Are we Pushing Direct Search to its Limit?

Luís Nunes Vicente
University of Coimbra

(joint work with Ana Luísa Custódio, New Univ. Lisbon)

Feb. 3, 2010 — RTRA STAE Workshop, CERFACS, Toulouse

http//www.mat.uc.pt/~lnv

Direct-search methods

Definition

Direct-search methods sample the objective function at a finite number of points at each iteration and base actions on those function values and without any derivative approximation or model building.

Direct-search methods

Definition

Direct-search methods sample the objective function at a finite number of points at each iteration and base actions on those function values and without any derivative approximation or model building.

- Direct search of directional type: Achieve descent by using positive spanning sets and moving in the directions of the best points.

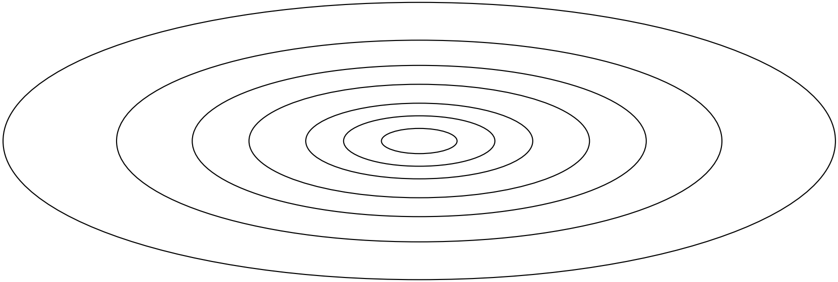
Direct-search methods

Definition

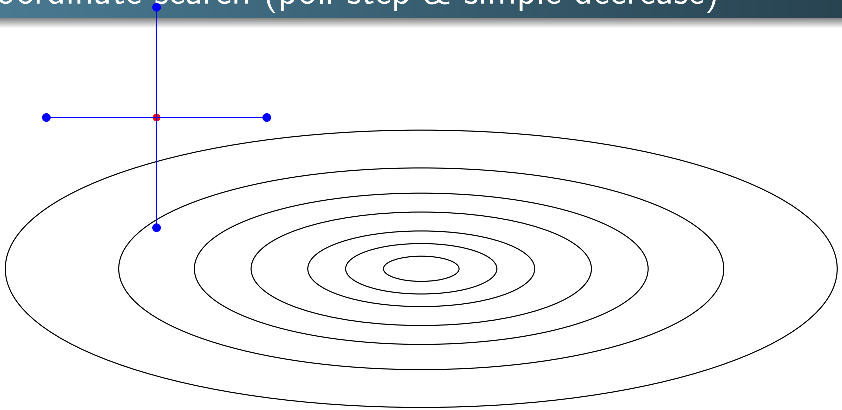
Direct-search methods sample the objective function at a finite number of points at each iteration and base actions on those function values and without any derivative approximation or model building.

- Direct search of directional type: Achieve descent by using positive spanning sets and moving in the directions of the best points.
- Direct search of simplicial type: e.g., Nelder-Mead based methods.

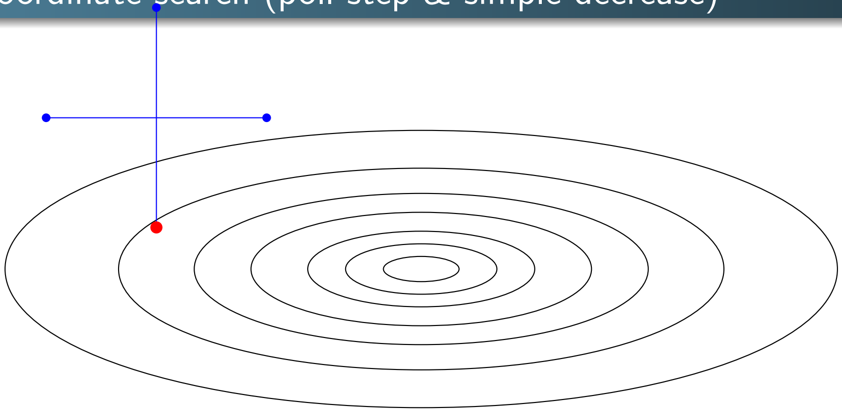
Coordinate search (poll step & simple decrease)



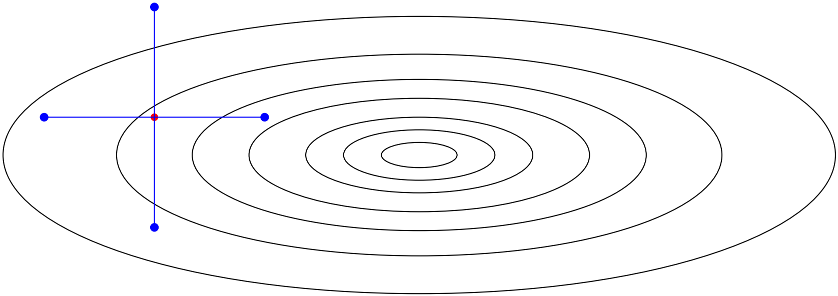
Coordinate search (poll step & simple decrease)



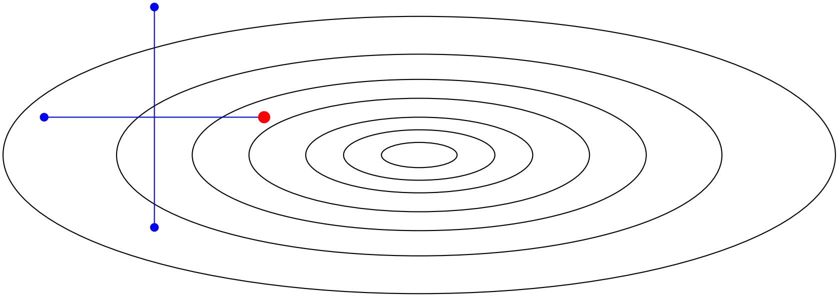
Coordinate search (poll step & simple decrease)



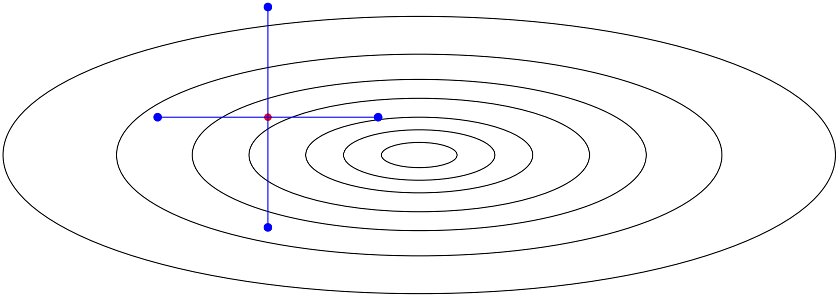
Coordinate search (poll step & simple decrease)



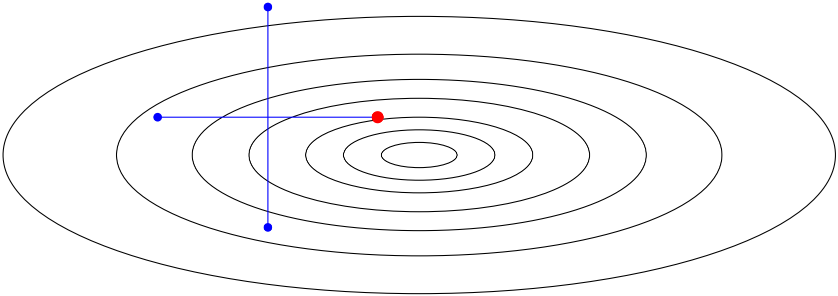
Coordinate search (poll step & simple decrease)



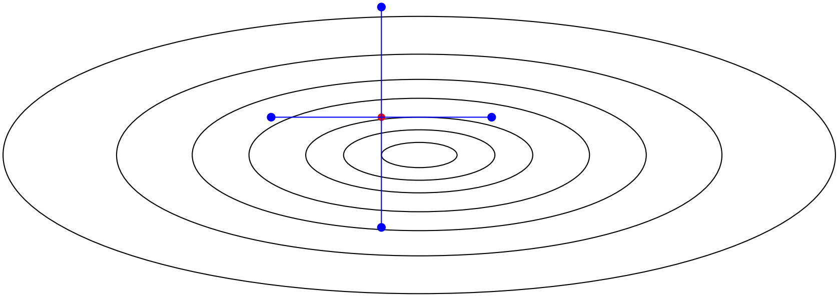
Coordinate search (poll step & simple decrease)



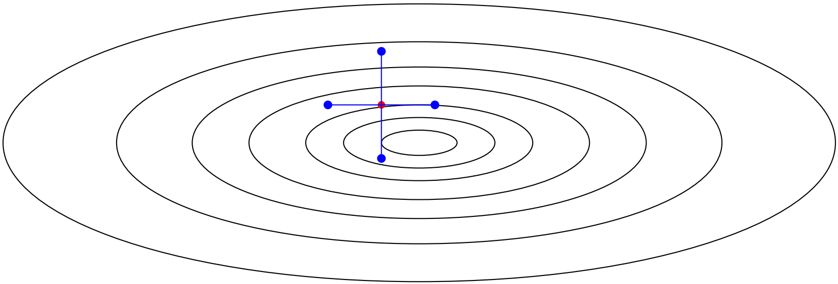
Coordinate search (poll step & simple decrease)



