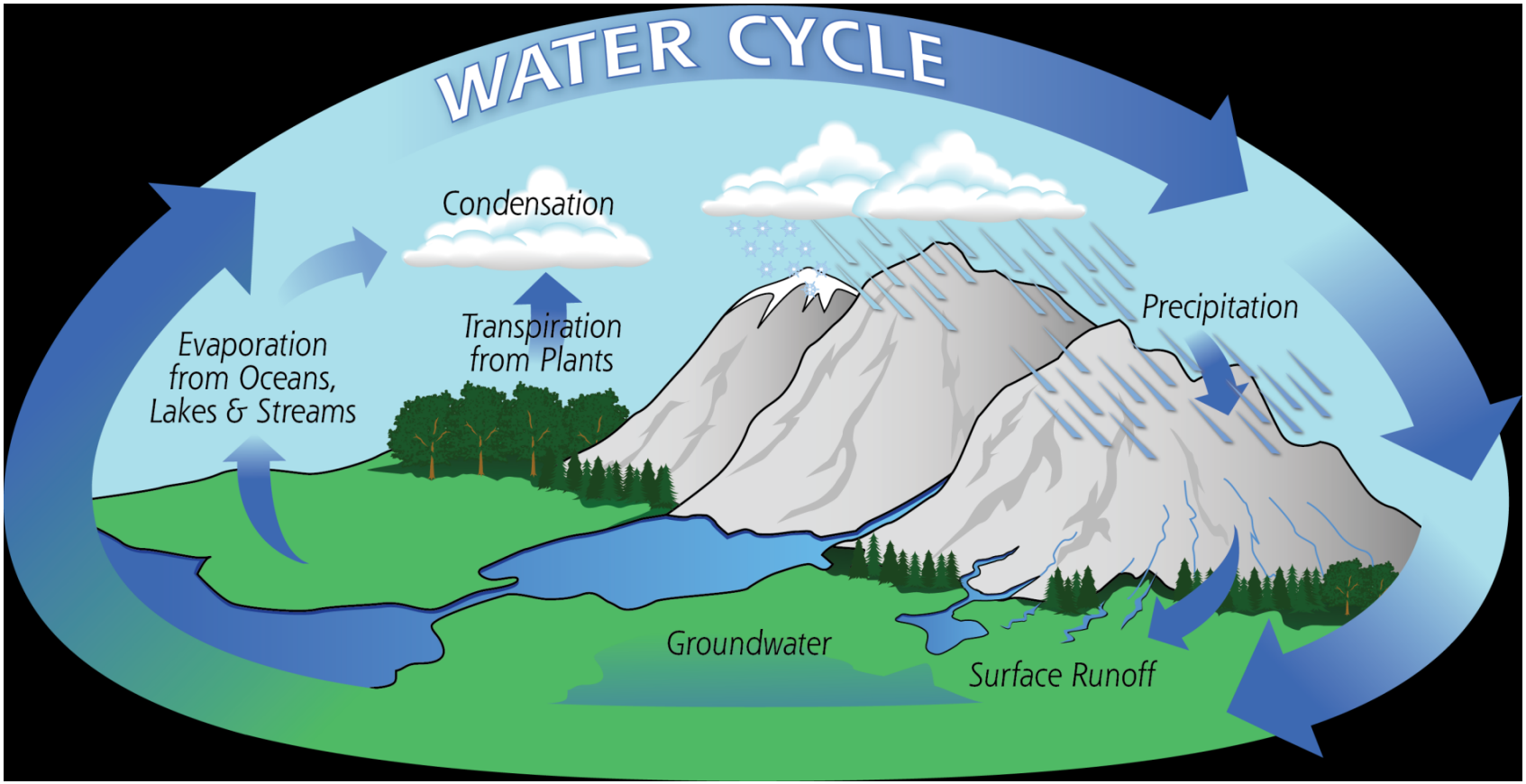
Models and Surrogates

Snow Water Equivalent estimation

Calibration of a Hydrologic Model

Biobjective optimization of aircraft takeoff trajectories

The Water Cycle



credit: NASA.

Evaporation + Transpiration = Evapotranspiration.

