MADS - Mesh Adaptive Direct Search for constrained optimization

Mark Abramson,
Charles Audet,
Gilles Couture,
John Dennis,

www.gerad.ca/Charles.Audet/

Thanks to: ExxonMobil, AFOSR, Boeing, LANL, FQRNT, NSERC, SANDIA, NSF.

Outline

- Statement of the optimization problem

Outline

- Statement of the optimization problem
- The GPS and MADS algorithm classes

Outline

- Statement of the optimization problem
- The GPS and MADS algorithm classes
- GPS theory and limiting examples

Outline

- Statement of the optimization problem
- The GPS and MADS algorithm classes
- GPS theory and limiting examples
- The MADS algorithm class

Outline

- Statement of the optimization problem
- The GPS and MADS algorithm classes
- GPS theory and limiting examples
- The MADS algorithm class
  - An implementable MADS instance
  - MADS theory
  - Numerical results

Outline

- Statement of the optimization problem
- The GPS and MADS algorithm classes
- GPS theory and limiting examples
- The MADS algorithm class
  - An implementable MADS instance
  - MADS theory
  - Numerical results
- Discussion

Outline

- Statement of the optimization problem
- The GPS and MADS algorithm classes
- GPS theory and limiting examples
- The MADS algorithm class
  - An implementable MADS instance
  - MADS theory
  - Numerical results
- Discussion

The target problem

$$(NLP) \qquad \text{minimize} \quad f(x)$$
$$\text{subject to} \quad x \in \Omega,$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ may be discontinuous and:

The target problem

$$(NLP) \qquad \text{minimize} \quad f(x)$$
$$\text{subject to} \quad x \in \Omega,$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{\infty\}$ may be discontinuous and:

- the functions are expensive black boxes,

The target problem

$$(NLP) \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega, \end{array}$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ may be discontinuous and:

- the functions are expensive black boxes,
- the functions provide few correct digits and may fail even for $x \in \Omega$

The target problem

$$(NLP) \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega, \end{array}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{\infty\}$ may be discontinuous and:

- the functions are expensive black boxes,
- the functions provide few correct digits and may fail even for $x \in \Omega$
- accurate approximation of derivatives is problematic

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X, \end{array}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{\infty\}$ may be discontinuous and:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X, \end{array}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{\infty\}$ may be discontinuous and:

- the functions are expensive black boxes, often produced by simulations or output of MDO codes

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X, \end{array}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{\infty\}$ may be discontinuous and:

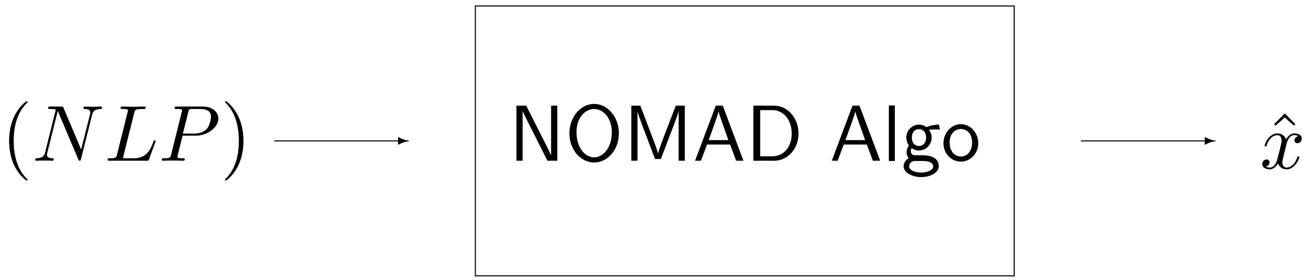
- the functions are expensive black boxes, often produced by simulations or output of MDO codes
- the functions provide few correct digits and may fail even for $x \in X$

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in X, \end{aligned}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{\infty\}$ may be discontinuous and:

- the functions are expensive black boxes, often produced by simulations or output of MDO codes
- the functions provide few correct digits and may fail even for $x \in X$
- accurate approximation of derivatives is problematic
- surrogate models $s \approx f$ and $P \approx X$ may be available

Goals – or validation of the method



Goals – or validation of the method

$(NLP) \longrightarrow$ NOMAD Algo $\longrightarrow \hat{x}$

if f is continuously differentiable then $\nabla f(\hat{x}) = 0$

Goals – or validation of the method

$(NLP) \longrightarrow$ NOMAD Algo $\longrightarrow \hat{x}$

if f is continuously differentiable then $\nabla f(\hat{x}) = 0$

if f is convex then $0 \in \underline{\partial} f(\hat{x})$

Goals – or validation of the method

$(NLP) \longrightarrow$ NOMAD Algo $\longrightarrow \hat{x}$

if f is continuously differentiable	then $\nabla f(\hat{x}) = 0$
if f is convex	then $0 \in \underline{\partial} f(\hat{x})$
if f is Lipschitz near $\hat{x}$	then $0 \in \partial f(\hat{x})$

Clarke Calculus – for f Lipschitz near x

- Clarke generalized derivative at x in the direction v:

$$f^\circ(x;v) \;=\; \limsup_{y\to x,\ t\downarrow 0} \frac{f(y+tv)-f(y)}{t}.$$

Clarke Calculus – for f Lipschitz near x

- Clarke generalized derivative at x in the direction v:

$$f^\circ(x;v) \;=\; \limsup_{y\to x,\ t\downarrow 0} \frac{f(y+tv)-f(y)}{t}.$$

- The generalized gradient of f at x is the set

$$\partial f(x) \;:=\; \{\zeta \in \mathbb{R}^n : f^\circ(x;v) \geq v^T\zeta \text{ for all } v \in \mathbb{R}^n\}$$

Clarke Calculus – for f Lipschitz near x

- Clarke generalized derivative at x in the direction v:

$$f^\circ(x; v) \;=\; \limsup_{y \to x,\; t \downarrow 0} \frac{f(y + tv) - f(y)}{t}.$$

- The generalized gradient of f at x is the set

$$\begin{aligned} \partial f(x) \;&:=\; \{\zeta \in \mathbb{R}^n : f^\circ(x; v) \geq v^T \zeta \text{ for all } v \in \mathbb{R}^n\} \\ &=\; \text{co}\{\lim \nabla f(y_i) \;:\; y_i \to x \text{ and } \nabla f(y_i) \text{ exists } \}. \end{aligned}$$

Clarke Calculus – for f Lipschitz near x

- Clarke generalized derivative at x in the direction v:

$$f^\circ(x;v) \;=\; \limsup_{y\to x,\ t\downarrow 0} \frac{f(y+tv)-f(y)}{t}.$$

- The generalized gradient of f at x is the set

$$\begin{aligned} \partial f(x) \;&:=\; \{\zeta\in\mathbb{R}^n : f^\circ(x;v)\ge v^T\zeta \text{ for all } v\in\mathbb{R}^n\} \\ &=\; \text{co}\{\lim \nabla f(y_i) \;:\; y_i\to x \text{ and } \nabla f(y_i) \text{ exists }\}. \end{aligned}$$

- $f^\circ(x;v)$ can be obtained from $\partial f(x)$:
$f^\circ(x;v)=\max\{v^T\zeta : \zeta\in\partial f(x)\}$.

Outline

- Statement of the optimization problem
- The GPS and MADS algorithm classes
- GPS theory and limiting examples
- The MADS algorithm class
  - An implementable MADS instance
  - MADS theory
  - Numerical results
- Discussion