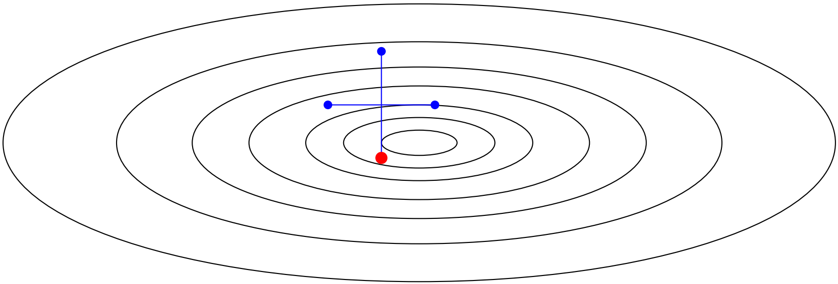
Coordinate search (poll step & simple decrease)



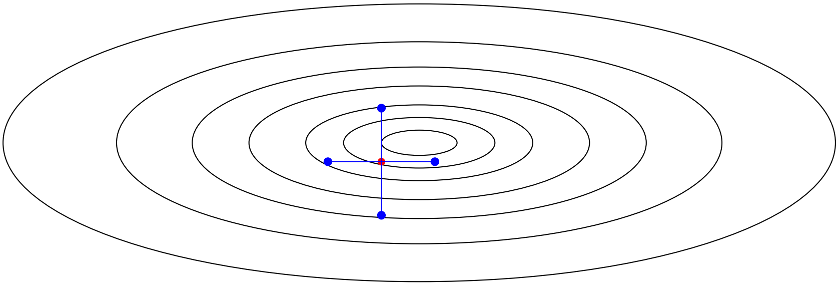
Coordinate search (poll step & simple decrease)

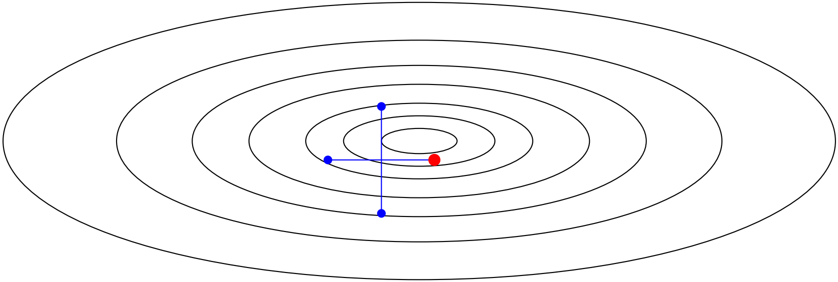


Coordinate search (poll step & simple decrease)

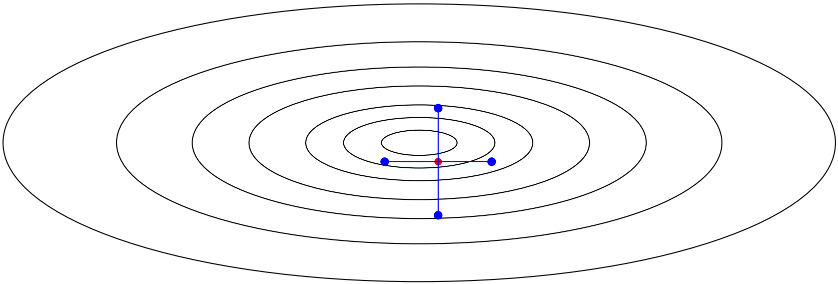


Coordinate search (poll step & simple decrease)

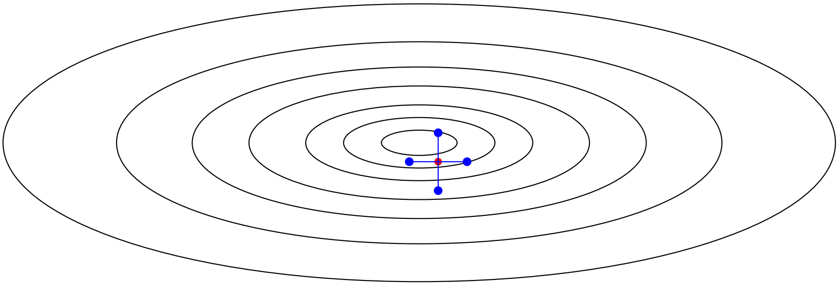




Coordinate search (poll step & simple decrease)



Coordinate search (poll step & simple decrease)



Problem setting and extreme barrier

In this talk, we consider a constrained minimization problem:

$$\begin{aligned} \min \quad & f(x), \\ \text{s.t.} \quad & x \in \Omega, \end{aligned}$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is nonsmooth and extended-real-valued and where $\Omega \subseteq \mathbb{R}^n$.

Problem setting and extreme barrier

In this talk, we consider a constrained minimization problem:

$$\min \quad f(x),$$
$$\text{s.t.} \quad x \in \Omega,$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is nonsmooth and extended-real-valued and where $\Omega \subseteq \mathbb{R}^n$.

We will make use of the extreme barrier function:

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

Forcing function

A forcing function $\rho(\cdot)$ is continuous, positive, and satisfies

$$\lim_{t \longrightarrow 0^+} \frac{\rho(t)}{t} = 0 \qquad \text{and} \qquad \rho(t_1) \leq \rho(t_2) \quad \text{if} \quad t_1 < t_2.$$

Forcing function

A forcing function $\rho(\cdot)$ is continuous, positive, and satisfies

$$\lim_{t \longrightarrow 0^+} \frac{\rho(t)}{t} = 0 \qquad \text{and} \qquad \rho(t_1) \leq \rho(t_2) \quad \text{if} \quad t_1 < t_2.$$

A simple example of a forcing function is $\rho(t) = t^2$.

Forcing function

A forcing function $\rho(\cdot)$ is continuous, positive, and satisfies

$$\lim_{t \longrightarrow 0^+} \frac{\rho(t)}{t} = 0 \qquad \text{and} \qquad \rho(t_1) \leq \rho(t_2) \quad \text{if} \quad t_1 < t_2.$$

A simple example of a forcing function is $\rho(t) = t^2$.

We will use $\bar{\rho}(\cdot)$ for either $\rho(\cdot)$ above or $\rho(\cdot) = 0$.

A class of directional direct-search methods

Includes Coordinate Search, GPS or GSS, and MADS ...

A class of directional direct-search methods

Includes Coordinate Search, GPS or GSS, and MADS ...

Initialization: Choose $x_0 \in \Omega$ with $f(x_0) < +\infty$, and $\alpha_0 > 0$.

Let $\mathcal{D}$ be a (possibly infinite) set of positive spanning sets.

Includes Coordinate Search, GPS or GSS, and MADS ...

Initialization: Choose $x_0 \in \Omega$ with $f(x_0) < +\infty$, and $\alpha_0 > 0$.

Let $\mathcal{D}$ be a (possibly infinite) set of positive spanning sets.

For $k = 0, 1, 2, \ldots$

(1) Search step (optional): Try to compute a point x with

$$f_\Omega(x) < f(x_k) - \bar{\rho}(\alpha_k)$$

by evaluating the function f at a finite number of points.

A class of directional direct-search methods

Includes Coordinate Search, GPS or GSS, and MADS ...

Initialization: Choose $x_0 \in \Omega$ with $f(x_0) < +\infty$, and $\alpha_0 > 0$.

Let $\mathcal{D}$ be a (possibly infinite) set of positive spanning sets.

For $k = 0, 1, 2, \ldots$

(1) Search step (optional): Try to compute a point x with

$$f_\Omega(x) < f(x_k) - \bar{\rho}(\alpha_k)$$

by evaluating the function f at a finite number of points.

If such a point is found then set $x_{k+1} = x$, declare the iteration and the search step successful, and skip the poll step.

(2) Poll step: Choose a positive spanning set D_k from the set $\mathcal{D}$.

Order $P_k = \{x_k + \alpha_k d : \, d \in D_k\}$ and start evaluating f_Ω following the chosen order.

(2) Poll step: Choose a positive spanning set D_k from the set $\mathcal{D}$.

Order $P_k = \{x_k + \alpha_k d : d \in D_k\}$ and start evaluating f_Ω following the chosen order.

If a point $x_k + \alpha_k d_k$ is found such that

$$f_\Omega(x_k + \alpha_k d_k) < f(x_k) - \bar{\rho}(\alpha_k)$$

then stop polling, set $x_{k+1} = x_k + \alpha_k d_k$, and declare the iteration and the poll step successful.

(2) Poll step: Choose a positive spanning set D_k from the set $\mathcal{D}$.

Order $P_k = \{x_k + \alpha_k d : \ d \in D_k\}$ and start evaluating f_Ω following the chosen order.