Objectives

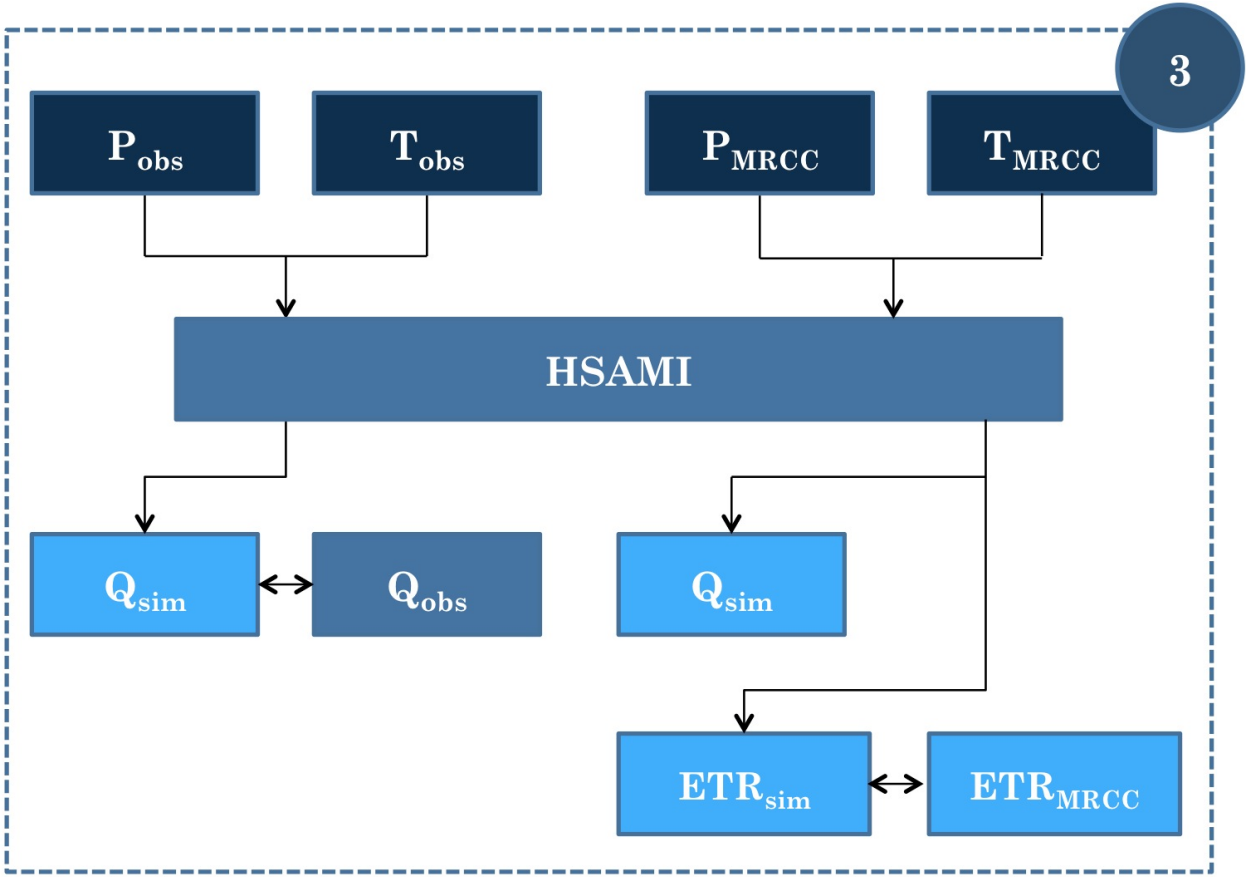
- Define a calibration (= parameters optimization) approach in order to improve the transposability of the hydrologic model.
- A transposable model should adequately reproduce hydrologic processes when they are employed with other data than those used to obtain the parameters (e.g. climate change).
- Emphasis on a realistic representation of evapotranspiration.
- Characteristics of the optimization problem: Nonsmoothness, multiple regions of attraction, and many local optima within each region of attraction.