The two iterated phases of GPS and MADS

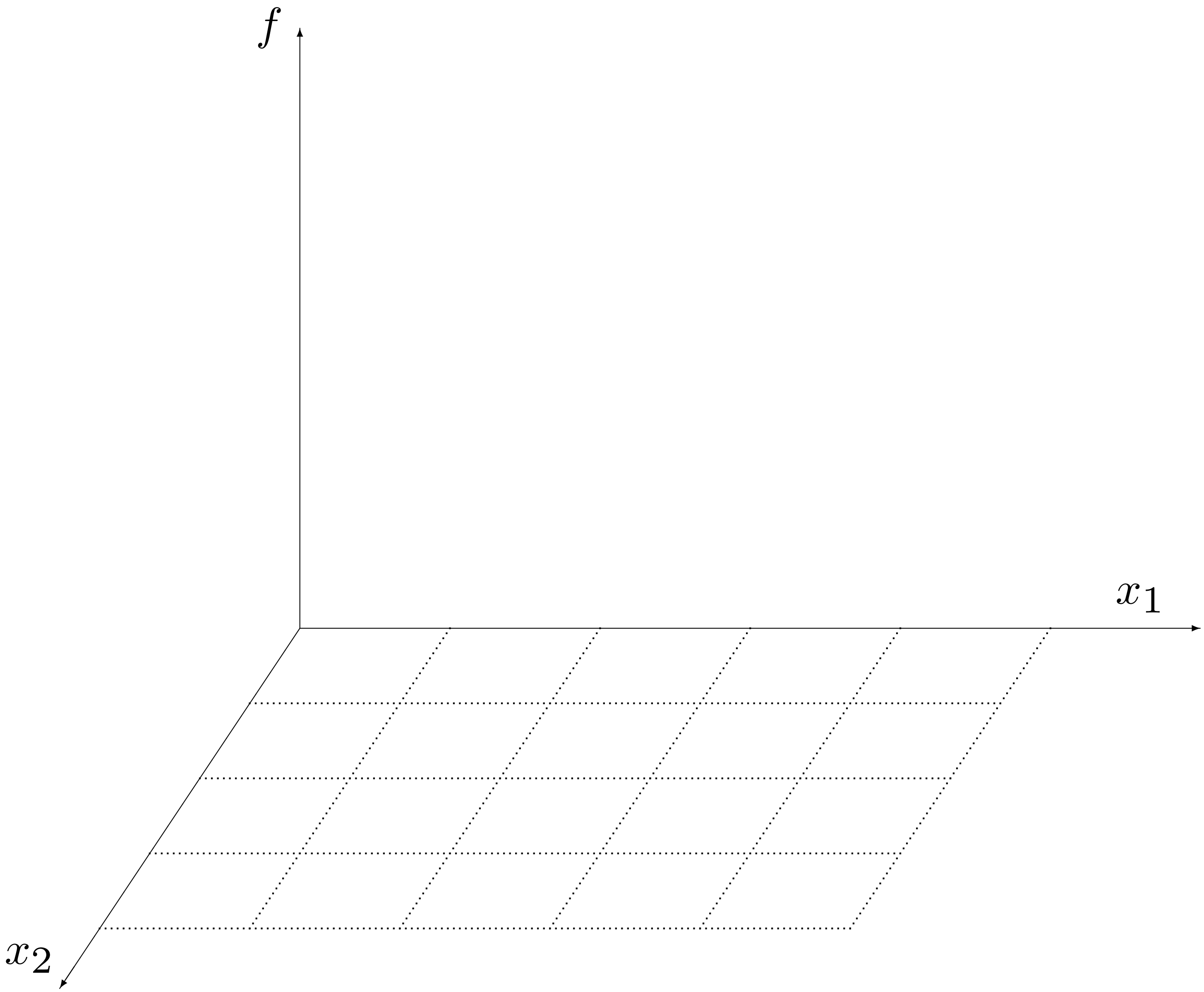
- The GLOBAL SEARCH in the variable space is flexible enough to allow user heuristics that incorporate knowledge of the driving simulation model and facilitate the use of surrogate functions.

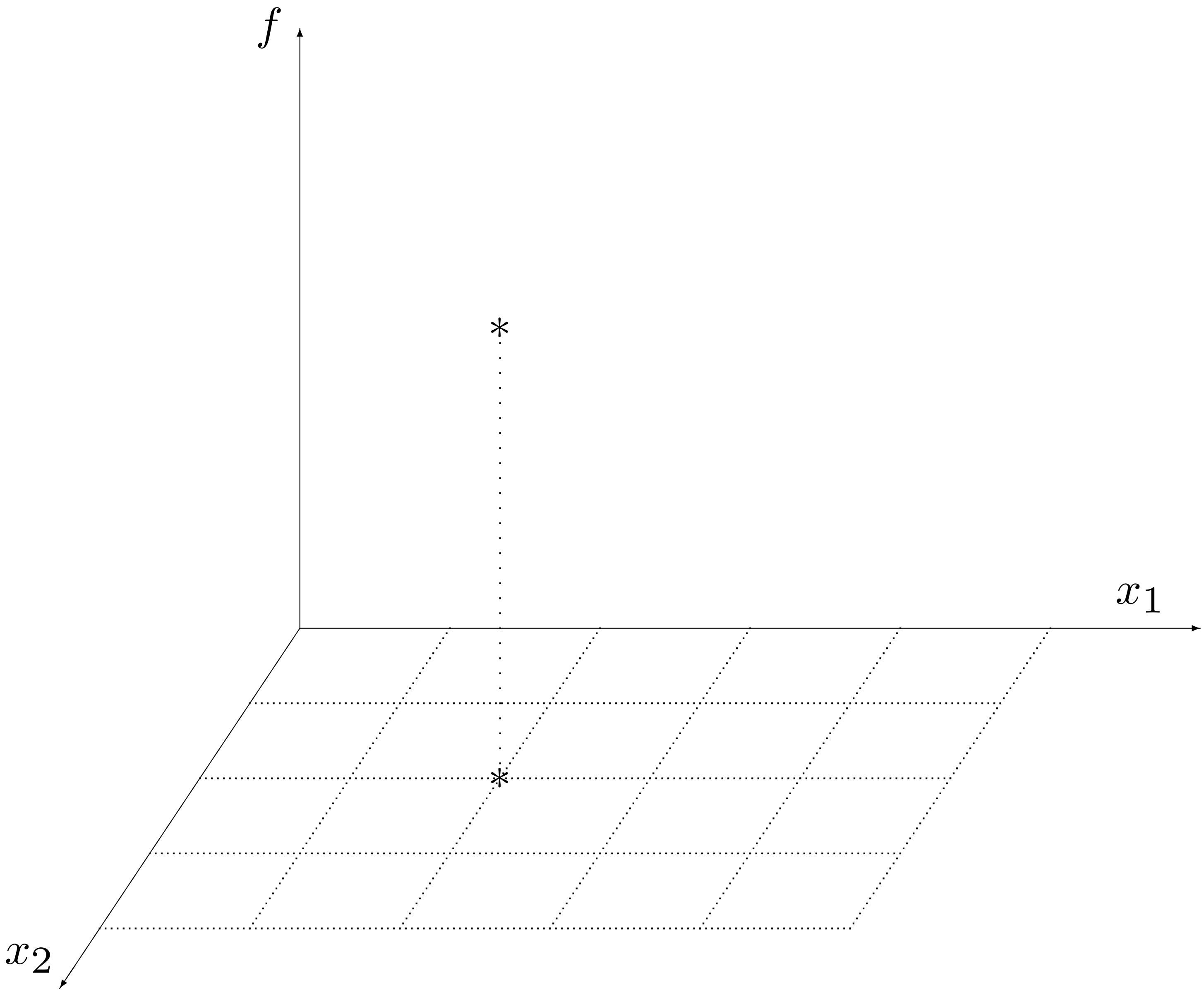
The two iterated phases of GPS and MADS

- The GLOBAL SEARCH in the variable space is flexible enough to allow user heuristics that incorporate knowledge of the driving simulation model and facilitate the use of surrogate functions.
- The LOCAL POLL around the incumbent solution is more rigidly defined, but it ensures convergence to a point satisfying necessary first order optimality conditions.

