Opportunism and ordering strategies in derivative-free optimization

Loïc Anthony Sarrazin-Mc Cann
Charles Audet, Sébastien Le Digabel and Christophe Tribes

Department of Mathematics and Industrial Engineering
École Polytechnique de Montréal

GERAD

Presentation Outline

1. What are Derivative-Free and Blackbox Optimization?
2. Direct-search Methods
3. Opportunism and ordering
4. Numerical results

Derivative-Free and Blackbox Optimization

Optimization problem :

$$\begin{cases} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ \text{s.à.} & c_j(x) \leq 0 \quad \forall j \in \{1, \dots, m\} \\ & l_i \leq x_i \leq u_i \quad \forall i \in \{1, \dots, n\} \end{cases}$$

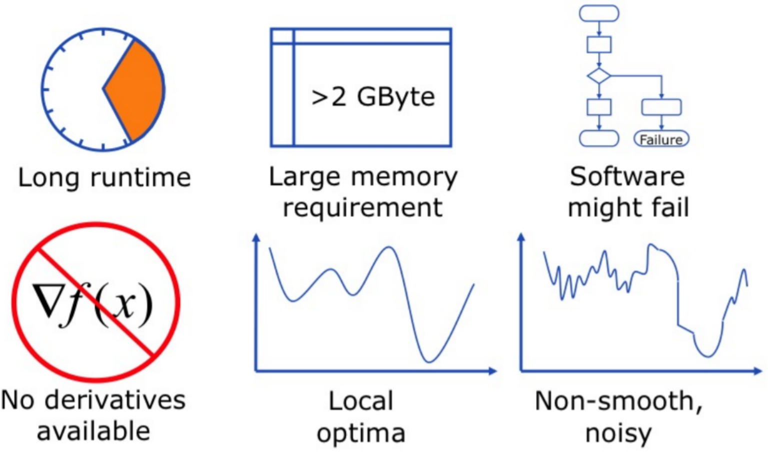
Derivative-Free and Blackbox Optimization

Optimization problem :

$$\begin{cases} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ \text{s.à.} & c_j(x) \leq 0 \ \ \forall j \in \{1, \ldots, m\} \\ & l_i \leq x_i \leq u_i \ \ \forall i \in \{1, \ldots, n\} \end{cases}$$

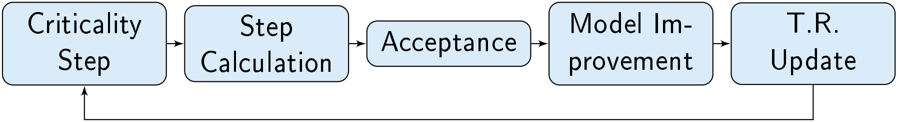
- $f(x)$ and $c_j(x)$ are treated as blackboxes.

Blackbox


>2 GByte
Failure
Long runtime
Large memory requirement
Software might fail
$\nabla f(x)$
No derivatives available
Local optima
Non-smooth, noisy

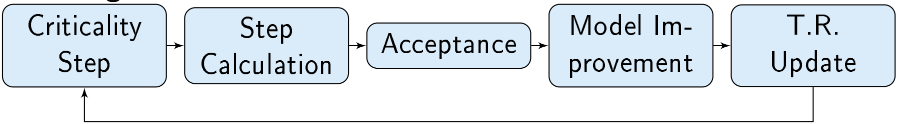
DFO and BBO methods

Trust Region Methods

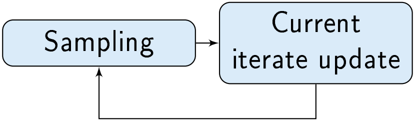


DFO and BBO methods

Trust Region Methods

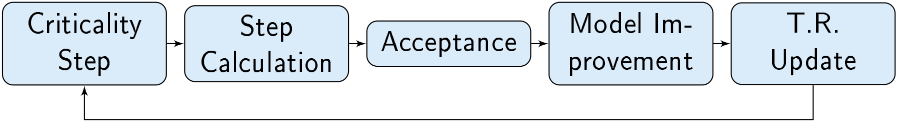


Direct-search Methods

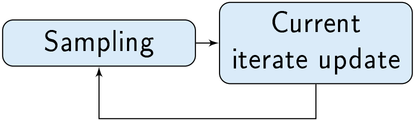


DFO and BBO methods

Trust Region Methods



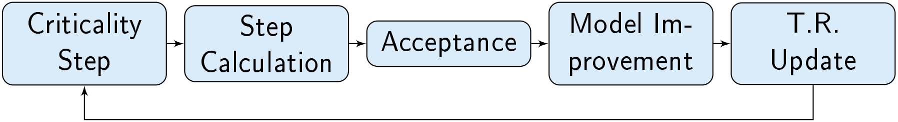
Direct-search Methods



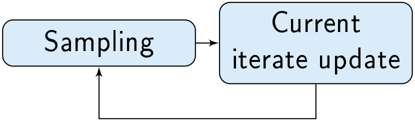
Directional (MADS)

DFO and BBO methods

Trust Region Methods



Direct-search Methods

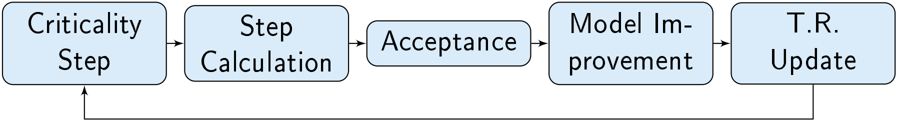


Directional (MADS)