The model

- HSAMI (*Service hydrométéorologique apports modulés intermédiaires*) [Bisson, Roberge, 1983] [Fortin, 1999].
- Hydrologic model developed and used at Hydro-Québec.
- 23 parameters: optimization variables.
- One evaluation takes $\simeq$ 1-2 seconds.
- We compare the simulated and observed streamflows and minimize the Nash-Sutcliffe criteria $\frac{\sum_{t=1}^{T}(Q_t^o - Q_t^s)^2}{\sum_{t=1}^{T}(Q_t^o - \overline{Q^o})^2}$.
- Cross-validation typically over half the data.

Definition of the evapotranspiration (ETR) constraint

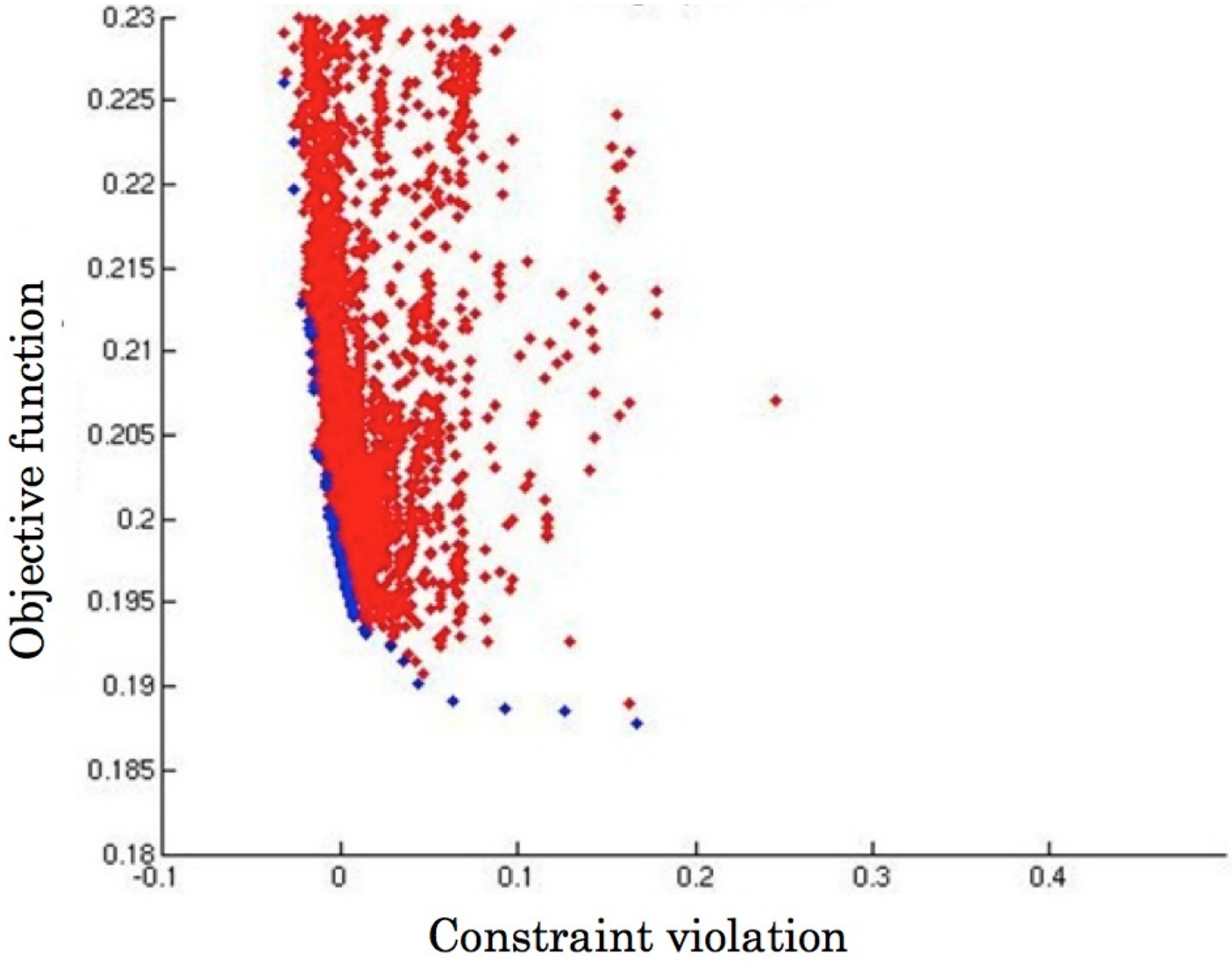
Calibration of the ETR is achieved by considering a climatic model (MRCC) for known values of P, T, and ETR.