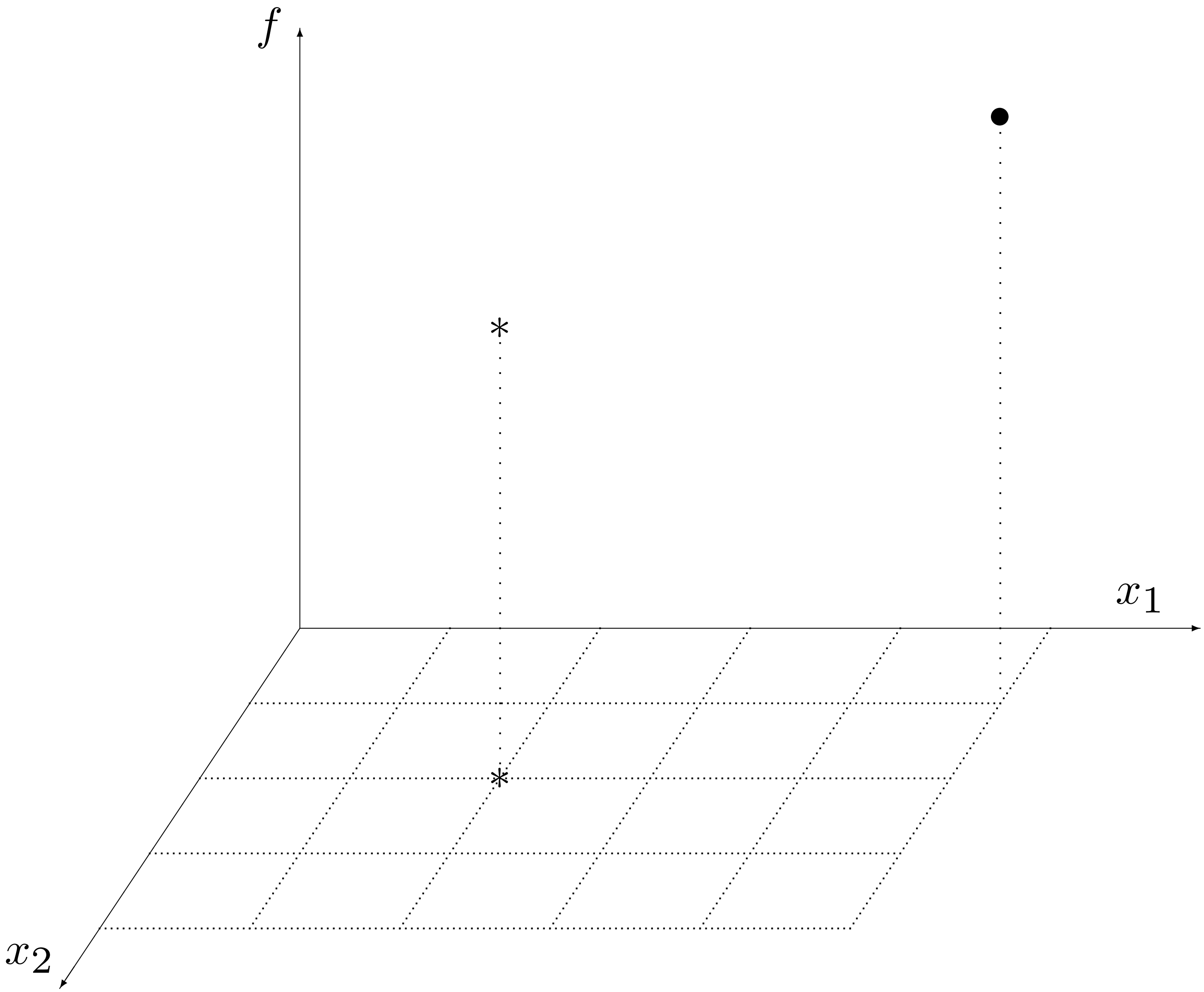
The two iterated phases of GPS and MADS

- The GLOBAL SEARCH in the variable space is flexible enough to allow user heuristics that incorporate knowledge of the driving simulation model and facilitate the use of surrogate functions.
- The LOCAL POLL around the incumbent solution is more rigidly defined, but it ensures convergence to a point satisfying necessary first order optimality conditions.
- This talk focusses on the basic algorithm, and the convergence analysis. In the next talks, Alison, Mark and Gilles will talk about surrogates in the SEARCH.