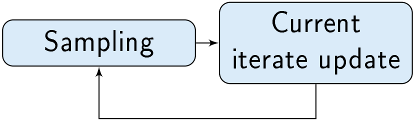
Simplicial (Nelder-Mead)

DFO and BBO methods

Trust Region Methods



Direct-search Methods



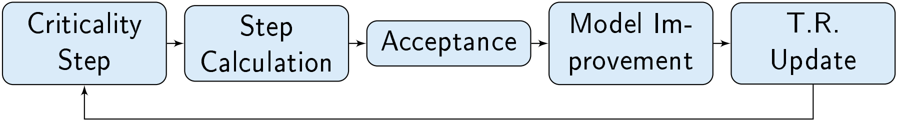
Directional (MADS)

Simplicial (Nelder-Mead)

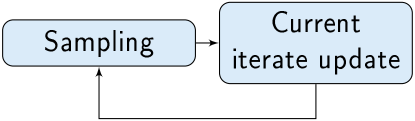
Other methods

DFO and BBO methods

Trust Region Methods



Direct-search Methods



Directional (MADS)

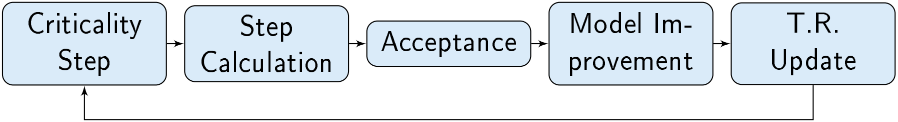
Simplicial (Nelder-Mead)

Other methods

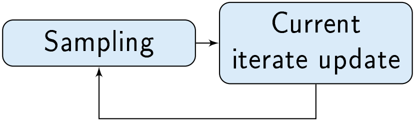
Heuristics (Particle Swarm, Simulated Annealing)

DFO and BBO methods

Trust Region Methods



Direct-search Methods



Directional (MADS)

Simplicial (Nelder-Mead)

Other methods

Heuristics (Particle Swarm, Simulated Annealing)

Hybrids (Implicit Filtering)

What Are We Trying to Achieve?

Our goal : reduce the amount of calls to the blackbox

What Are We Trying to Achieve?

Our goal : reduce the amount of calls to the blackbox

To do so, we consider the **Opportunistic Strategy**.

What Are We Trying to Achieve?

Our goal : reduce the amount of calls to the blackbox

To do so, we consider the **Opportunistic Strategy**.

Opportunistic Strategy

The opportunistic strategy designates the premature termination of an algorithmic step as soon as the necessary conditions to proceed to the next step are met.

What Are We Trying to Achieve?

Our goal : reduce the amount of calls to the blackbox

To do so, we consider the **Opportunistic Strategy**.

Opportunistic Strategy

The opportunistic strategy designates the premature termination of an algorithmic step as soon as the necessary conditions to proceed to the next step are met.

Often mentionned but never studied per se.

Identifying suitable methods

Question 1.

For which methods is the opportunistic strategy applicable?

Identifying suitable methods

Question 1.

For which methods is the opportunistic strategy applicable?

Trust Region

Identifying suitable methods

Question 1.

For which methods is the opportunistic strategy applicable?

Trust Region ✗

- Only one call to the blackbox per iteration of the method

Identifying suitable methods

Question 1.

For which methods is the opportunistic strategy applicable?

Trust Region ✗

- Only one call to the blackbox per iteration of the method

Simplicial direct-search methods

Identifying suitable methods

Question 1.

For which methods is the opportunistic strategy applicable?

Trust Region ✗

- Only one call to the blackbox per iteration of the method

Simplicial direct-search methods ✗

- Already and inevitably opportunistic

Identifying suitable methods

Question 1.
For which methods is the opportunistic strategy applicable?

Trust Region ✗

- Only one call to the blackbox per iteration of the method

Simplicial direct-search methods ✗

- Already and inevitably opportunistic

Directional direct-search methods

Identifying suitable methods

Question 1.

For which methods is the opportunistic strategy applicable?

Trust Region ✗

- Only one call to the blackbox per iteration of the method

Simplicial direct-search methods ✗

- Already and inevitably opportunistic

Directional direct-search methods ✓

- Allows opportunistic termination.

Identifying suitable methods

Question 1.

For which methods is the opportunistic strategy applicable?

Trust Region ✗

- Only one call to the blackbox per iteration of the method

Simplicial direct-search methods ✗

- Already and inevitably opportunistic

Directional direct-search methods ✓

- Allows opportunistic termination.

Hybrid directional direct-search methods

Identifying suitable methods

Question 1.

For which methods is the opportunistic strategy applicable?

Trust Region ✗

- Only one call to the blackbox per iteration of the method

Simplicial direct-search methods ✗

- Already and inevitably opportunistic

Directional direct-search methods ✓

- Allows opportunistic termination.

Hybrid directional direct-search methods ✓

- Same as direct-search methods, but might impact performance.

Identifying suitable methods

Question 1.

For which methods is the opportunistic strategy applicable?

Trust Region ✗

- Only one call to the blackbox per iteration of the method

Simplicial direct-search methods ✗

- Already and inevitably opportunistic

Directional direct-search methods ✓

- Allows opportunistic termination.

Hybrid directional direct-search methods ✓

- Same as direct-search methods, but might impact performance.

Heuristics

Identifying suitable methods

Question 1.

For which methods is the opportunistic strategy applicable?

Trust Region ✗

- Only one call to the blackbox per iteration of the method

Simplicial direct-search methods ✗

- Already and inevitably opportunistic

Directional direct-search methods ✓

- Allows opportunistic termination.

Hybrid directional direct-search methods ✓

- Same as direct-search methods, but might impact performance.