If a point $x_k + \alpha_k d_k$ is found such that

$$f_\Omega(x_k + \alpha_k d_k) < f(x_k) - \bar{\rho}(\alpha_k)$$

then stop polling, set $x_{k+1} = x_k + \alpha_k d_k$, and declare the iteration and the poll step successful.

Otherwise declare the iteration (and the poll step) unsuccessful and set $x_{k+1} = x_k$.

(3) Step size update: If the iteration was successful then maintain or increase the step size parameter: $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$.

(3) Step size update: If the iteration was successful then maintain or increase the step size parameter: $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$.

Otherwise decrease the step size parameter: $\alpha_{k+1} \in [\beta_1\alpha_k, \beta_2\alpha_k]$.

(3) Step size update: If the iteration was successful then maintain or increase the step size parameter: $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$.

Otherwise decrease the step size parameter: $\alpha_{k+1} \in [\beta_1\alpha_k, \beta_2\alpha_k]$.

The parameters are chosen at initialization: $0 < \beta_1 \leq \beta_2 < 1$, and $\gamma \geq 1$

Poll step

The points $x_k + \alpha_k d \in P_k$ are called the poll points and the vectors $d \in D_k$ the poll vectors or poll directions.

Poll step

The points $x_k + \alpha_k d \in P_k$ are called the poll points and the vectors $d \in D_k$ the poll vectors or poll directions.

The purpose of the poll step is to ensure a decrease of the objective function for a sufficiently small step size parameter α_k.

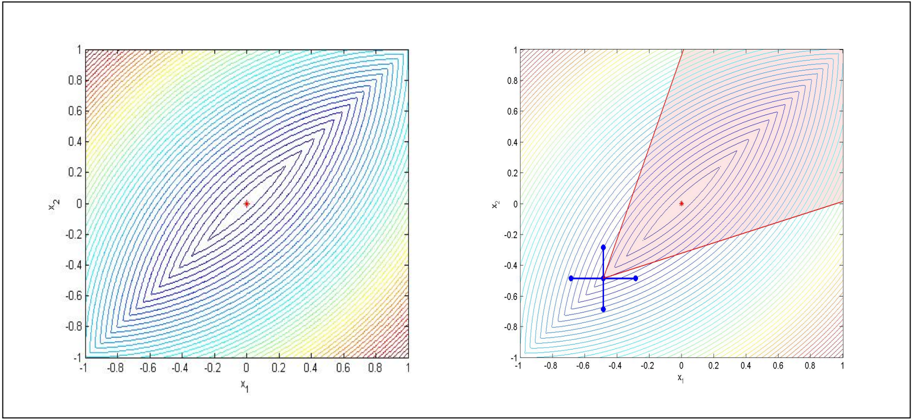
Poll step

The points $x_k + \alpha_k d \in P_k$ are called the poll points and the vectors $d \in D_k$ the poll vectors or poll directions.

The purpose of the poll step is to ensure a decrease of the objective function for a sufficiently small step size parameter α_k.

Polling can be opportunistic (like before) or complete (all the poll points are evaluated and the best is taken if better than the current iterate).

Finite directions in the nonsmooth case



The cone of descent directions at the poll center is shaded.

Need for infinitely many directions

There are two known ways to use an infinite number of directions and still guarantee global convergence (i.e., convergence from an arbitrary starting point):

Need for infinitely many directions

There are two known ways to use an infinite number of directions and still guarantee global convergence (i.e., convergence from an arbitrary starting point):

(1) Using integer lattices and simple decrease:

$\longrightarrow$ directions must be extracted from a finite set with integer/rational properties

Need for infinitely many directions

There are two known ways to use an infinite number of directions and still guarantee global convergence (i.e., convergence from an arbitrary starting point):

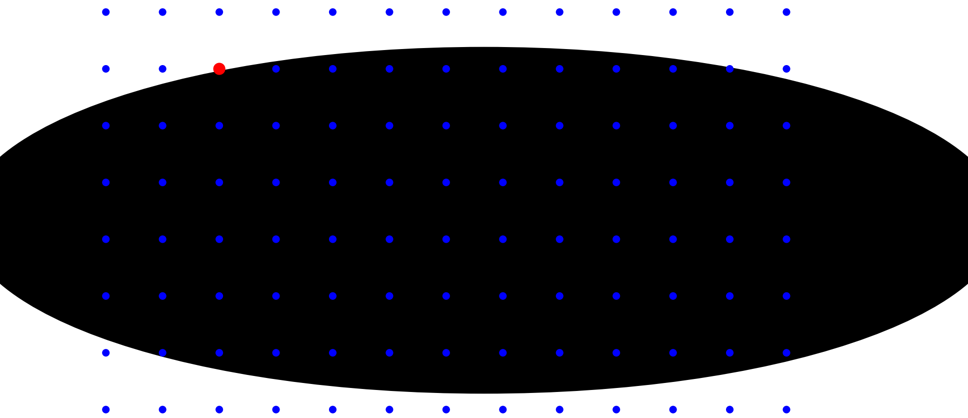
(1) Using integer lattices and simple decrease:

$\longrightarrow$ directions must be extracted from a finite set with integer/rational properties

$\longrightarrow$ search step must be restricted to the lattice (and the step size update must also follow integer/rational requirements).

(MADS by Audet and Dennis, 2006)

Integer lattice requirements (when using simple decrease)



$$L(x_0) = \{x \in \mathbb{R}^n : \ f(x) \leq f(x_0)\}$$

The intersection of $L(x_0)$ with the underlying lattice must be finite.

Need for infinitely many directions (suff. decrease)

(2) Using sufficient decrease:

$\longrightarrow$ directions can be randomly generated.

(like in GSS — see the 2003 SIAM Review paper of Kolda, Lewis, and Torczon)

Conditions on the directions

In both cases, one must also have:

Condition

The distance between x_k and the point $x_k + \alpha_k d_k$ tends to zero if and only if α_k does:

$$\lim_{k \in K} \alpha_k \|d_k\| \; = \; 0 \iff \lim_{k \in K} \alpha_k \; = \; 0,$$

for any infinite subsequence K.

Conditions on the directions

In both cases, one must also have:

Condition

The distance between x_k and the point $x_k + \alpha_k d_k$ tends to zero if and only if α_k does:

$$\lim_{k \in K} \alpha_k \|d_k\| \; = \; 0 \Longleftrightarrow \lim_{k \in K} \alpha_k \; = \; 0,$$

for any infinite subsequence K.

and

Condition

The set of refining directions at a limit point (associated with any refining subsequence) is dense is the unit sphere.

Behavior of the step size parameter

Assumption

The level set $L(x_0) = \{x \in \Omega : \ f(x) \leq f(x_0)\}$ *is bounded. The function* f *is bounded below in* $L(x_0)$.

Behavior of the step size parameter

Assumption

The level set $L(x_0) = \{x \in \Omega : \ f(x) \leq f(x_0)\}$ *is bounded. The function* f *is bounded below in* $L(x_0)$.

Lemma

If one uses (1) MADS or (2) suff. decrease, then $\exists$ *a subsequence* K *of unsuccessful iterations:*

$$\lim_{k \in K} x_k = x_* \quad \text{and} \quad \lim_{k \in K} \alpha_k = 0.$$

Behavior of the step size parameter

Assumption

The level set $L(x_0) = \{x \in \Omega : \; f(x) \leq f(x_0)\}$ *is bounded. The function* f *is bounded below in* $L(x_0)$*.*

Lemma

If one uses (1) MADS or (2) suff. decrease, then $\exists$ *a subsequence* K *of unsuccessful iterations:*

$$\lim_{k \in K} x_k = x_* \quad \textit{and} \quad \lim_{k \in K} \alpha_k = 0.$$

The proof goes back to Torczon 1997 (and was adapted to MADS later by Audet and Dennis, 2003 and 2006).

The case of suff. decrease is trivial (see IDFO book or the SIAM Rev. paper).

Refining subsequence and refining direction

This shows the existence of:

Definition

A subsequence $\{x_k\}_{k \in K}$ of iterates corresponding to unsuccessful poll steps is said to be a refining subsequence if $\alpha_k \longrightarrow 0$ in K.

Refining subsequence and refining direction

This shows the existence of:

Definition

A subsequence $\{x_k\}_{k \in K}$ of iterates corresponding to unsuccessful poll steps is said to be a refining subsequence if $\alpha_k \longrightarrow 0$ in K.

Then we have:

Definition

Let x_* be the limit point of a convergent refining subsequence.

Refining subsequence and refining direction

This shows the existence of:

Definition

A subsequence $\{x_k\}_{k \in K}$ *of iterates corresponding to unsuccessful poll steps is said to be a refining subsequence if* $\alpha_k \longrightarrow 0$ *in* K.

Then we have:

Definition

Let x_* *be the limit point of a convergent refining subsequence.*

If $\lim_{k \in L} d_k / \|d_k\|$ *exists, where* $L \subseteq K$ *and* $d_k \in D_k$, *then this limit is said to be a refining direction for* x_*

Refining subsequence and refining direction

This shows the existence of:

Definition

A subsequence $\{x_k\}_{k \in K}$ of iterates corresponding to unsuccessful poll steps is said to be a refining subsequence if $\alpha_k \longrightarrow 0$ in K.