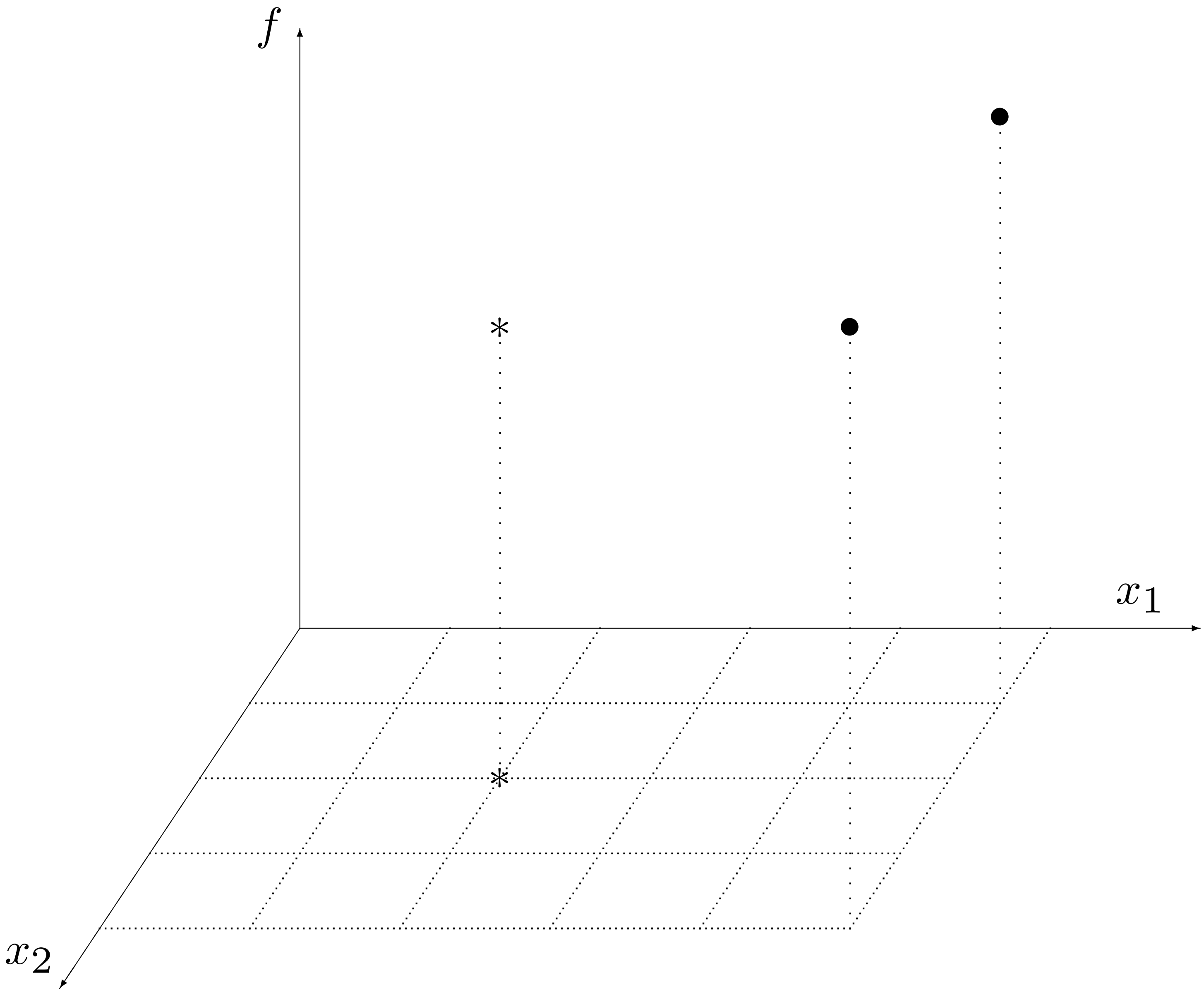

f
x1
x2



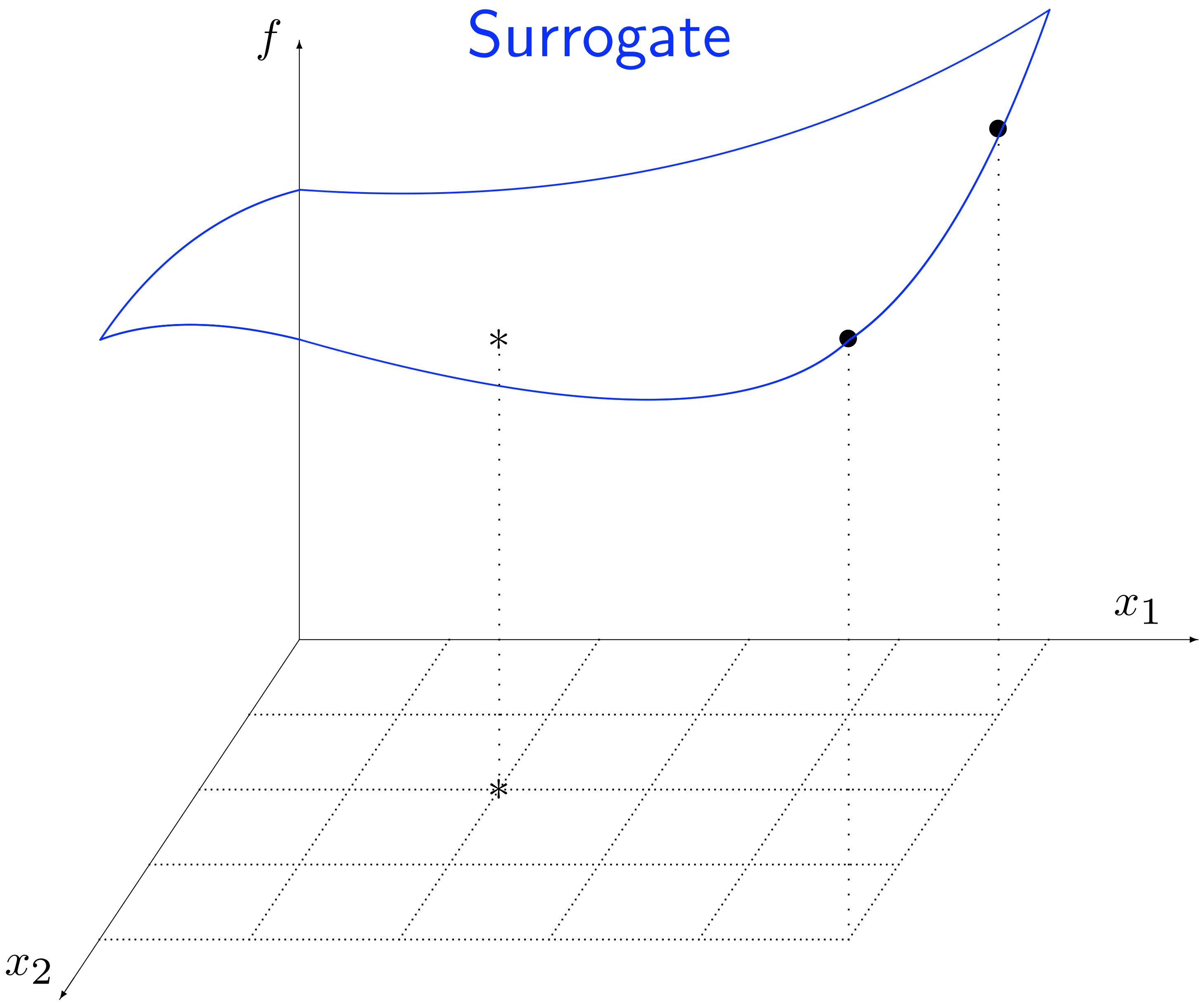
$*$ is the incumbent solution



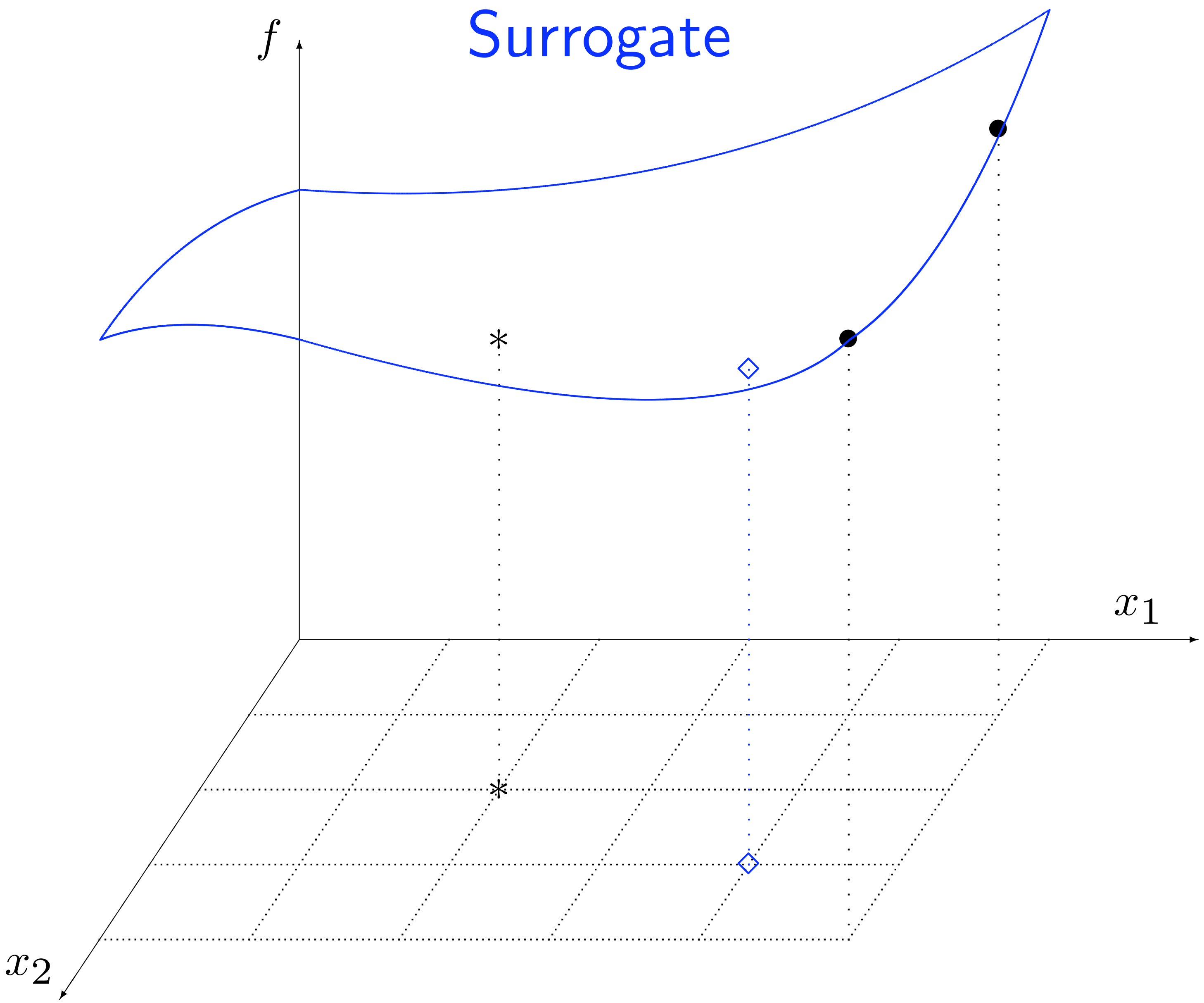
$*$ is the incumbent solution



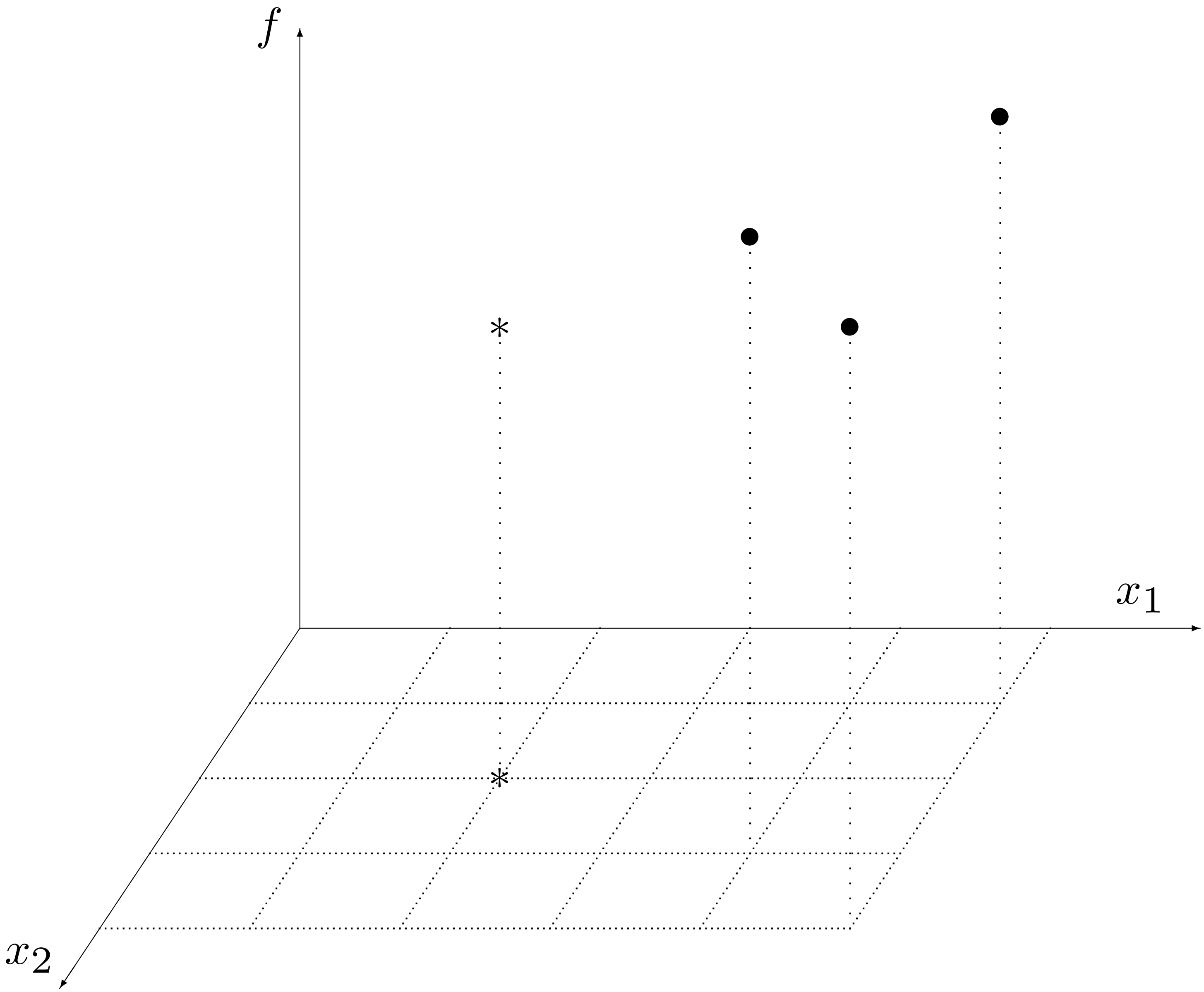
$*$ is the incumbent solution



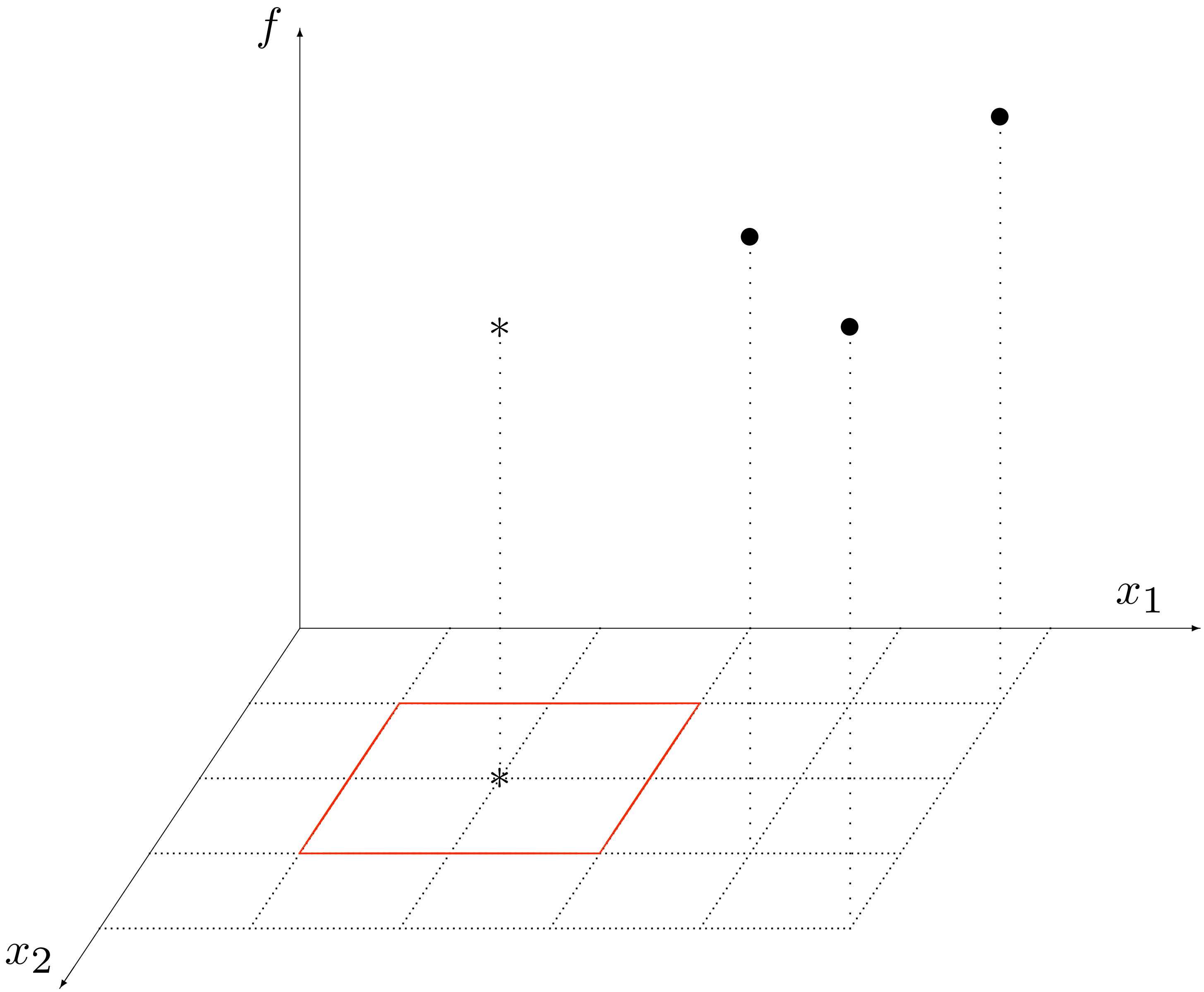
$*$ is the incumbent solution



$*$ is the incumbent solution

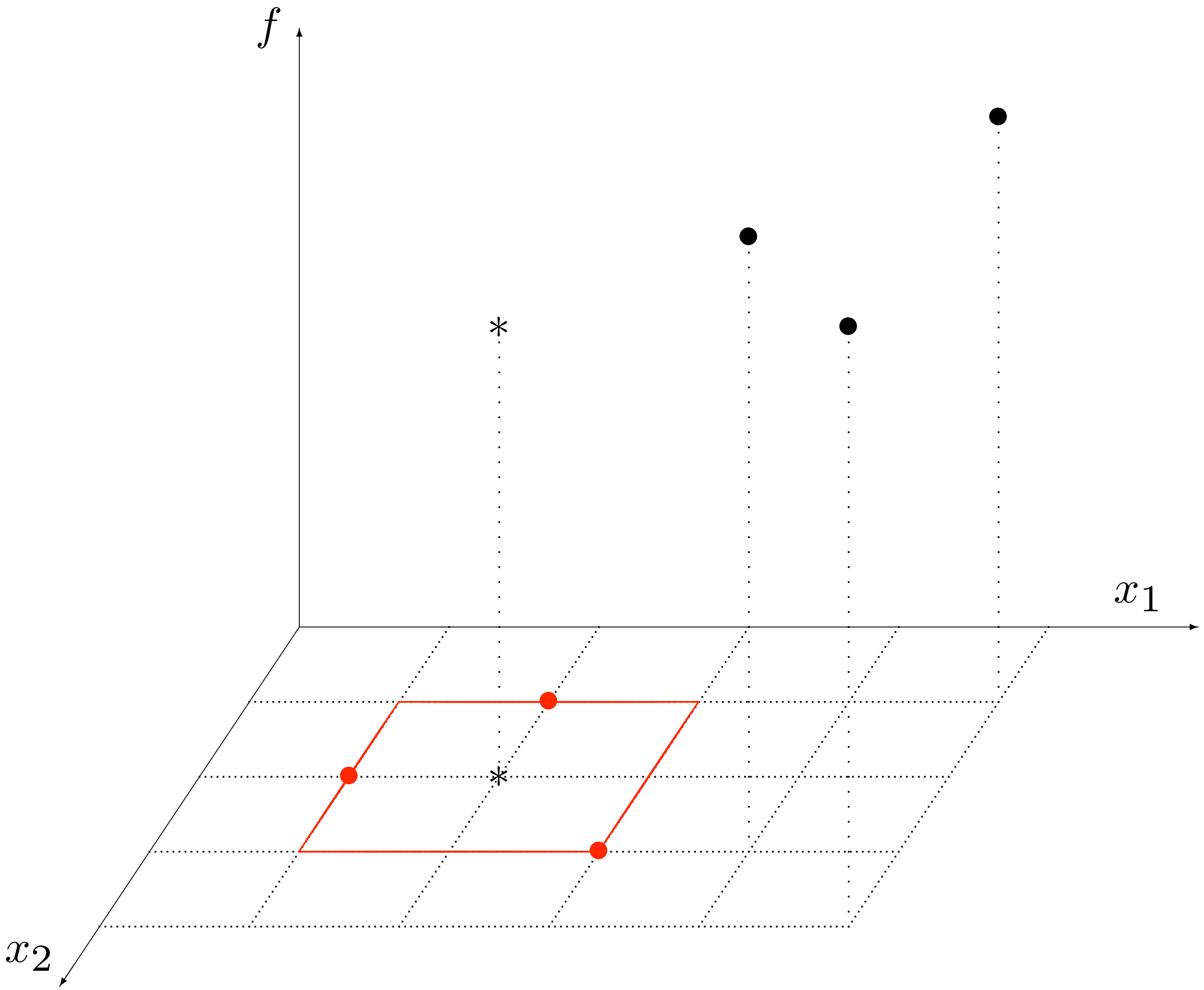


$*$ is still the incumbent solution



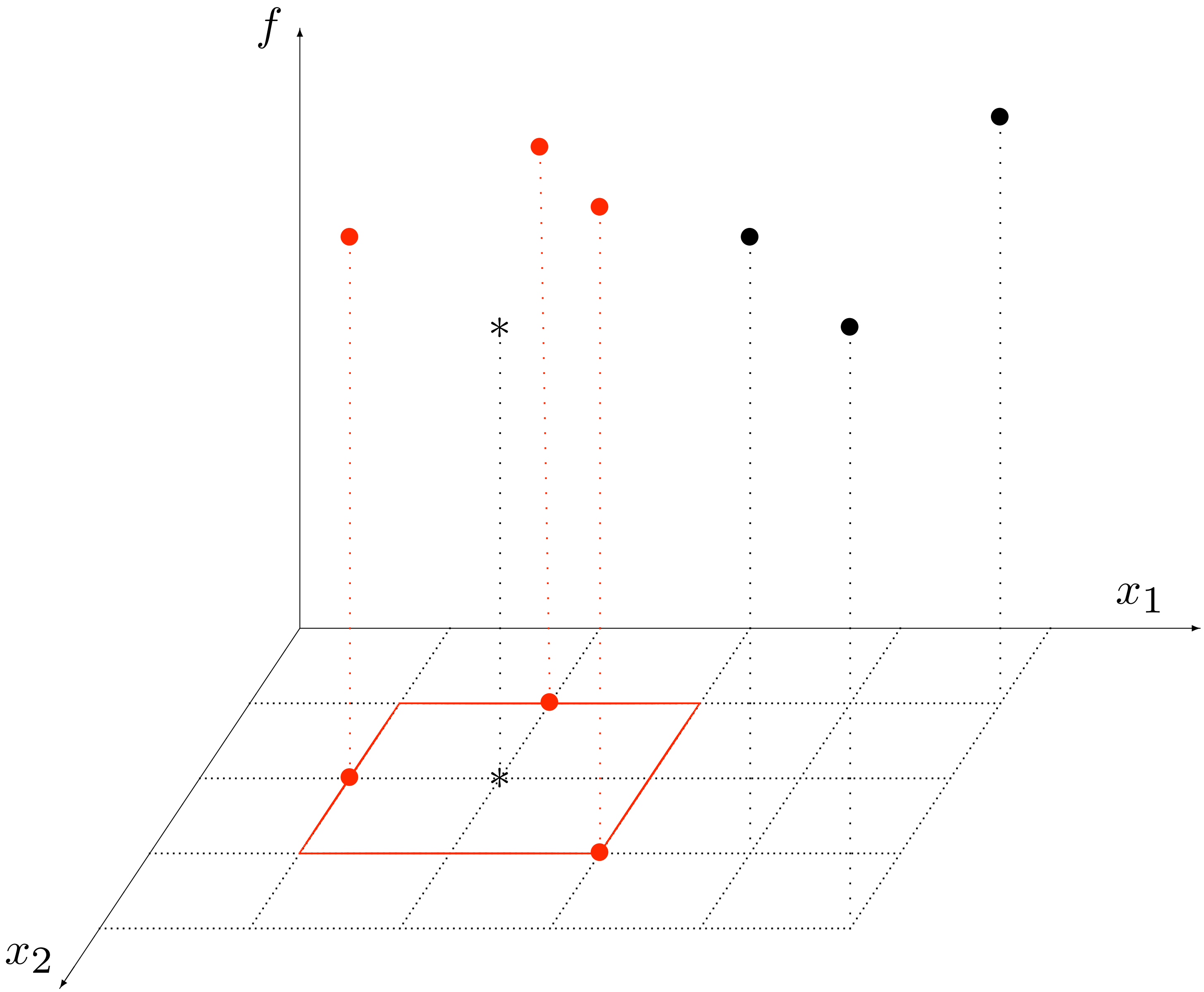
$*$ is still the incumbent solution

We poll near the incumbent solution

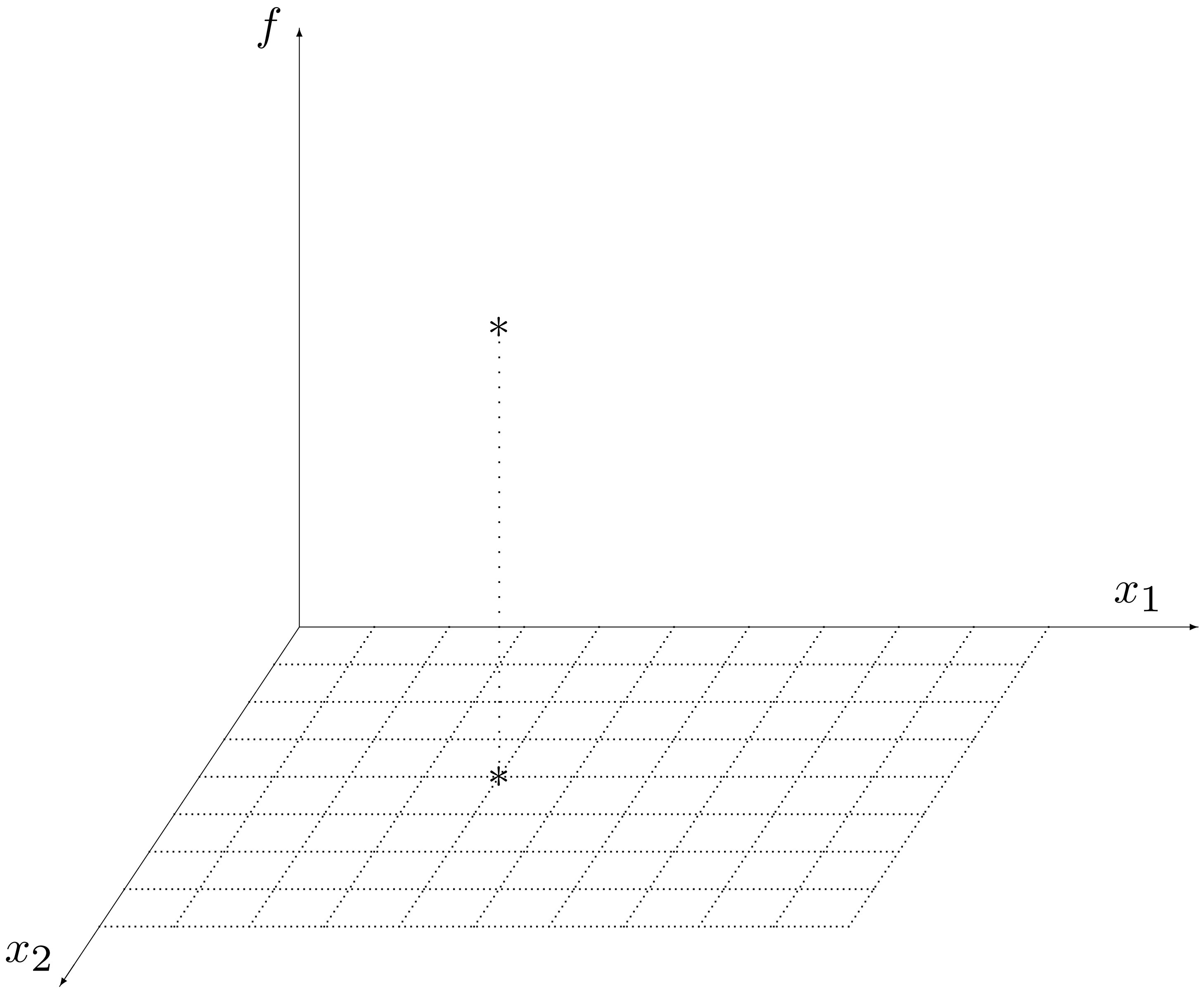


$*$ is still the incumbent solution