Heuristics ?

- Highly dependent on the heuristic.

Direcitonal direct-search framework

Directional direct-search methods :

Direcitonal direct-search framework

Directional direct-search methods :

- Samples $f(x)$ and $c(x)$ on $\mathcal{L}^k$, a list of point specific to the k^{th} iteration.

Direcitonal direct-search framework

Directional direct-search methods :

- Samples $f(x)$ and $c(x)$ on $\mathcal{L}^k$, a list of point specific to the k^{th} iteration.
- Generates the next list of candidates $\mathcal{L}^{k+1}$ based on these values.

Direcitonal direct-search framework

Directional direct-search methods :

- Samples $f(x)$ and $c(x)$ on $\mathcal{L}^k$, a list of point specific to the k^{th} iteration.
- Generates the next list of candidates $\mathcal{L}^{k+1}$ based on these values.

Algorithm 1 Direcitonal direct-search framework

for $k = 1, 2, \ldots$ **do**
  Search : Evaluate $f(x)$ at a finite set of point S^k.
  If successful, update x^k

  Poll : Evaluate $f(x)$ at a finite set of point
  $P^k := \{x^k + \delta^k d : d \in D\}$, where D is a positive spanning set of **directions**.
  If successful, update x^k and mesh parameters.
end for

Remark : this work only encompasses the study of the impact of the opportunistic strategy on poll steps.

Coordinate Search (CS)

Algorithm 2 Coordinate Search

for $k = 1, 2, \ldots$ **do**

  Poll : Evaluate $f(x)$ at
  $P^k := \{x^k + \delta^k d : d \in D_\oplus\}$, where
  $D_\oplus := \{\pm e_1, \pm e_2, \ldots, \pm e_n\}$.

  If $\exists\ t$ for which $f(t) < f(x^k)$, $t \in P^k$
  Successful step
  update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \delta^k$.

  Else $\nexists\ t$ for which $f(t) < f(x^k)$, $t \in P^k$
  Unsuccessful step
  update $x^{k+1} \leftarrow x^k$ et $\delta^{k+1} \leftarrow \frac{\delta^k}{2}$.

end for

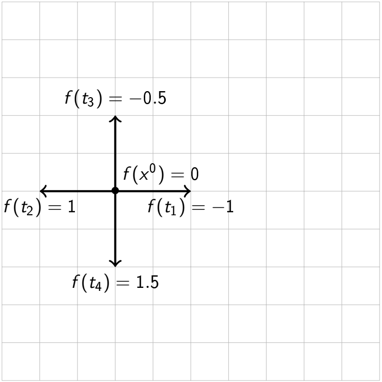


Figure: CS

Coordinate Search (CS)

Algorithm 2 Coordinate Search

for $k = 1, 2, \dots$ **do**

Poll : Evaluate $f(x)$ at
$P^k := \{x^k + \delta^k d : d \in D_\oplus\}$, where
$D_\oplus := \{\pm e_1, \pm e_2, \dots, \pm e_n\}$.

If $\exists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Successful step
update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \delta^k$.

Else $\nexists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Unsuccessful step
update $x^{k+1} \leftarrow x^k$ et $\delta^{k+1} \leftarrow \frac{\delta^k}{2}$.

end for

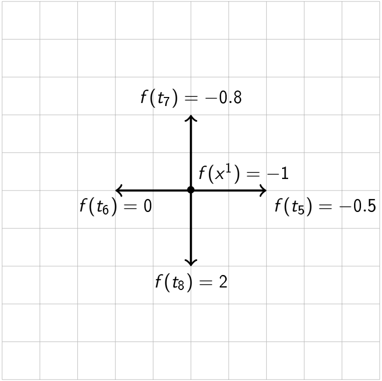


Figure: CS

Coordinate Search (CS)

Algorithm 2 Coordinate Search

for $k = 1, 2, \ldots$ **do**

Poll : Evaluate $f(x)$ at
$P^k := \{x^k + \delta^k d : d \in D_\oplus\}$, where
$D_\oplus := \{\pm e_1, \pm e_2, \ldots, \pm e_n\}$.

If $\exists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Successful step
update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \delta^k$.

Else $\nexists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Unsuccessful step
update $x^{k+1} \leftarrow x^k$ et $\delta^{k+1} \leftarrow \frac{\delta^k}{2}$.

end for

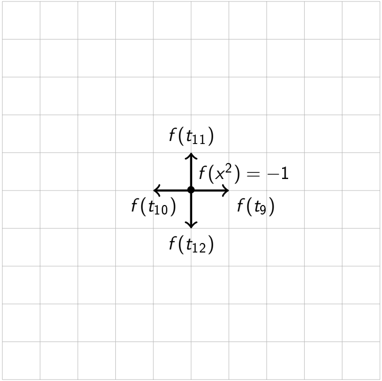


Figure: CS

Generalized Pattern Search (GPS)

Algorithm 3 Generalized Pattern Search

for $k = 1, 2, \dots$ **do**

with $\tau \in \{0, 1\}$.

Poll : Evaluate $f(x)$ at $P^k := \{x^k + \delta^k d : d \in D\}$, where D is a positive spanning set.

If $\exists\ t$ for which $f(t) < f(x^k)$, $t \in P^k$

Successful step

update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \tau^{-1}\delta^k$.