Then we have:

Definition

Let x_* be the limit point of a convergent refining subsequence.

If $\lim_{k \in L} d_k / \|d_k\|$ exists, where $L \subseteq K$ and $d_k \in D_k$, then this limit is said to be a refining direction for x_*

($x_k + \alpha_k d_k$ must be in Ω for sufficiently large $k \in L$).

Focus first on the unconstrained case

For the analysis, let us consider first the case $\Omega = \mathbb{R}^n$.

Clarke generalized directional derivative

Definition

For f Lipschitz continuous near x_*, the Clarke generalized directional derivative is:

$$f_C^\circ(x_*; v) = \limsup_{x' \to x_* \; t \downarrow 0} \frac{f(x' + tv) - f(x')}{t}.$$

Clarke generalized directional derivative

Definition

For f Lipschitz continuous near x_*, the Clarke generalized directional derivative is:

$$f_C^\circ(x_*; v) = \limsup_{x' \to x_* \; t \downarrow 0} \frac{f(x' + tv) - f(x')}{t}.$$

If f is increasing from x_*, along v, then

$$f_C^\circ(x_*; v) \;\geq\; 0.$$

Clarke generalized directional derivative

Definition

For f Lipschitz continuous near x_*, the Clarke generalized directional derivative is:

$$f_C^\circ(x_*;v) = \limsup_{x' \to x_*} \limsup_{t \downarrow 0} \frac{f(x'+tv) - f(x')}{t}.$$

If f is increasing from x_*, along v, then

$$f_C^\circ(x_*;v) \ \geq \ 0.$$

Definition

x_* is a Clarke stationary point if

$$f_C^\circ(x_*;v) \ \geq \ 0, \quad \forall v \in \mathbb{R}^n.$$

Analysis of MADS or of the suff. decrease variant

Assumption

- $\{x_k\}_{k \in K}$ *refining subsequence converging to* x_*.
- f *Lipschitz continuous near* x_*.

Analysis of MADS or of the suff. decrease variant

Assumption

- $\{x_k\}_{k\in K}$ *refining subsequence converging to* x_*.
- f *Lipschitz continuous near* x_*.

Theorem

If v *is a refining direction for* x_*:

$$f_C^\circ(x_*;v) \ \geq \ 0.$$

For the case of MADS, the proof is due to Audet and Dennis, 2006.

Non-Lipschitzian case

What happens in the case where f is not Lipschitz continuous near x_*?

- The previous result does not apply!

Non-Lipschitzian case

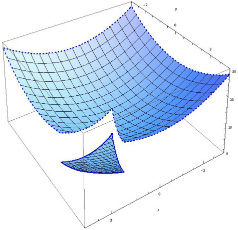
What happens in the case where f is not Lipschitz continuous near x_*?

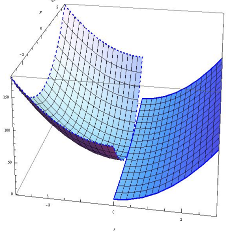
- The previous result does not apply!
- However, f could still have directional derivatives...

Non-Lipschitzian case

What happens in the case where f is not Lipschitz continuous near x_*?

- The previous result does not apply!
- However, f could still have directional derivatives...
- and they could be nonnegative!





Rockafellar generalized directional derivative

Definition

For f lower semicontinuous at x_*, the Rockafellar generalized directional derivative is:

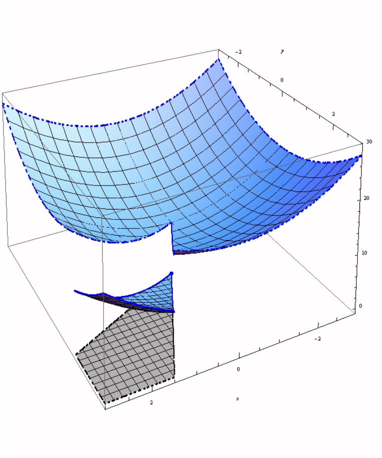
$$f_R^\circ(x_*; v) = \limsup_{x' \to_f x_*, t \downarrow 0} \frac{f(x' + tv) - f(x')}{t}.$$

The notation $x' \to_f x_*$ represents $x' \to x_*$ and $f(x') \to f(x_*)$.

The next question is...

When do these derivatives exist?

When do these derivatives exist?



Directionally Lipschitzian function w.r.t. a direction

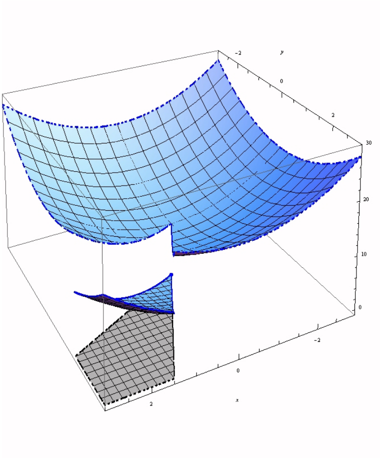
They exist when the following property holds:

Definition

f *directionally Lipschitzian at* x_* *with respect to* v *when*

$$\limsup_{x' \to_f x_*, t \downarrow 0} \sup_{v' \to v} \frac{f(x' + tv') - f(x')}{t} < +\infty.$$

Directionally Lipschitzian function w.r.t. a direction



This function is directionally Lipschitzian w.r.t. the directions in the interior of the shaded region.

Back to our direct-search results...

Assumption

- $\{x_k\}_{k \in K}$ *refining subsequence converging to* $x_* \in \Omega$.
- f *lower semicontinuous at* x_*.
- $\lim_{k \in K} f(x_k) = f(x_*)$.

Back to our direct-search results...

Assumption

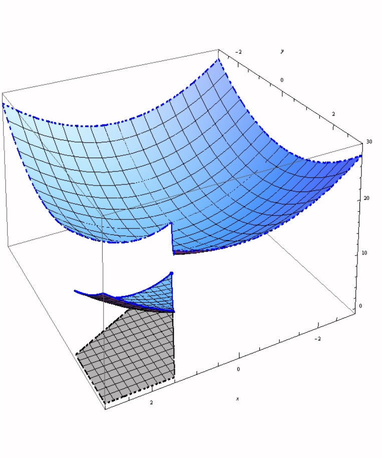
- $\{x_k\}_{k \in K}$ *refining subsequence converging to* $x_* \in \Omega$.
- f *lower semicontinuous at* x_*.
- $\lim_{k \in K} f(x_k) = f(x_*)$.

Theorem

If v *is a refining direction for* x_* *w.r.t. which* f *directionally Lipschitzian at* x_**:*

$$f_R^\circ(x_*; v) \ \geq \ 0.$$

Directionally Lipschitzian function w.r.t. a direction



This theorem allow us to state results in the interior of the shaded region.

One also has

Definition

The Rockafellar upper subderivative is defined by

$$f^{\uparrow}(x_*;v) \;=\; \limsup_{x' \to_f x_*,\, t \downarrow 0} \; \inf_{v' \to v} \frac{f(x'+tv') - f(x')}{t}.$$

Rockafellar upper subderivative

Due to...

Theorem (Rockafellar)

$$f^{\uparrow}(x_*; v) \;=\; \liminf_{v' \to v} f_R^{\circ}(x_*; v').$$

... we would also get, in this example, $f^{\uparrow}(x_*; v) \geq 0$ along the border of the shaded region.

The constrained case

As in the continuous case (Audet and Dennis, 2006):

(1) Redefine the derivatives accordingly: $f_R^\circ(x_*; v)$ and $f^\uparrow(x_*; v)$.

The constrained case

As in the continuous case (Audet and Dennis, 2006):

(1) Redefine the derivatives accordingly: $f_R^\circ(x_*; v)$ and $f^\uparrow(x_*; v)$.

(2) Rederive the basic result for refining directions v in the hypertangent cone (for which f is dir. Lips. w.r.t.):

$$f_R^\circ(x_*; v) \;\geq\; 0.$$

The constrained case

As in the continuous case (Audet and Dennis, 2006):

(1) Redefine the derivatives accordingly: $f_R^\circ(x_*; v)$ and $f^\uparrow(x_*; v)$.

(2) Rederive the basic result for refining directions v in the hypertangent cone (for which f is dir. Lips. w.r.t.):

$$f_R^\circ(x_*; v) \;\geq\; 0.$$

(3) For directions in the tangent cone (not in the hypertangent cone) use continuity of $f_R^\circ(x_*; v)$ and l.s.c. of $f^\uparrow(x_*; v)$.