We poll near the incumbent solution



$*$ is still the incumbent solution
We poll near the incumbent solution



New iteration from the same incumbent solution, but on a finer mesh

Positive spanning sets and meshes

- A positive spanning set D is a set of vectors whose nonnegative linear combinations span $\mathbb{R}^n$.

Positive spanning sets and meshes

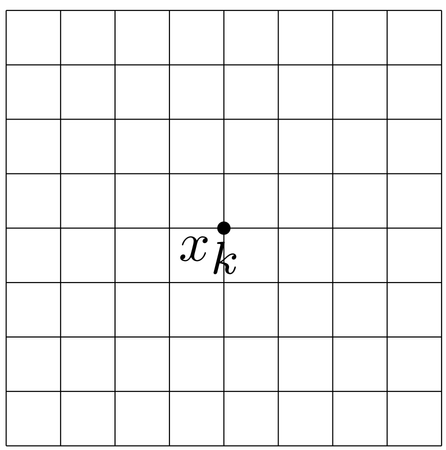
- A positive spanning set D is a set of vectors whose nonnegative linear combinations span $\mathbb{R}^n$.
- The mesh is centered around $x_k \in \mathbb{R}^n$ and its fineness is parameterized by $\Delta_k^m > 0$ as follows

$$M_k = \{x_k + \Delta_k^m D z : z \in \mathbb{N}^{|D|}\}.$$

Positive spanning sets and meshes

- A positive spanning set D is a set of vectors whose nonnegative linear combinations span $\mathbb{R}^n$.
- The mesh is centered around $x_k \in \mathbb{R}^n$ and its fineness is parameterized by $\Delta_k^m > 0$ as follows

$$M_k = \{x_k + \Delta_k^m Dz : z \in \mathbb{N}^{|D|}\}.$$



Ex: $D=[I; \ -I]$

Basic pattern search algorithm for unconstrained optimization

Given Δ_0^m, $x_0 \in M_0$ with $f(x_0) < \infty$, and D,

Basic pattern search algorithm for unconstrained optimization

Given Δ_0^m, $x_0 \in M_0$ with $f(x_0) < \infty$, and D,
for $k = 0, 1, \cdots$, do