Else $\nexists\ t$ for which $f(t) < f(x^k)$, $t \in P^k$

Unsuccessful step

update $x^{k+1} \leftarrow x^k$ and $\delta^{k+1} \leftarrow \tau\delta^k$.

end for

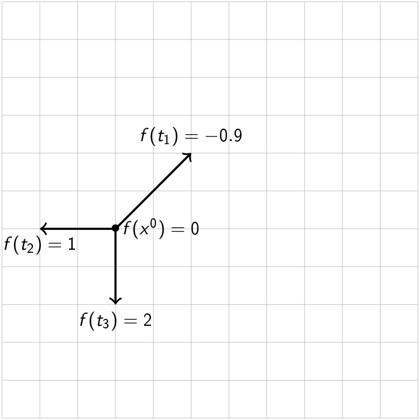


Figure: GPS

Generalized Pattern Search (GPS)

Algorithm 4 Generalized Pattern Search

for $k = 1, 2, \dots$ **do**
with $\tau \in \{0, 1\}$.
Poll : Evaluate $f(x)$ at
$P^k := \{x^k + \delta^k d : d \in D\}$, where
D is a positive spanning set.

If $\exists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Successful step
update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \tau^{-1}\delta^k$.

Else $\nexists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Unsuccessful step
update $x^{k+1} \leftarrow x^k$ and $\delta^{k+1} \leftarrow \tau\delta^k$.
end for

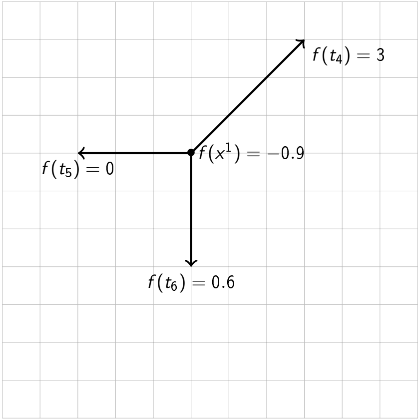


Figure: GPS

Generalized Pattern Search (GPS)

Algorithm 4 Generalized Pattern Search

for $k = 1, 2, \dots$ **do**
with $\tau \in \{0, 1\}$.
Poll : Evaluate $f(x)$ at $P^k := \{x^k + \delta^k d : d \in D\}$, where D is a positive spanning set.

If $\exists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Successful step
update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \tau^{-1}\delta^k$.

Else $\nexists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Unsuccessful step
update $x^{k+1} \leftarrow x^k$ and $\delta^{k+1} \leftarrow \tau\delta^k$.
end for

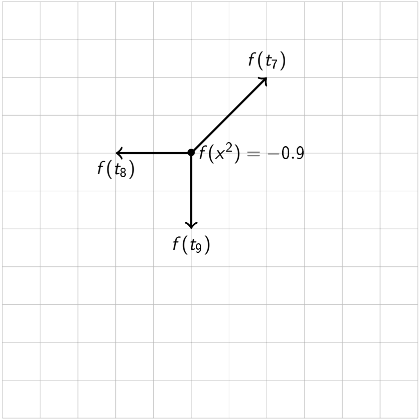


Figure: GPS

Generating Set Search (GSS)

Algorithm 4 Generating Set Search

for $k = 1, 2, \dots$ **do**
with $\tau \in \{0, 1\}$.
Poll : Evaluate $f(x)$ at
$P^k := \{x^k + \delta^k d : d \in D\}$, where
D is a positive spanning set
respecting multiple conditions.

If $\exists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Successful step
update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \phi\delta^k$.

Else $\nexists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Unsuccessful step
update $x^{k+1} \leftarrow x^k$ and $\delta^{k+1} \leftarrow \tau\delta^k$.
end for

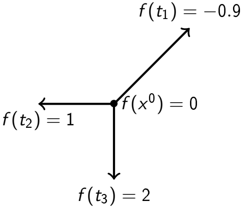


Figure: GSS

Generating Set Search (GSS)

Algorithm 4 Generating Set Search

for $k = 1, 2, \dots$ **do**
with $\tau \in \{0, 1\}$.
Poll : Evaluate $f(x)$ at
$P^k := \{x^k + \delta^k d : d \in D\}$, where
D is a positive spanning set
respecting multiple conditions.

If $\exists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Successful step
update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \phi\delta^k$.

Else $\nexists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Unsuccessful step
update $x^{k+1} \leftarrow x^k$ and $\delta^{k+1} \leftarrow \tau\delta^k$.
end for

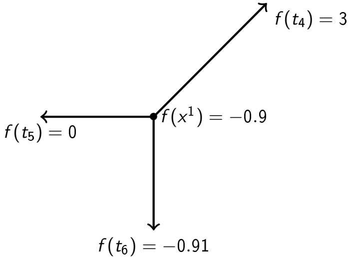


Figure: GSS

Generating Set Search (GSS)

Algorithm 4 Generating Set Search

for $k = 1, 2, \ldots$ **do**
  with $\tau \in \{0, 1\}$.
  Poll : Evaluate $f(x)$ at
  $P^k := \{x^k + \delta^k d : d \in D\}$, where
  D is a positive spanning set
  respecting multiple conditions.

  If $\exists\ t$ for which $f(t) < f(x^k)$, $t \in P^k$
  Successful step
  update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \phi\delta^k$.

  Else $\nexists\ t$ for which $f(t) < f(x^k)$, $t \in P^k$
  Unsuccessful step
  update $x^{k+1} \leftarrow x^k$ and $\delta^{k+1} \leftarrow \tau\delta^k$.
end for

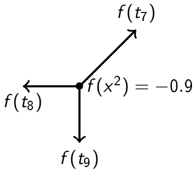


Figure: GSS

Mesh Adaptive Direct Search (MADS)

Algorithm 5 Mesh Adaptive Direct Search

for $k = 1, 2, \ldots$ **do**
with $\tau \in \{0, 1\}$.
Update : $\delta^k \leftarrow \min(\Delta^k, (\Delta^k)^2)$
Poll : Evaluate $f(x)$ at
$P^k := \{x^k + \delta^k d : d \in D\}$, where
$D \subset F^k$, with F^k frame of size Δ^k.