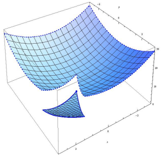
Finite number of step discontinuities

Let us look, now, at functions locally defined by a finite number of steps or branches:

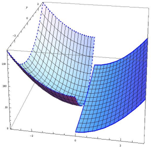
Assumption

There exists a neighborhood $B = \bigcup_{i=1}^{n_B} B_i$ *of* x_* *such that:*

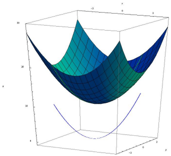
- $int(B_i) \neq \emptyset$
- $cl(B_i)$ *has the exterior cone property*
- f *is Lipschitz continuous in* $int(B_i)$ *and can be cont. extended to* ∂B_i



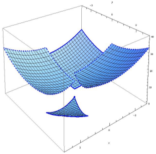
f_1



f_2



f_4 fails!



f_5

Finite number of step discontinuities

Theorem

If x_* belongs to the interior of a partition set, then $f_C^\circ(x_*; v) \geq 0$ for all refining directions $v \in T_\Omega(x_*)$.

Finite number of step discontinuities

Theorem

If x_* belongs to the interior of a partition set, then $f_C^\circ(x_*; v) \geq 0$ for all refining directions $v \in T_\Omega(x_*)$.

Otherwise, there exists a partition set B' and $K' \subset K$ such that $\{x_k\}_{k \in K'} \subset \mathrm{cl}(B')$ and there is an infinite number of poll points belonging to both $\mathrm{int}(B')$ and $\mathbb{R}^n \setminus \mathrm{cl}(B')$.

Finite number of step discontinuities

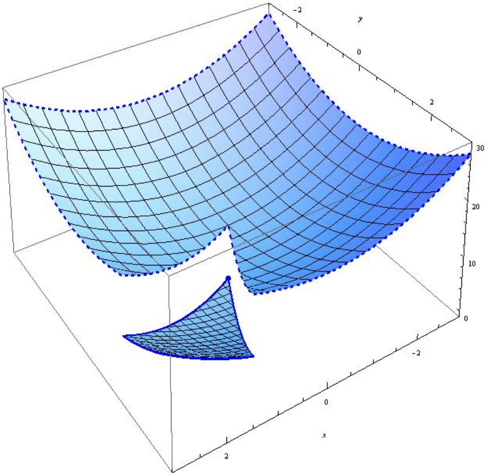
Theorem

If x_* belongs to the interior of a partition set, then $f_C^\circ(x_*; v) \geq 0$ for all refining directions $v \in T_\Omega(x_*)$.

Otherwise, there exists a partition set B' and $K' \subset K$ such that $\{x_k\}_{k \in K'} \subset \mathrm{cl}(B')$ and there is an infinite number of poll points belonging to both $\mathrm{int}(B')$ and $\mathbb{R}^n \setminus \mathrm{cl}(B')$.

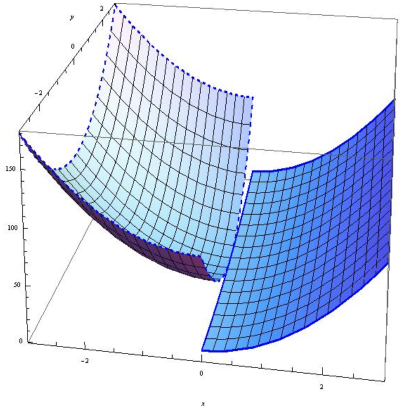
Corollary

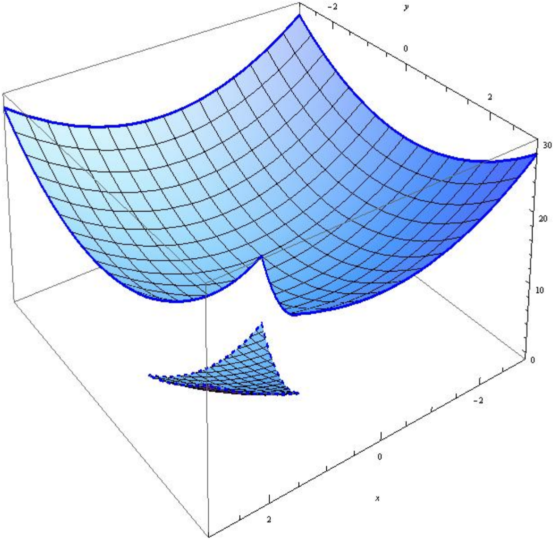
Additionally, if $n_B = 2$, then f attains its lowest values around x_* in B' and $\lim_{k \in K_*} f(x_k) = f(x_*)$ for $K_* \subset K'$.



-2
y
0
2
30
20
10
0
-2
0
x
2

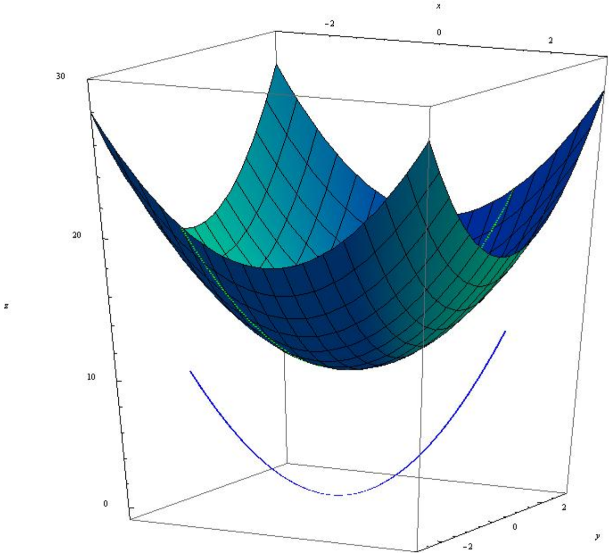
Function f_2



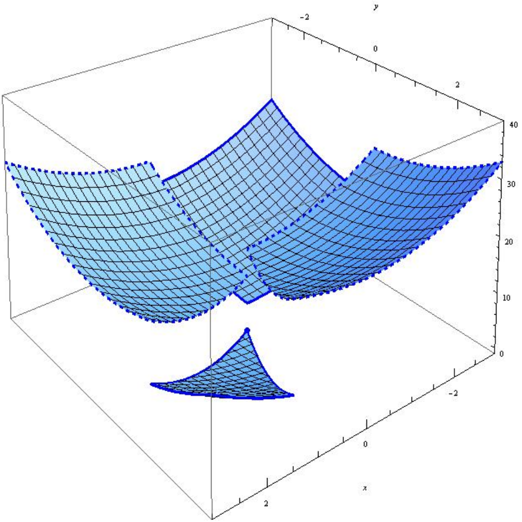


-2
y
0
2
30
20
10
0
-2
0
x
2

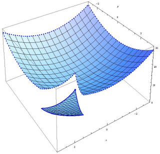
Function f_4



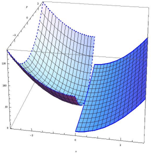
Function f_5



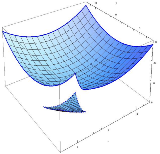
Numerical illustration



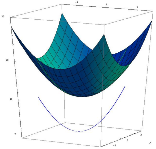
f_1



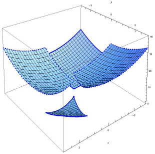
f_2



f_3

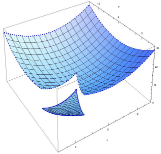


f_4

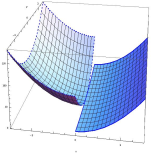


f_5

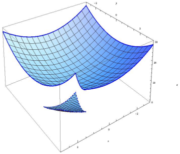
function	10 runs for the same initial point			
	MADS		Suff. Decrease	
	#failures	#fevals	#failures	#fevals
f_1	0	233	0	175.6
f_2	0	193.9	0	494.6
f_3	0	228.8	0	175.6
f_4	10	173.7	10	144.4
f_5	2	220.6	1	177.7



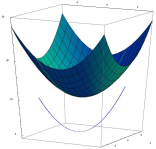
f_1



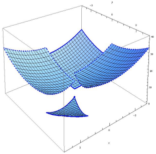
f_2



f_3



f_4



f_5

function	10 runs for the same initial point			
	MADS		Suff. Decrease	
	#failures	#fevals	#failures	#fevals
f_1	0	233	0	175.6
f_2	0	193.9	0	494.6
f_3	0	228.8	0	175.6
f_4	10	173.7	10	144.4
f_5	2	220.6	1	177.7

Suff. Decrease	
function	#failures
f_1	2
f_2	0
f_3	2
f_4	1000
f_5	61

10 runs for
100 initial points

References

- L. N. Vicente and A. L. Custódio, Analysis of direct searches for non-Lipschitzian functions, 2009.

(2) Poll step: Choose a positive spanning set D_k from the set $\mathcal{D}$.

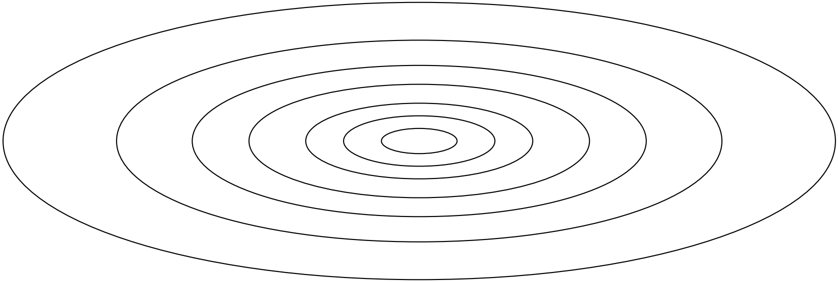
Order $P_k = \{x_k + \alpha_k d : \, d \in D_k\}$ and start evaluating f_Ω following the chosen order.