1. Employ some finite strategy to try to choose $x_{k+1} \in M_k$ such that $f(x_{k+1}) < f(x_k)$ and then set $\Delta_{k+1}^m = \Delta_k^m$ or $\Delta_{k+1}^m = 2\Delta_k^m$ (x_{k+1} is called an *improved mesh point*);

Basic pattern search algorithm for unconstrained optimization

Given Δ_0^m, $x_0 \in M_0$ with $f(x_0) < \infty$, and D,
for $k = 0, 1, \cdots$, do

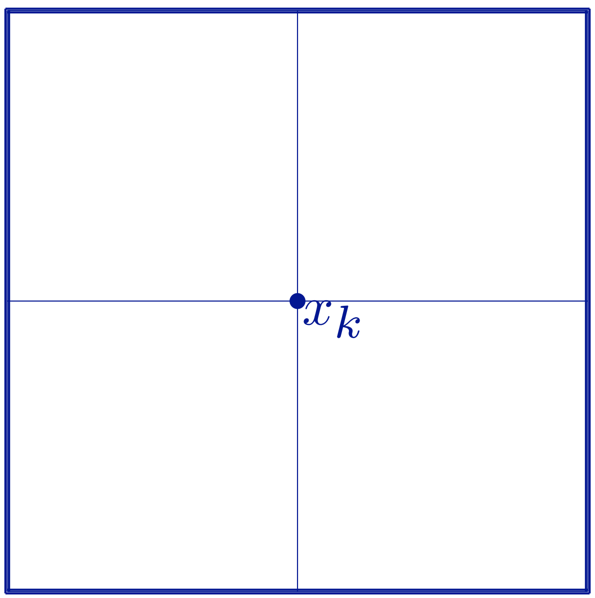
1. Employ some finite strategy to try to choose $x_{k+1} \in M_k$ such that $f(x_{k+1}) < f(x_k)$ and then set $\Delta_{k+1}^m = \Delta_k^m$ or $\Delta_{k+1}^m = 2\Delta_k^m$ (x_{k+1} is called an *improved mesh point*);

2. Else if x_k minimizes $f(x)$ for $x \in P_k$, then set $x_{k+1} = x_k$ and $\Delta_{k+1}^m = \Delta_k^m/2$ (x_k is called a *minimal frame center*).

The Coordinate Search (CS) frame P_k

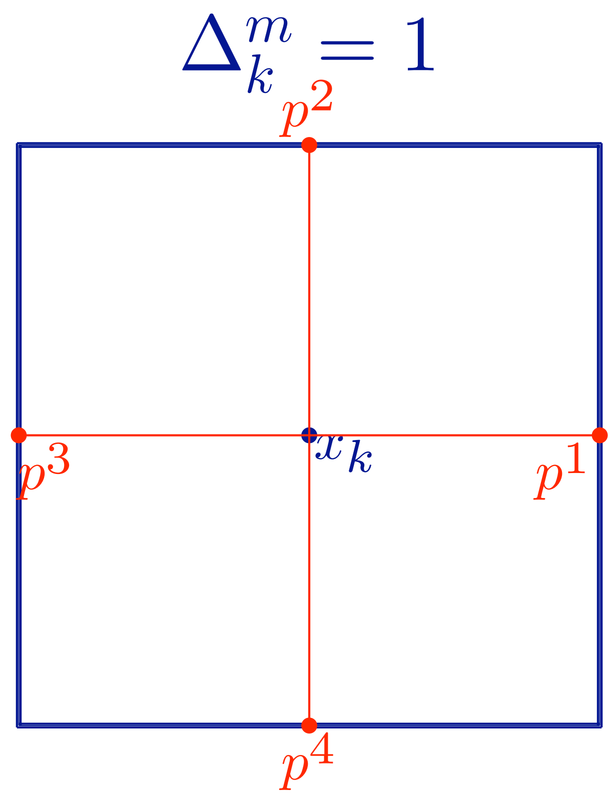
$P_k = \{x_k + \Delta_k^m d : d \in [I; -I]\}$;
$2n$ points adjacent to x_k in M_k.

$\Delta_k^m = 1$



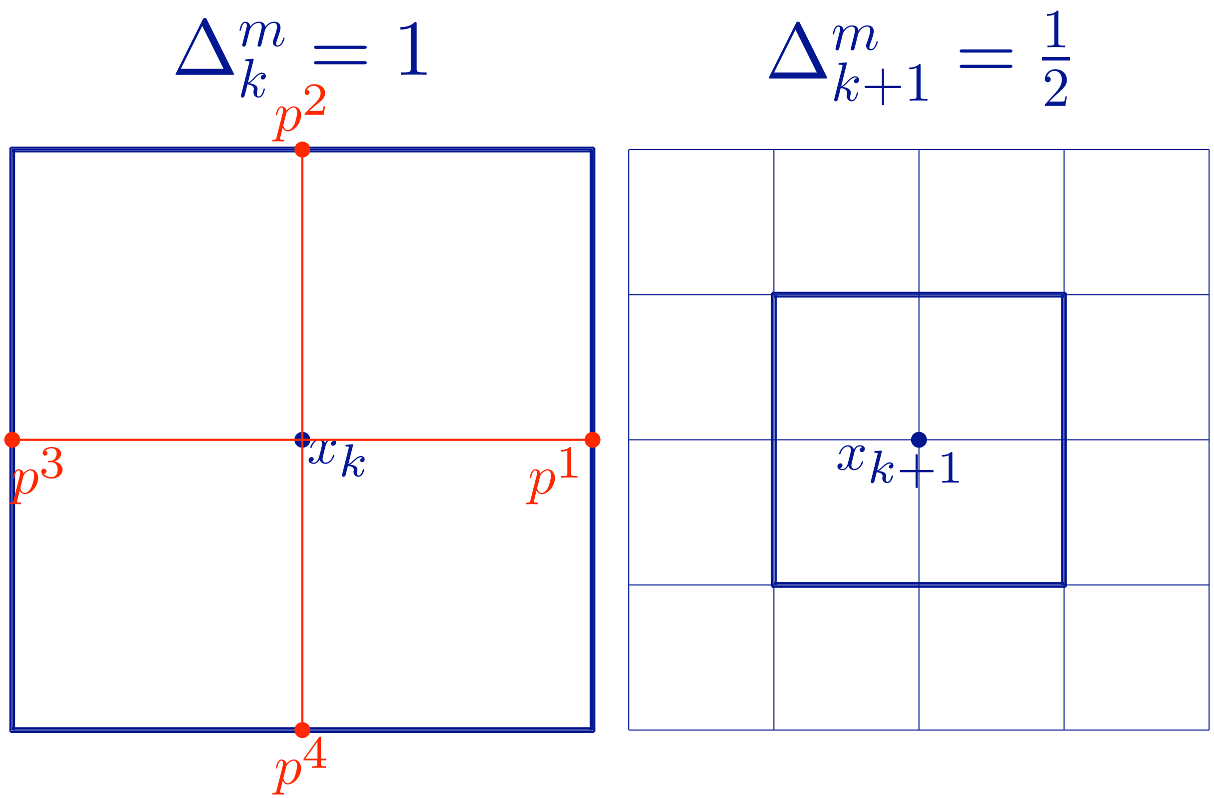
The Coordinate Search (CS) frame P_k

$P_k = \{x_k + \Delta_k^m d : d \in [I; -I]\}$;
$2n$ points adjacent to x_k in M_k.



The Coordinate Search (CS) frame P_k

$P_k = \{x_k + \Delta_k^m d : d \in [I; -I]\}$;
$2n$ points adjacent to x_k in M_k.