Special features

- Progressive Barrier [SIOPT 2009] to treat the constraint.
- VNS (Variable Neighborhood Search) [JOGO 2008]: Useful in the presence of many local optima. Costs more evaluations but helps to achieve global optimization. For the present project, VNS gave improvements of up to 12%.
- Tool for the sensitivity analysis of the constraints [OMS 2012].

Sensitivity Analysis



The MADS algorithm

Models and Surrogates

Snow Water Equivalent estimation

Calibration of a Hydrologic Model

Biobjective optimization of aircraft takeoff trajectories

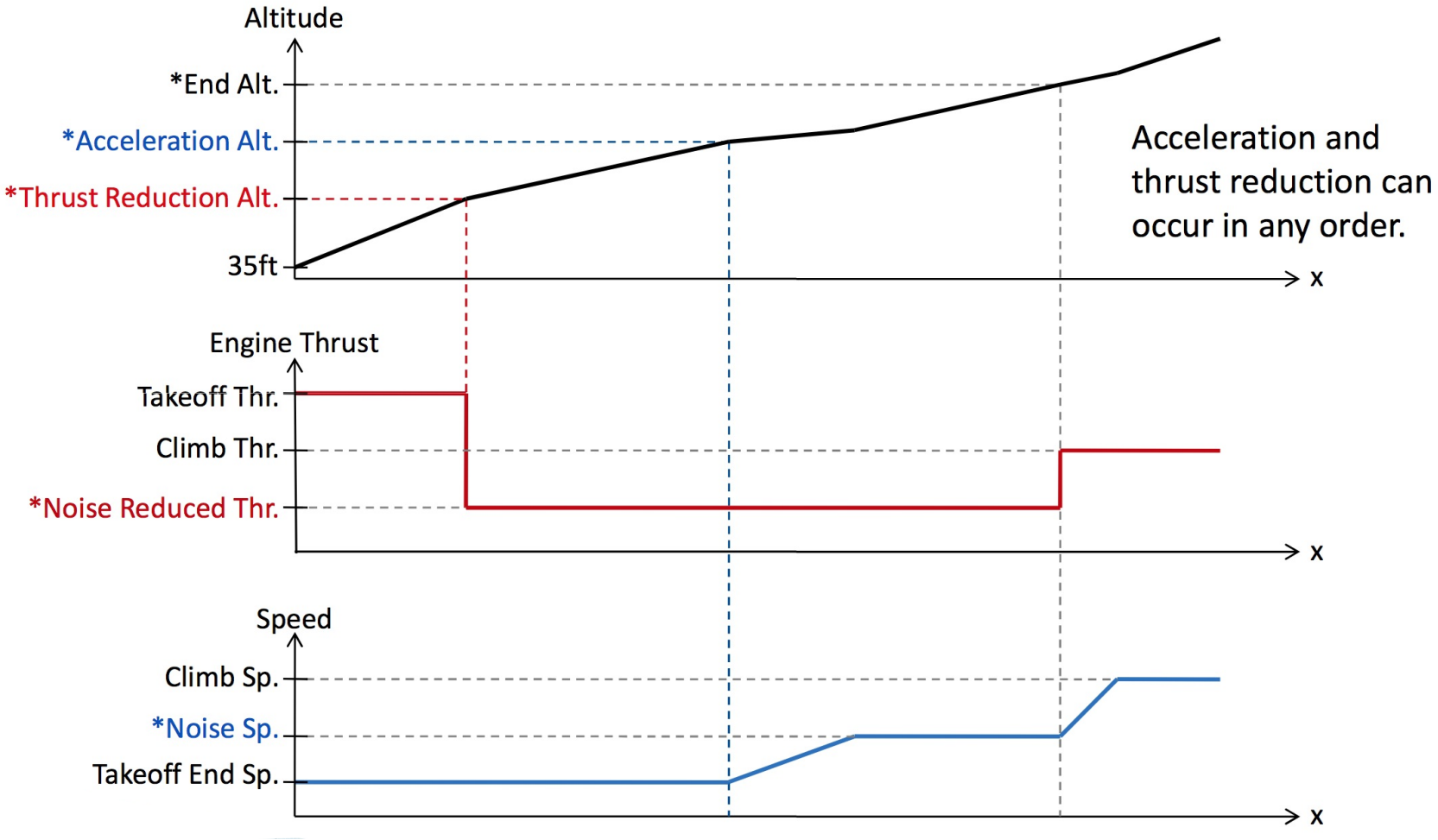
Aircraft takeoff trajectories

- AIRBUS problem involving (among others): O. Babando, C. Bes, J. Chaptal, J.-B. Hiriart-Urruty, B. Talgorn, B. Tessier, and R. Torres.