If a point $x_k + \alpha_k d_k$ is found such that

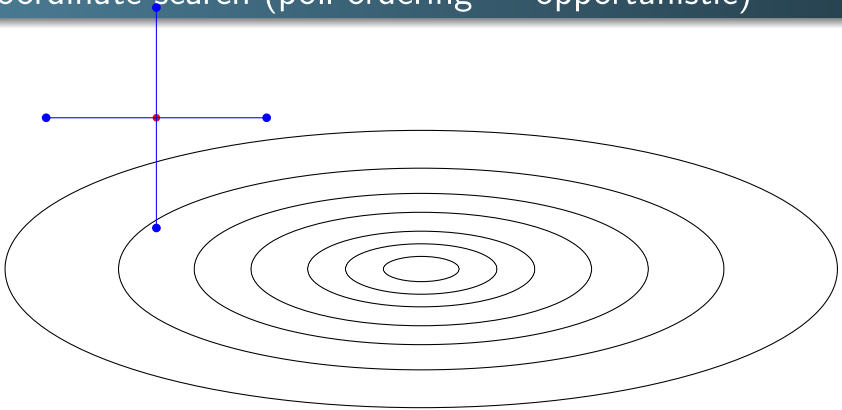
$$f_\Omega(x_k + \alpha_k d_k) < f(x_k) - \bar{\rho}(\alpha_k)$$

then stop polling, set $x_{k+1} = x_k + \alpha_k d_k$, and declare the iteration and the poll step successful.

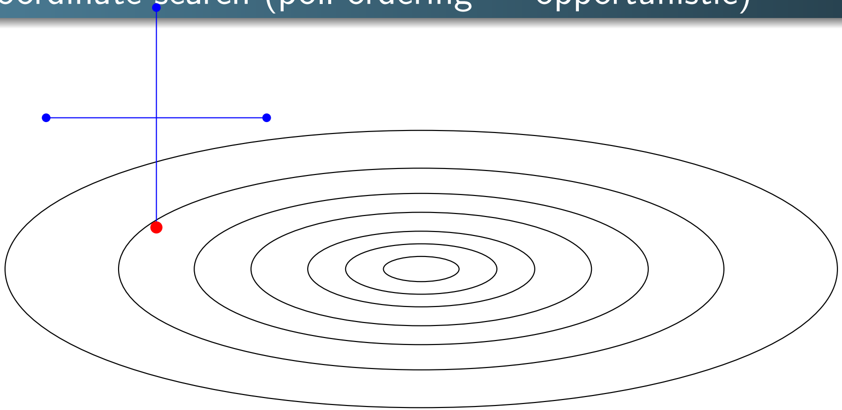
Coordinate search (poll ordering — opportunistic)



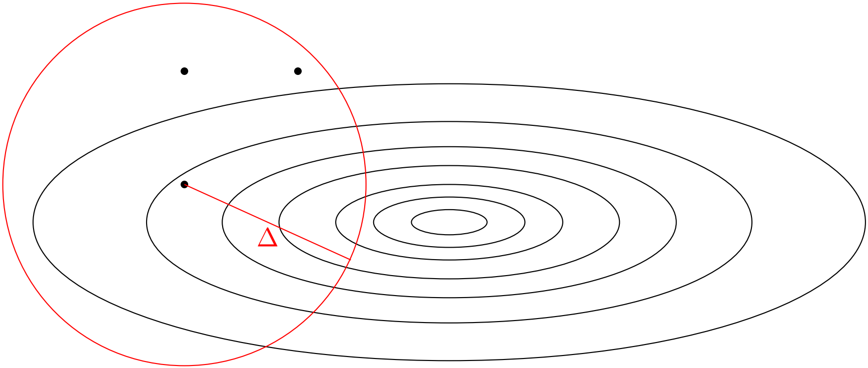
Coordinate search (poll ordering — opportunistic)



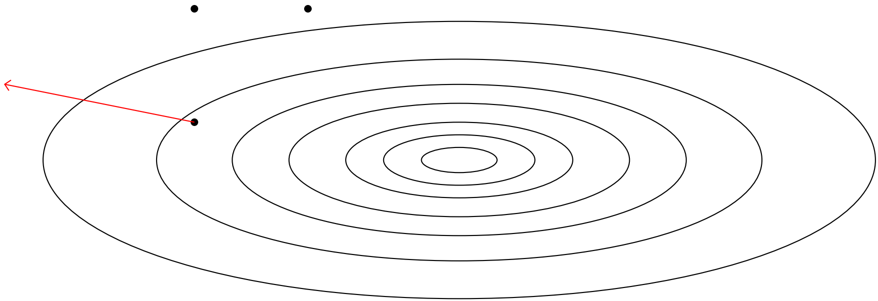
Coordinate search (poll ordering — opportunistic)



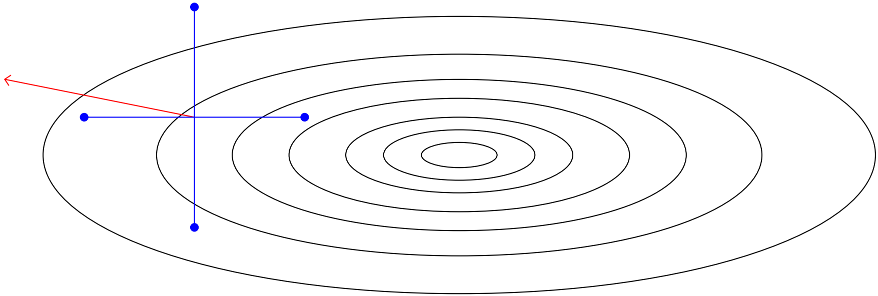
Coordinate search (poll ordering — opportunistic)



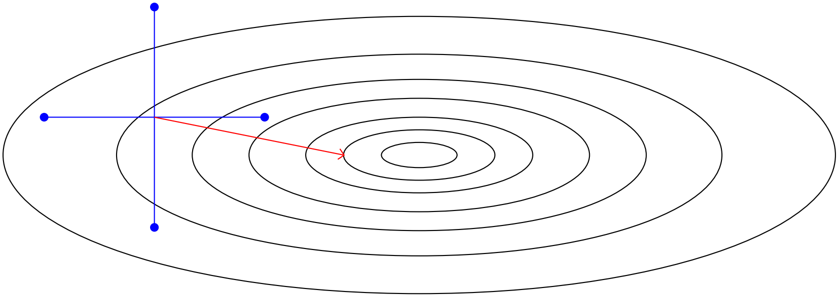
Coordinate search (poll ordering — opportunistic)



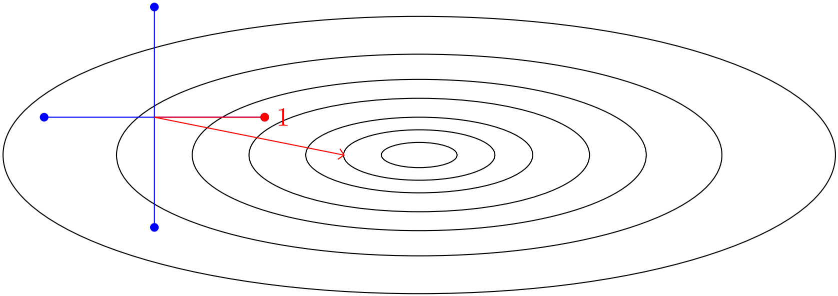
Coordinate search (poll ordering — opportunistic)



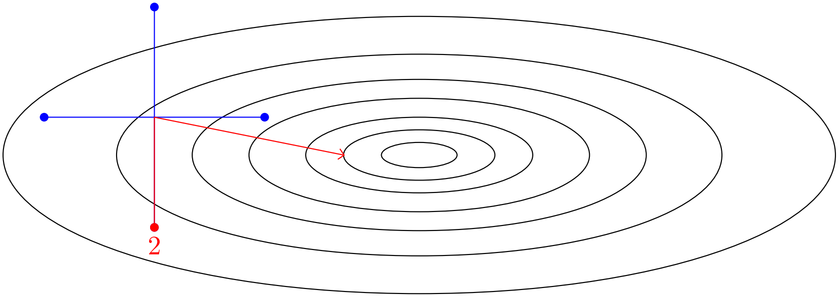
Coordinate search (poll ordering — opportunistic)