If $\exists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Successful step
update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \phi\delta^k$.

Else $\nexists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Unsuccessful step
update $x^{k+1} \leftarrow x^k$ and $\delta^{k+1} \leftarrow \tau\delta^k$.
end for

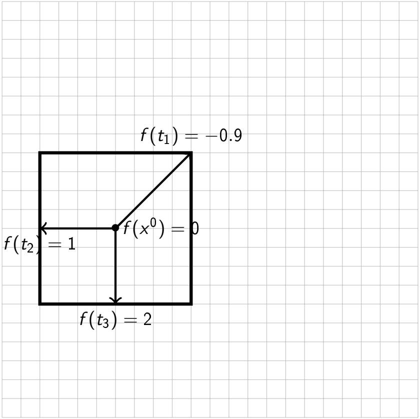


Figure: MADS

Mesh Adaptive Direct Search (MADS)

Algorithm 5 Mesh Adaptive Direct Search

for $k = 1, 2, \dots$ **do**
  with $\tau \in \{0, 1\}$.
  Update : $\delta^k \leftarrow \min(\Delta^k, (\Delta^k)^2)$
  Poll : Evaluate $f(x)$ at
  $P^k := \{x^k + \delta^k d : d \in D\}$, where
  $D \subset F^k$, with F^k frame of size Δ^k.

  If $\exists$ t for which $f(t) < f(x^k)$, $t \in P^k$
  Successful step
  update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \phi\delta^k$.

  Else $\nexists$ t for which $f(t) < f(x^k)$, $t \in P^k$
  Unsuccessful step
  update $x^{k+1} \leftarrow x^k$ and $\delta^{k+1} \leftarrow \tau\delta^k$.
end for

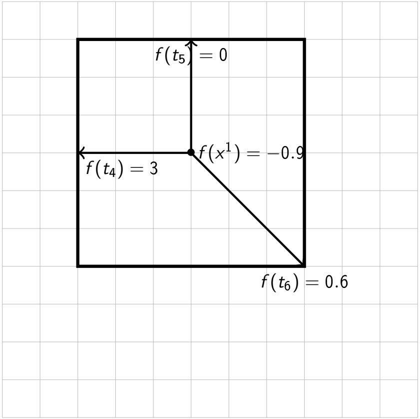


Figure: MADS

Mesh Adaptive Direct Search (MADS)

Algorithm 5 Mesh Adaptive Direct Search

for $k = 1, 2, \ldots$ **do**
with $\tau \in \{0, 1\}$.
Update : $\delta^k \leftarrow \min(\Delta^k, (\Delta^k)^2)$
Poll : Evaluate $f(x)$ at
$P^k := \{x^k + \delta^k d : d \in D\}$, where
$D \subset F^k$, with F^k frame of size Δ^k.

If $\exists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Successful step
update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \phi\delta^k$.

Else $\nexists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Unsuccessful step
update $x^{k+1} \leftarrow x^k$ and $\delta^{k+1} \leftarrow \tau\delta^k$.
end for

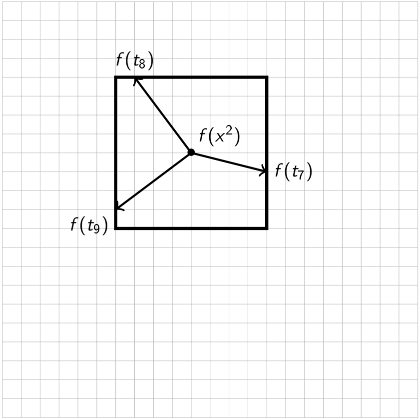


Figure: MADS

Implicit Filtering (IMFIL)

Algorithm 6 Implicit Filtering

for $k = 1, 2, \ldots$ **do**

Poll : Evaluate $f(x)$ at $P^k := \{x^k + \delta^k d : d \in D_\oplus\}$, where $D_\oplus := \{\pm e_1, \pm e_2, \ldots, \pm e_n\}$.

If $\exists\ t$ for which $f(t) < f(x^k)$, $t \in P^k$
Successful step
Line search following $-\nabla_s f(x^k)$.
update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \delta^k$.

Else $\nexists\ t$ for which $f(t) < f(x^k)$, $t \in P^k$
Unsuccessful step
update $x^{k+1} \leftarrow x^k$ et $\delta^{k+1} \leftarrow \frac{\delta^k}{2}$.

end for

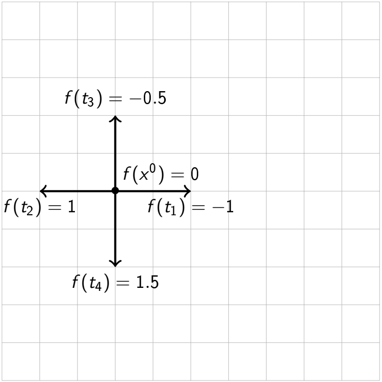


Figure: IMFIL

Implicit Filtering (IMFIL)

Algorithm 6 Implicit Filtering

for $k = 1, 2, \ldots$ **do**

Poll : Evaluate $f(x)$ at $P^k := \{x^k + \delta^k d : d \in D_\oplus\}$, where $D_\oplus := \{\pm e_1, \pm e_2, \ldots, \pm e_n\}$.

If $\exists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Successful step
Line search following $-\nabla_s f(x^k)$.
update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \delta^k$.