- Motivations for MADS and NOMAD:
  - A blackbox is involved.
  - Biobjective optimization.
  - Free software.
  - Must execute on different platforms including some old Solaris distributions.

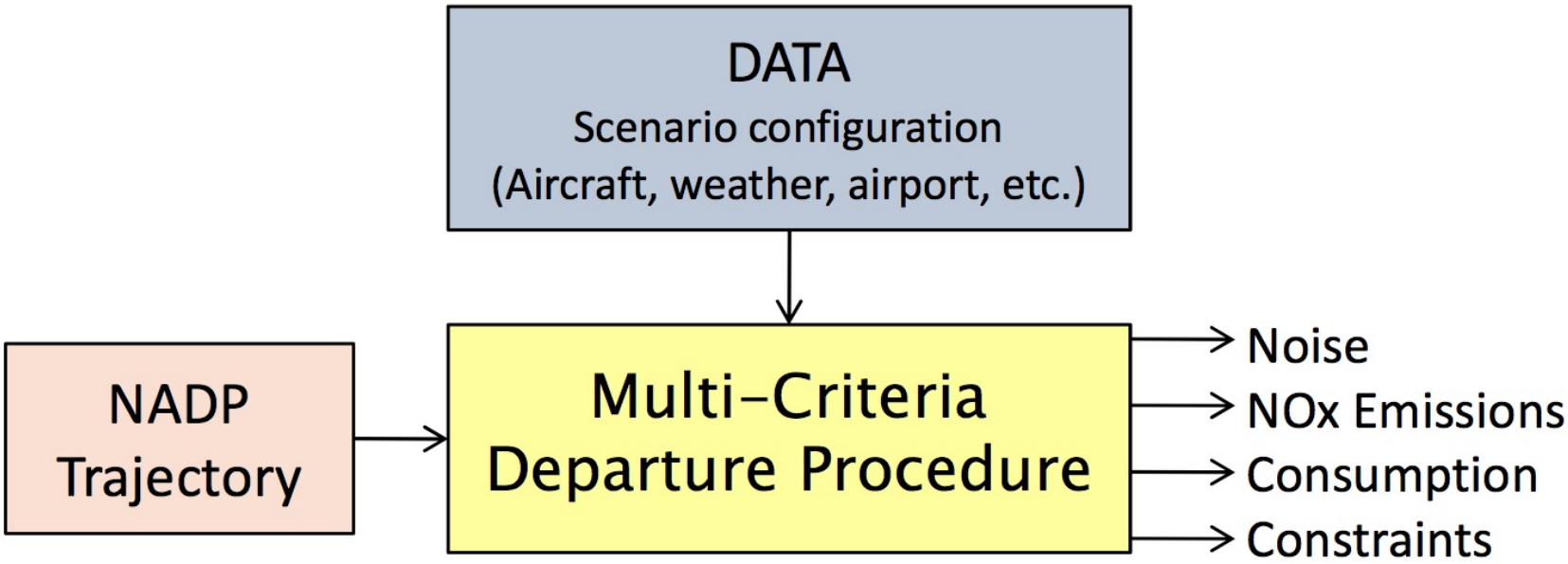
Definition of the optimization problem

- Concept : Optimization of vertical flight path based on procedures designed to reduce noise emission at departure to protect airport vicinity.
- Minimization of environmental and economical impact: Noise and fuel consumption.
- NADP (Noise Abatement Departure Procedure), variables: During departure phase, the aircraft will target its climb configuration:
  - Increase the speed up to climb speed (acceleration phase).
  - Reduce the engine rate to climb thrust (reduction phase).
  - Gain altitude.