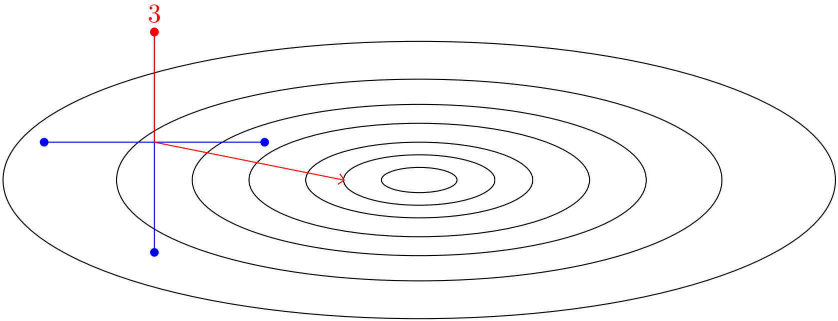
Coordinate search (poll ordering — opportunistic)



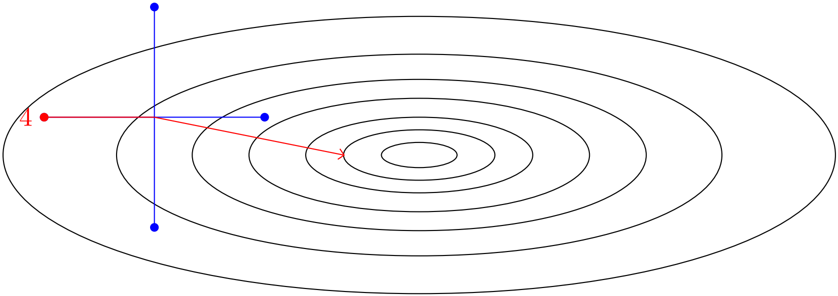
Coordinate search (poll ordering — opportunistic)



Coordinate search (poll ordering — opportunistic)



Coordinate search (poll ordering — opportunistic)



Poll ordering strategies

1. consecutive order of storage
2. random order

Poll ordering strategies

1. consecutive order of storage
2. random order
3. dynamic order (start by last successful direction)

Poll ordering strategies

1. consecutive order of storage
2. random order
3. dynamic order (start by last successful direction)
4. cycling order (start by the vector right after the last tested in the previous iteration)

Poll ordering strategies

1. consecutive order of storage
2. random order
3. dynamic order (start by last successful direction)
4. cycling order (start by the vector right after the last tested in the previous iteration)
5. (simplex descent) indicator order
   $\longrightarrow$ uses the negative simplex gradient

Poll ordering strategies

1. consecutive order of storage
2. random order
3. dynamic order (start by last successful direction)
4. cycling order (start by the vector right after the last tested in the previous iteration)
5. (simplex descent) indicator order
   $\longrightarrow$ uses the negative simplex gradient
6. (model descent) indicator order
7. model values order

Poll ordering strategies

1. consecutive order of storage
2. random order
3. dynamic order (start by last successful direction)
4. cycling order (start by the vector right after the last tested in the previous iteration)
5. (simplex descent) indicator order
   $\longrightarrow$ uses the negative simplex gradient
6. (model descent) indicator order
7. model values order
8. combinations of previous strategies:

Best strategy (for us):

negative simplex gradient + cycling order

Poll ordering by simplex gradients

$\longrightarrow$ Provided reductions of over 50% for smooth problems and over 25% for nonsmooth problems.

Poll ordering by simplex gradients

⟶ Provided reductions of over 50% for smooth problems and over 25% for nonsmooth problems.

⟶ It is possible to identify subsequences of well-poised sets in GPS and MADS.

⟶ Analysis (in particular in the nonsmooth case) indicates favorable directional properties of the negative simplex gradient.

Poll ordering by simplex gradients

⟶ Provided reductions of over 50% for smooth problems and over 25% for nonsmooth problems.

⟶ It is possible to identify subsequences of well-poised sets in GPS and MADS.

⟶ Analysis (in particular in the nonsmooth case) indicates favorable directional properties of the negative simplex gradient.

1. A. L. Custódio and L. N. Vicente, Using sampling and simplex derivatives in pattern search methods, SIAM Journal on Optimization, 18 (2007), 537-555
2. A. L. Custódio, J. E. Dennis Jr., and L. N. Vicente, Using simplex gradients of nonsmooth functions in direct search methods, IMA Journal of Numerical Analysis, 28 (2008), 770-784

Recall... the class of directional direct-search methods

(1) Search step (optional): Try to compute a point x with

$$f_\Omega(x) < f(x_k) - \bar{\rho}(\alpha_k)$$

by evaluating the function f at a finite number of points.

If such a point is found then set $x_{k+1} = x$, declare the iteration and the search step successful, and skip the poll step.

Search step

The search step consists of evaluating the objective function at a finite number of points.

Search step

The search step consists of evaluating the objective function at a finite number of points.

The search step is optional and is not necessary for the convergence properties of the method.

Search step

The search step consists of evaluating the objective function at a finite number of points.

The search step is optional and is not necessary for the convergence properties of the method.

The search step can take advantage of the existence of surrogate models for f to improve the efficiency of the direct-search method.

Search step requirements (when using simple decrease)

The search step must return a lattice point when $\bar{\rho}(\cdot) = 0$.

Minimum Frobenius Norm quadratic models

Given a sample set, MFN quadratic models are formed by:

$$\begin{aligned} \min \quad & \|\text{model Hessian}\|_F \\ \text{s.t} \quad & \text{interpolation conditions.} \end{aligned}$$

Minimum Frobenius Norm quadratic models

Given a sample set, MFN quadratic models are formed by:

$$\begin{array}{rl} \min & \|\text{model Hessian}\|_F \\ \text{s.t} & \text{interpolation conditions.} \end{array}$$

$\longrightarrow$ MFN models can be shown 'fully linear' (see IDFO book).

$\longrightarrow$ MFN models provide very good numerical results when used in trust-region interpolation-based methods (within the codes `DFO` and `NEWUOA`).

Our search step

We collect previously evaluated poll points into a sample set.

Our search step

We collect previously evaluated poll points into a sample set.

Then, our search step consists of:

- Computing a MFN quadratic model when interpolation is underdetermined.

Our search step

We collect previously evaluated poll points into a sample set.

Then, our search step consists of:

- Computing a MFN quadratic model when interpolation is underdetermined.
- Computing an ℓ_2–regression quadratic model when interpolation is overdetermined.

Our search step

We collect previously evaluated poll points into a sample set.

Then, our search step consists of:

- Computing a MFN quadratic model when interpolation is underdetermined.
- Computing an ℓ_2–regression quadratic model when interpolation is overdetermined.
- Minimizing the model in a trust region in $B(x_k; \Delta_k)$ with

$$\Delta_k \;=\; \mathcal{O}(\alpha_k),$$

where α_k is the step size parameter in direct search.

Our code sid-psm does essentially:

- Search step as described above.

Our code sid-psm does essentially:

- Search step as described above.
- Poll step (no constraints and simple bounds): $D_k = [\, e \;\; -e \;\; I \;\; -I \,]$.

Our code sid-psm does essentially:

- Search step as described above.
- Poll step (no constraints and simple bounds): $D_k = [\, e \;\; -e \;\; I \;\; -I \,]$.
- Poll step: negative simplex gradient + cycling order.

Our code sid-psm does essentially:

- Search step as described above.
- Poll step (no constraints and simple bounds): $D_k = [\, e \;\; -e \;\; I \;\; -I \,]$.
- Poll step: negative simplex gradient + cycling order.

Software freely available at: `http://www.mat.uc.pt/sid-psm`

Numerical tests

Comparisons are made against the codes:

- APPSPACK — generalized pattern search (poll by random order), by T. G. Kolda's group.
- NEWUOA — interpolation-based trust-region method (least updating MFN models), by M. J. D. Powell.
- NMSMAX — Nelder-Mead method, by N. J. Higham.

Comparisons are made against the codes:

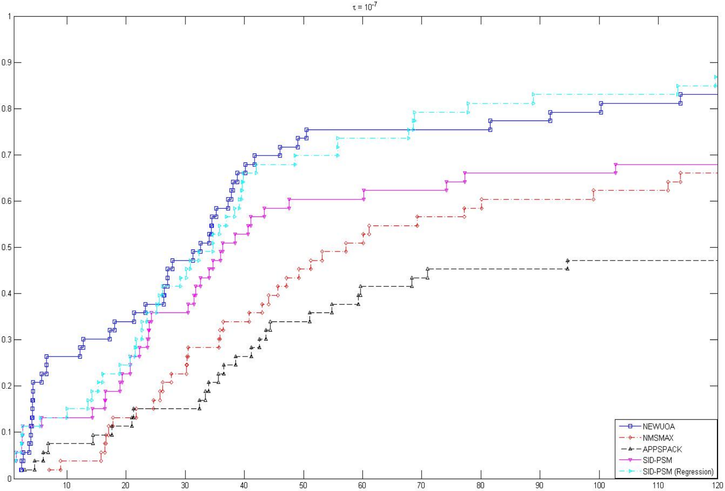
- APPSPACK — generalized pattern search (poll by random order), by T. G. Kolda's group.
- NEWUOA — interpolation-based trust-region method (least updating MFN models), by M. J. D. Powell.
- NMSMAX — Nelder-Mead method, by N. J. Higham.