Else $\nexists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Unsuccessful step
update $x^{k+1} \leftarrow x^k$ et $\delta^{k+1} \leftarrow \frac{\delta^k}{2}$.

end for

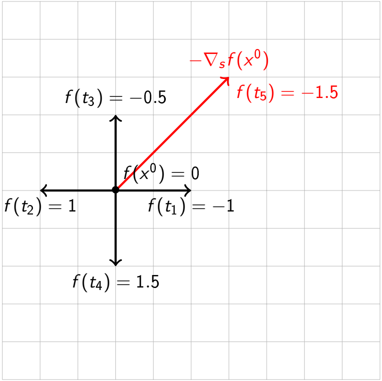


Figure: IMFIL

Implicit Filtering (IMFIL)

Algorithm 6 Implicit Filtering

for $k = 1, 2, \ldots$ **do**

Poll : Evaluate $f(x)$ at
$P^k := \{x^k + \delta^k d : d \in D_\oplus\}$, where
$D_\oplus := \{\pm e_1, \pm e_2, \ldots, \pm e_n\}$.

If $\exists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Successful step
Line search following $-\nabla_s f(x^k)$.
update $x^{k+1} \leftarrow t$ et $\delta^{k+1} \leftarrow \delta^k$.

Else $\nexists$ t for which $f(t) < f(x^k)$, $t \in P^k$
Unsuccessful step
update $x^{k+1} \leftarrow x^k$ et $\delta^{k+1} \leftarrow \frac{\delta^k}{2}$.

end for

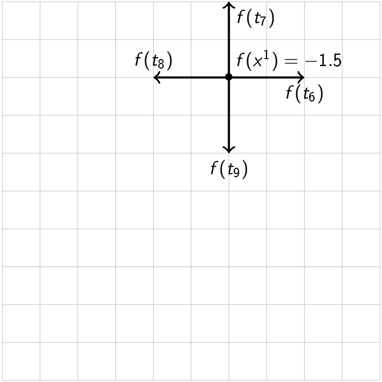


Figure: IMFIL

Opportunistic strategies

Complete polling

Designates the evaluation of $f(x)$ and $c(x)$ at every point generated in the poll step.

Opportunistic strategies

Complete polling

Designates the evaluation of $f(x)$ and $c(x)$ at every point generated in the poll step.

Simple opportunistic strategy

Designates the opportunistic termination of the poll after **a single successful point**

Opportunistic strategies

Complete polling

Designates the evaluation of $f(x)$ and $c(x)$ at every point generated in the poll step.

Simple opportunistic strategy

Designates the opportunistic termination of the poll after **a single successful point**

Opportunistic strategy after p success

Opportunistic termination of the poll following the evaluation of p successful points.

Opportunistic strategies

Complete polling

Designates the evaluation of $f(x)$ and $c(x)$ at every point generated in the poll step.

Simple opportunistic strategy

Designates the opportunistic termination of the poll after **a single successful point**

Opportunistic strategy after p success

Opportunistic termination of the poll following the evaluation of p successful points.

Opportunistic strategy after q evaluations

Opportunistic termination of the poll following the evaluations of q points.

Ordering strategies

Ordering strategy

Rule guiding the ordering of points in a set $\mathcal{L}$.

1. **Lexicographic**

Ordering strategies

Ordering strategy

Rule guiding the ordering of points in a set $\mathcal{L}$.

1. **Lexicographic**

   Ordered as in a dictionary, i.e. $(0,0,1) \prec (0,0,3) \prec (0,1,0)$.

Ordering strategies

Ordering strategy

Rule guiding the ordering of points in a set $\mathcal{L}$.

1. **Lexicographic**

   Ordered as in a dictionary, i.e. $(0,0,1) \prec (0,0,3) \prec (0,1,0)$.
2. **Random**

Ordering strategies

Ordering strategy

Rule guiding the ordering of points in a set $\mathcal{L}$.

1. **Lexicographic**

   Ordered as in a dictionary, i.e. $(0,0,1) \prec (0,0,3) \prec (0,1,0)$.
2. **Random**
3. **Last success direction**

Ordering strategies

Ordering strategy

Rule guiding the ordering of points in a set $\mathcal{L}$.

1. **Lexicographic**

   Ordered as in a dictionary, i.e. $(0,0,1) \prec (0,0,3) \prec (0,1,0)$.
2. **Random**
3. **Last success direction**

   Ordered by the angle made with the last successful point's corresponding direction

Ordering strategies

Ordering strategy

Rule guiding the ordering of points in a set $\mathcal{L}$.

1. **Lexicographic**

   Ordered as in a dictionary, i.e. $(0,0,1) \prec (0,0,3) \prec (0,1,0)$.

2. **Random**
3. **Last success direction**

   Ordered by the angle made with the last successful point's corresponding direction

4. **Quadratic models**

Ordering strategies

Ordering strategy

Rule guiding the ordering of points in a set $\mathcal{L}$.

1. **Lexicographic**

   Ordered as in a dictionary, i.e. $(0,0,1) \prec (0,0,3) \prec (0,1,0)$.
2. **Random**
3. **Last success direction**

   Ordered by the angle made with the last successful point's corresponding direction
4. **Quadratic models**

   $A \prec B$ if $\tilde{f}(A) < \tilde{f}(B)$

Ordering strategies

Ordering strategy

Rule guiding the ordering of points in a set $\mathcal{L}$.

1. **Lexicographic**

   Ordered as in a dictionary, i.e. $(0,0,1) \prec (0,0,3) \prec (0,1,0)$.