Parametric Trajectory: 5 optimization variables (*)



The blackbox: MCDP: Multi-Criteria Departure Procedure



One evaluation $\simeq$ 2 seconds.

Special features

- The best trajectory parameters are returned to the pilot who enters them in the aircraft system manually.
- Finite precision on optimization parameters: Discretization of optimization variables (100 to 1000 different values for each parameter).
- The variables have been defined as integers in `NOMAD` (minimum mesh size of 1 and rounding of directions).

Results

Detailed results are confidential. But we can say:

- Tested for the Munich airport.
- Aircraft: A321.
- $\simeq$ 3000 evaluations for $\simeq$ 30 undominated points.

Discussion

- Description of the MADS algorithm for blackbox optimization.
- Focus on constraints handling, biobjective optimization, and on the use of models and surrogates.
- Three different engineering applications. There are others (alloys, metamaterials, bioinformatics, etc.)
- Many special features of MADS and NOMAD have been exploited. The algorithm and the code are robust and mature enough to adapt to many different situations.
- NOMAD is now widely spread and used in industry.

- C. Audet and J. E. Dennis, Jr.: *Mesh adaptive direct search algorithms for constrained optimization.* SIAM Journal on Optimization, 2006.
- C. Audet and J. E. Dennis, Jr.: *A progressive barrier for derivative-free nonlinear programming.* SIAM Journal on Optimization, 2009.
- C. Audet, G. Savard and W. Zghal: *Multiobjective optimization through a series of single-objective formulations.* SIAM Journal on Optimization, 2008.
- C. Audet, J.E. Dennis, Jr., and S. Le Digabel: *Trade-off studies in blackbox optimization.* Optimization Methods and Software, 2012.
- C. Audet, V. Béchard, and S. Le Digabel: *Nonsmooth Optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search.* Journal of Global Optimization, 2008.
- S. Le Digabel: *Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm.* ACM Transactions on Mathematical Software, 2011.
- A.R. Conn and S. Le Digabel: *Use of quadratic models with mesh-adaptive direct search for constrained black box optimization.* Optimization Methods and Software, 2013.
- M.A. Taddy, R.B. Gramacy, and N.G. Polson: *Dynamic trees for learning and design.* Journal of the American Statistical Association, 2011.

- S. Alarie, C. Audet, V. Garnier, S. Le Digabel, and L.A. Leclaire: *Snow water equivalent estimation using blackbox optimization.* Pacific Journal of Optimization, 2013.
- M. Minville, D. Cartier, C. Guay, L.-A. Leclaire, C. Audet, S. Le Digabel, and J. Merleau: *New calibration approaches for conceptual hydrological models: incorporating external information for oriented parametrisations based on physical processes.* In revision for Water Resources Research.
- R. Torres, J. Chaptal, C. Bès, and J.-B. Hiriart-Urruty: *Optimal, Environmentally Friendly Departure Procedures for Civil Aircraft.* Journal of Aircraft, 2011.
- A.E. Gheribi, C. Robelin, S. Le Digabel, C. Audet, and A.D. Pelton: *Calculating all local minima on liquidus surfaces using the FactSage software and databases and the Mesh Adaptive Direct Search algorithm.* The Journal of Chemical Thermodynamics, 2011.
- E.M. Gertz, T. Hiekkalinna, S. Le Digabel, C. Audet, J.D. Terwilliger, and A.A. Schaffer: *PSEUDOMARKER 2.0: efficient computation of likelihoods using NOMAD.* To appear in BMC Bioinformatics.
- C. Audet, K. Diest, S. Le Digabel, L.A. Sweatlock, and D.E. Marthaler: *Metamaterial Design by Mesh Adaptive Direct Search.* Numerical Methods for Metamaterial Design, Volume 127 of Topics in Applied Physics , p. 71-96, Springer, 2013.