Using an unconstrained test set (Moré and Wild) formed by:

- Smooth (53 nonlinear least squares problems obtained from CUTEr functions, with $n \in [2, 12]$).
- Non-stochastic noisy (adding oscillatory noise to the smooth ones).
- Non-differentiable (as in the smooth case but by taking ℓ_1 norms).
- Stochastic noisy (adding random noise to the smooth ones).

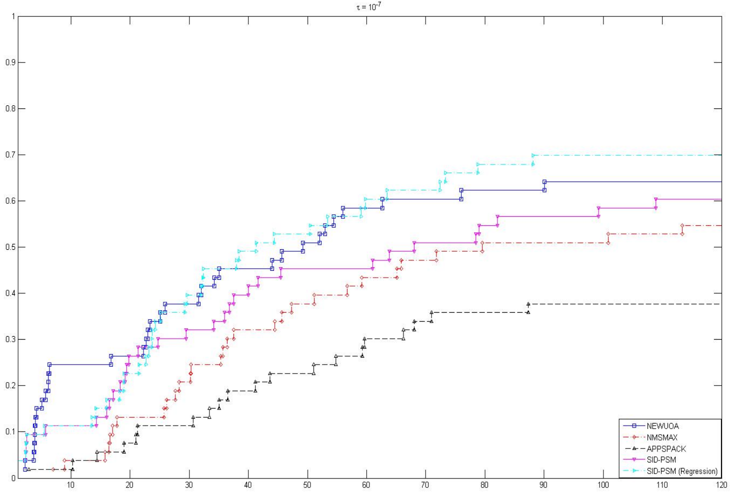
Data profiles — smooth

$\tau = 10^{-7}$

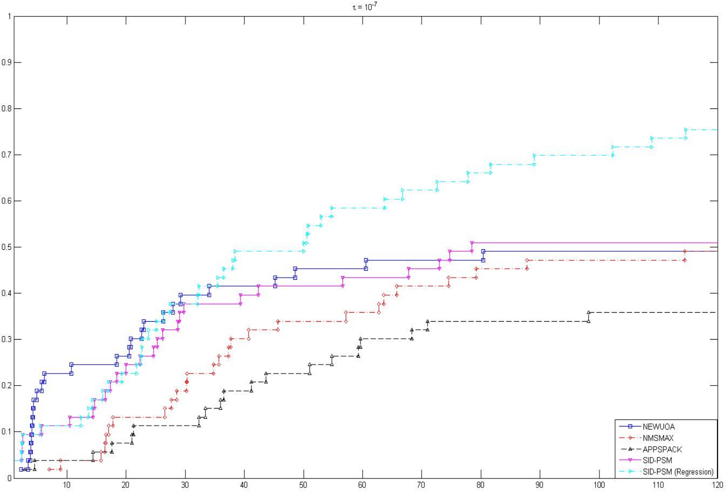


Data profiles — non-stochastic noisy

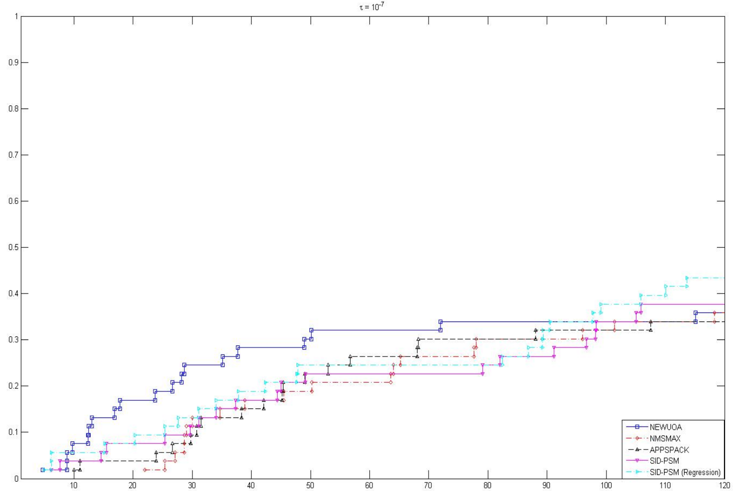
$\tau = 10^{-7}$



$\tau = 10^{-7}$


τ = 10^-7
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10
20
30
40
50
60
70
80
90
100
110
120
NEWUOA
NMSMAX
APPSPACK
SID-PSM
SID-PSM (Regression)

$\tau = 10^{-7}$


τ = 10^-7
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10
20
30
40
50
60
70
80
90
100
110
120
NEWUOA
NMSMAX
APPSPACK
SID-PSM
SID-PSM (Regression)

The study by Sahinihis and Rios

23 DFO codes have recently been tested and compared by:

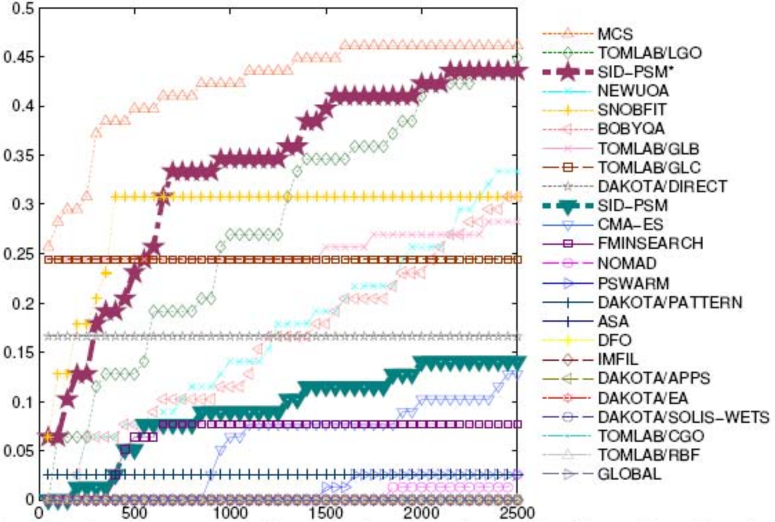
- L. M. Rios and N. Sahinidis, Derivative-free optimization: A review of algorithms and comparison of software implementations, 2010.

The study by Sahinihis and Rios

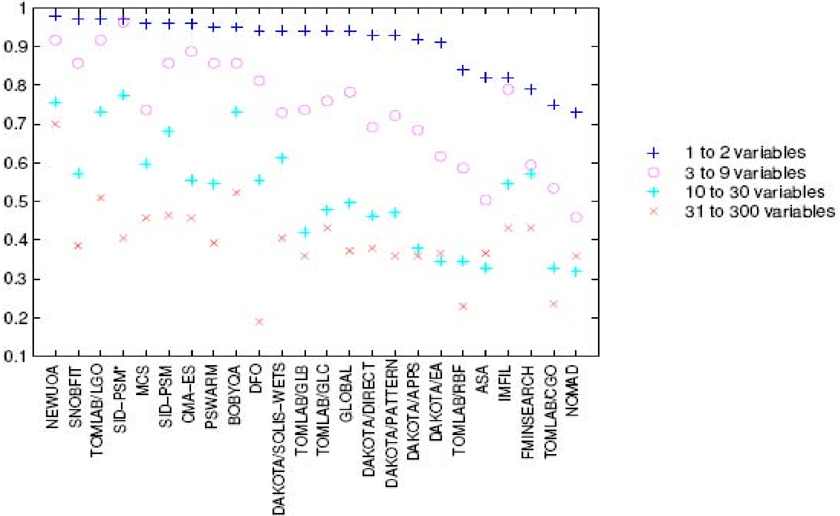
23 DFO codes have recently been tested and compared by:

- L. M. Rios and N. Sahinidis, Derivative-free optimization: A review of algorithms and comparison of software implementations, 2010.

$\longrightarrow$ Test set of 505 problems (convex & nonconvex).



Fraction of convex smooth problems solved as a function of allowable number of function evaluations.



Fraction of problems solved from a near-optimal solution.

References

- A. L. Custódio, H. Rocha, and L. N. Vicente, Incorporating minimum Frobenius norm models in direct search, published online in Computational Optimization and Applications.

Upcoming papers

Model based:

- Bilevel Derivative-Free Optimization and its Application to Robust Optimization

Upcoming papers

Model based:

- Bilevel Derivative-Free Optimization and its Application to Robust Optimization
- Sparse Reconstruction of First and Second Order Information and its Application to Derivative-Free Optimization

Upcoming papers

Model based:

- Bilevel Derivative-Free Optimization and its Application to Robust Optimization
- Sparse Reconstruction of First and Second Order Information and its Application to Derivative-Free Optimization

Direct search:

- Direct Multisearch for Multiobjective Optimization

Upcoming papers

Model based:

- Bilevel Derivative-Free Optimization and its Application to Robust Optimization
- Sparse Reconstruction of First and Second Order Information and its Application to Derivative-Free Optimization

Direct search:

- Direct Multisearch for Multiobjective Optimization
- Complexity of Direct Search can be $\mathcal{O}(\epsilon^{-2})$

The book!

- A. R. Conn, K. Scheinberg, and L. N. Vicente, Introduction to Derivative-Free Optimization, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2009.