2. **Random**

3. **Last success direction**

   Ordered by the angle made with the last successful point's corresponding direction

4. **Quadratic models**

   $A \prec B$ if $\tilde{f}(A) < \tilde{f}(B)$

   where $\tilde{f}(x)$ is a dynamic quadratic surrogate of $f(x)$.

Omniscient strategies

Use $f(x)$ as a surrogate for $f(x)$ to simulate the best ordering possible.

Omniscient strategies

Use $f(x)$ as a surrogate for $f(x)$ to simulate the best ordering possible.

5. **Omniscient**

Omniscient strategies

Use $f(x)$ as a surrogate for $f(x)$ to simulate the best ordering possible.

5. **Omniscient**

   $A \prec B$ if $f(A) < f(B)$

Omniscient strategies

Use $f(x)$ as a surrogate for $f(x)$ to simulate the best ordering possible.

5. **Omniscient**

   $A \prec B$ if $f(A) < f(B)$

Use $-f(x)$ as a surrogate for $f(x)$ to simulate the worst ordering possible.

Omniscient strategies

Use $f(x)$ as a surrogate for $f(x)$ to simulate the best ordering possible.

5. **Omniscient**

   $A \prec B$ if $f(A) < f(B)$

Use $-f(x)$ as a surrogate for $f(x)$ to simulate the worst ordering possible.

6. **Reverse Omniscient**

Omniscient strategies

Use $f(x)$ as a surrogate for $f(x)$ to simulate the best ordering possible.

5. **Omniscient**

   $A \prec B$ if $f(A) < f(B)$

Use $-f(x)$ as a surrogate for $f(x)$ to simulate the worst ordering possible.

6. **Reverse Omniscient**

   $A \prec B$ if $f(A) > f(B)$

Omniscient strategies

Use $f(x)$ as a surrogate for $f(x)$ to simulate the best ordering possible.

5. **Omniscient**

   $A \prec B$ if $f(A) < f(B)$

Use $-f(x)$ as a surrogate for $f(x)$ to simulate the worst ordering possible.

6. **Reverse Omniscient**

   $A \prec B$ if $f(A) > f(B)$

No practical use, for comparaison only.

Test problems

1. 212 instances of problems taken from [?]

Test problems

1. 212 instances of problems taken from [?]
2. 18 constrained problems taken from [?]

Test problems

1. 212 instances of problems taken from [?]
2. 18 constrained problems taken from [?]
   Infeasable starting point x^0

Test problems

1. 212 instances of problems taken from [?]
2. 18 constrained problems taken from [?]
   Infeasable starting point x^0
3. A blackbox taken from [?]

Test problems

1. 212 instances of problems taken from [?]
2. 18 constrained problems taken from [?]
   Infeasable starting point x^0
3. A blackbox taken from [?]
   $f : R^8 \mapsto R$, $c : R^8 \mapsto R^{11}$, 4 binary constraints, 7 relaxables constraints

Test problems

1. 212 instances of problems taken from [?]
2. 18 constrained problems taken from [?]
   Infeasable starting point x^0
3. A blackbox taken from [?]
   $f : R^8 \mapsto R$, $c : R^8 \mapsto R^{11}$, 4 binary constraints, 7 relaxables constraints

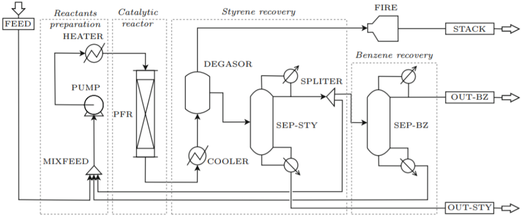


Figure: Styrene production chart [?]

Opportunistic strategies comparison

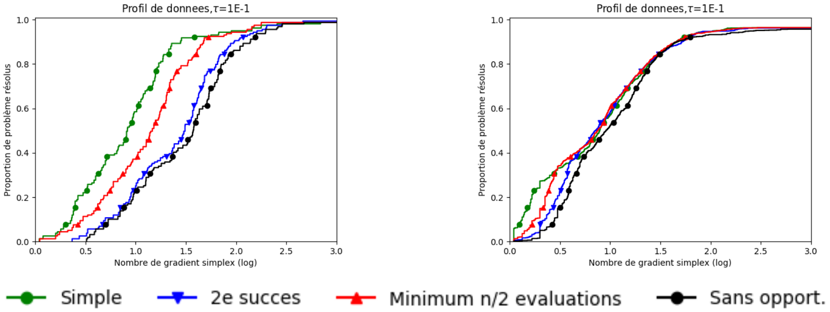


Figure: Left : CS on Moré-Wild, Right MADS on Moré-Wild

Opportunistic strategies comparison

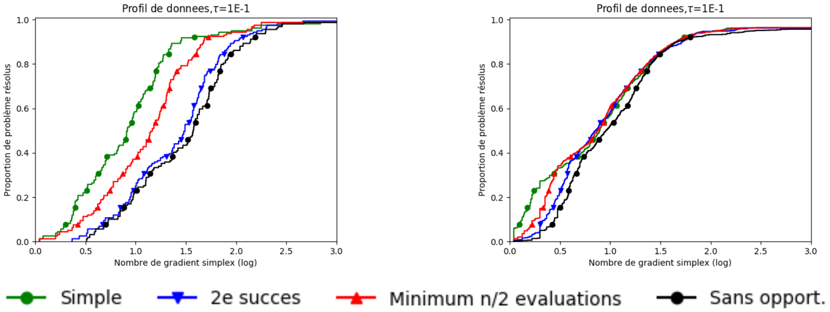


Figure: Left : CS on Moré-Wild, Right MADS on Moré-Wild

1. Model ordering used.

Opportunistic strategies comparison

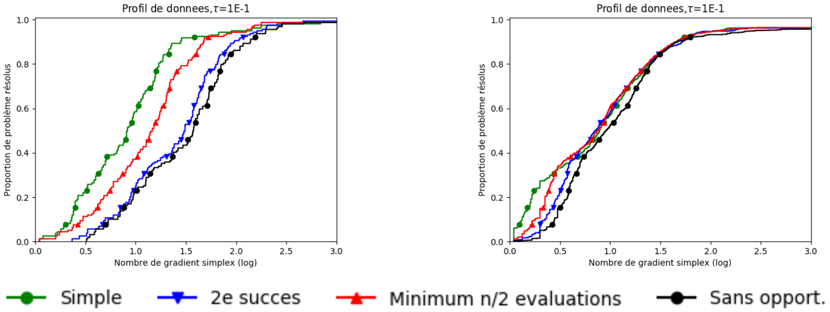


Figure: Left : CS on Moré-Wild, Right MADS on Moré-Wild

1. Model ordering used.
2. Most efficient opprtunistic strategy : simple

Opportunistic strategies comparison

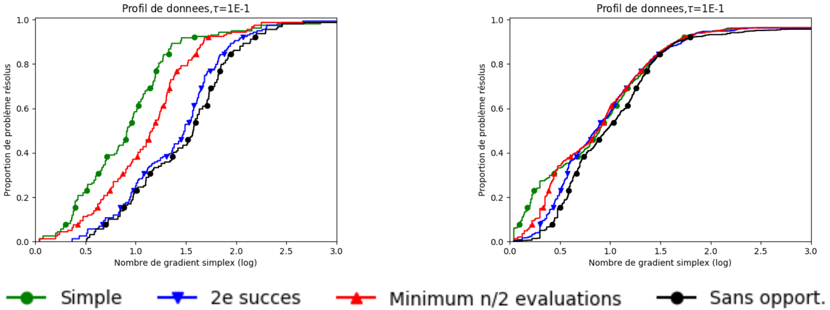


Figure: Left : CS on Moré-Wild, Right MADS on Moré-Wild

1. Model ordering used.
2. Most efficient opprtunistic strategy : simple
3. Impact less important on MADS.

Ordering strategies comparison

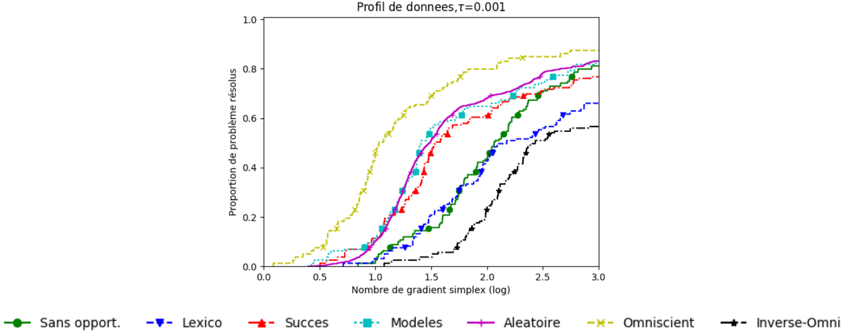


Figure: CS on Moré-Wild

Ordering strategies comparison

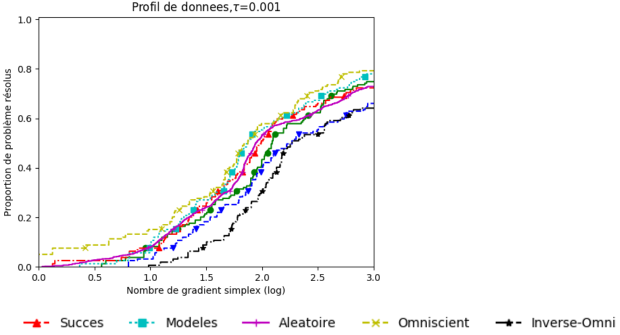


Figure: GPS on Moré-Wild

Ordering strategies comparison

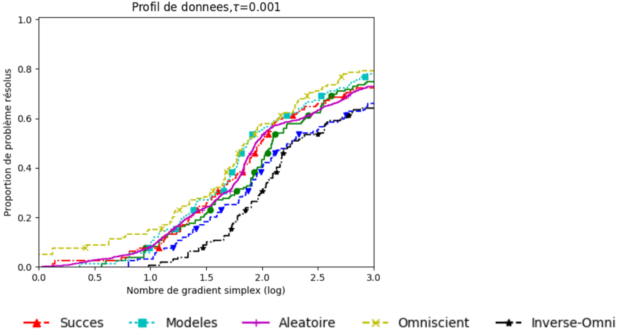


Figure: GPS on Moré-Wild

1. Omniscient strategy less imactful.

Ordering strategies comparison

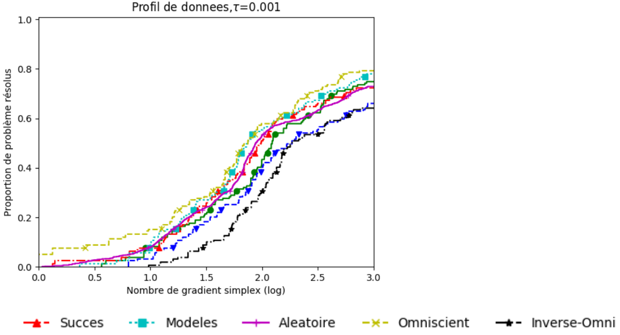


Figure: GPS on Moré-Wild

1. Omniscient strategy less imactful.
2. Model ordering less dominant