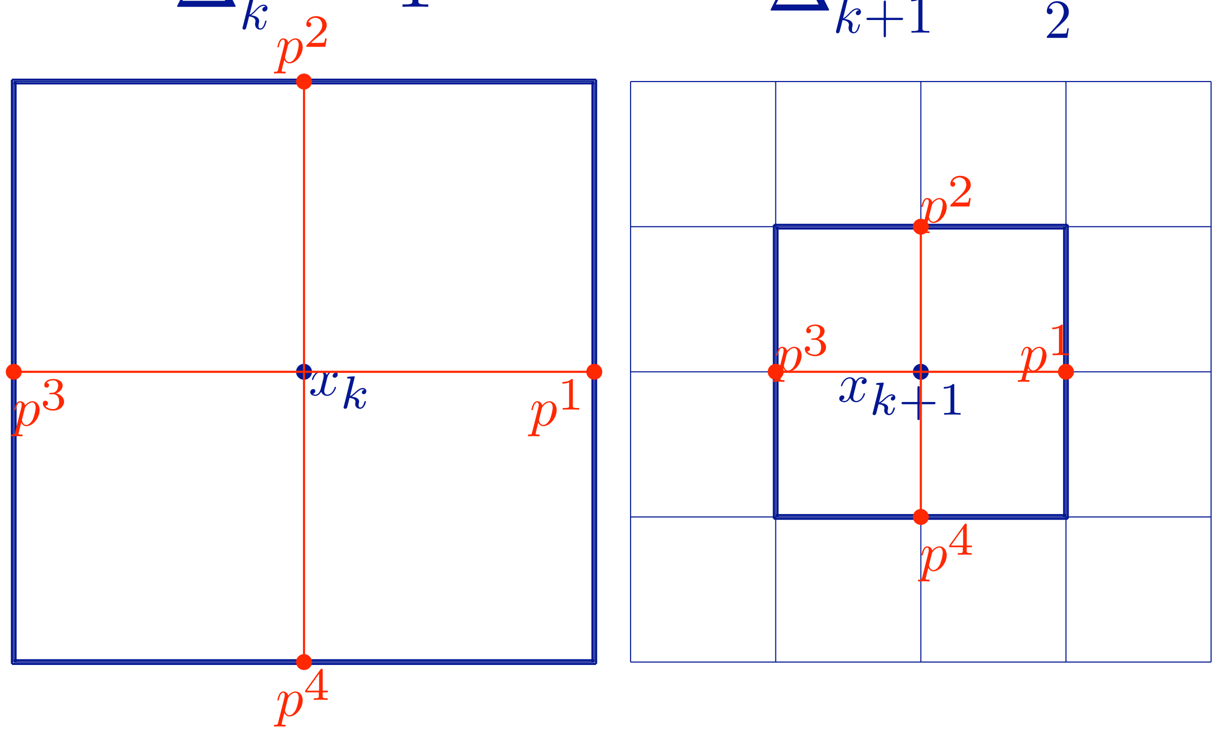


The Coordinate Search (CS) frame P_k

$P_k = \{x_k + \Delta_k^m d : d \in [I; -I]\}$;

$2n$ points adjacent to x_k in M_k.

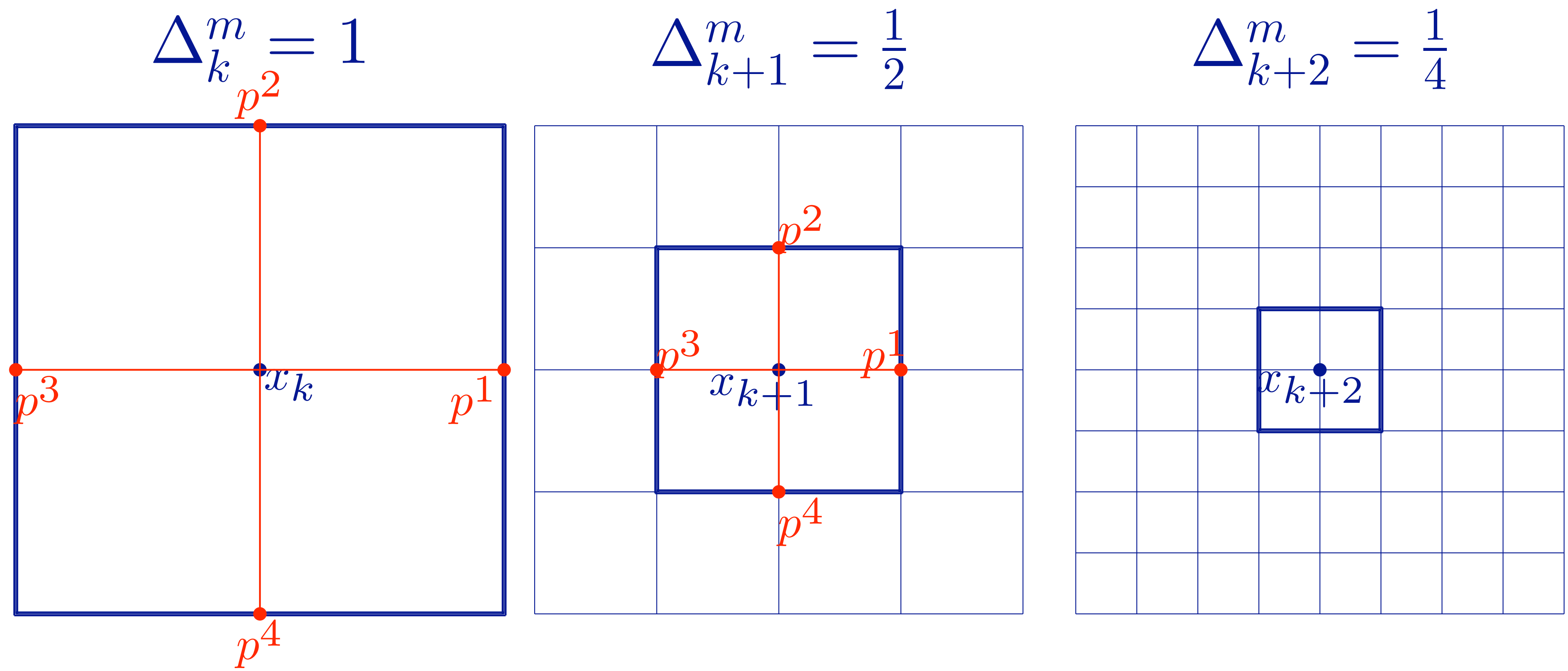
$\Delta_k^m = 1$ $\Delta_{k+1}^m = \frac{1}{2}$



The Coordinate Search (CS) frame P_k

$P_k = \{x_k + \Delta_k^m d : d \in [I; -I]\}$;

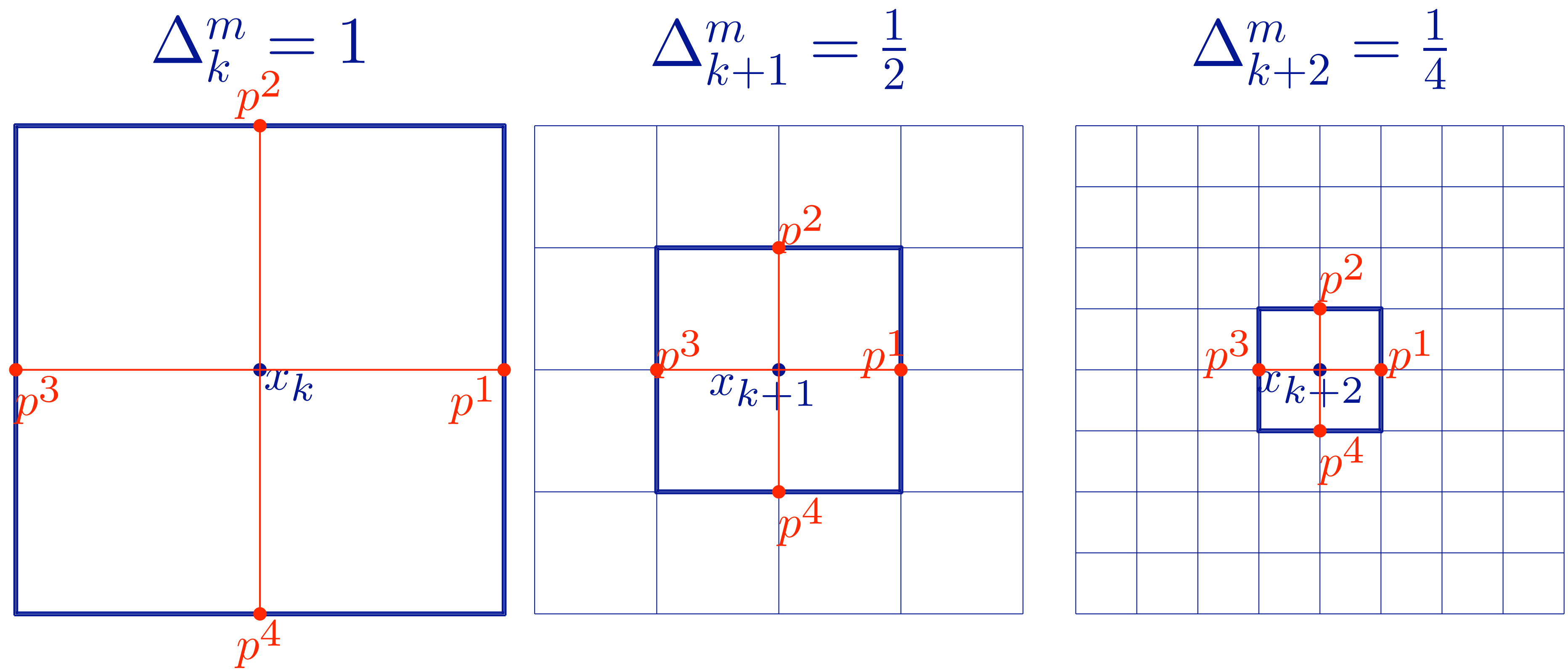
$2n$ points adjacent to x_k in M_k.



The Coordinate Search (CS) frame P_k

$P_k = \{x_k + \Delta_k^m d : d \in [I; -I]\}$;

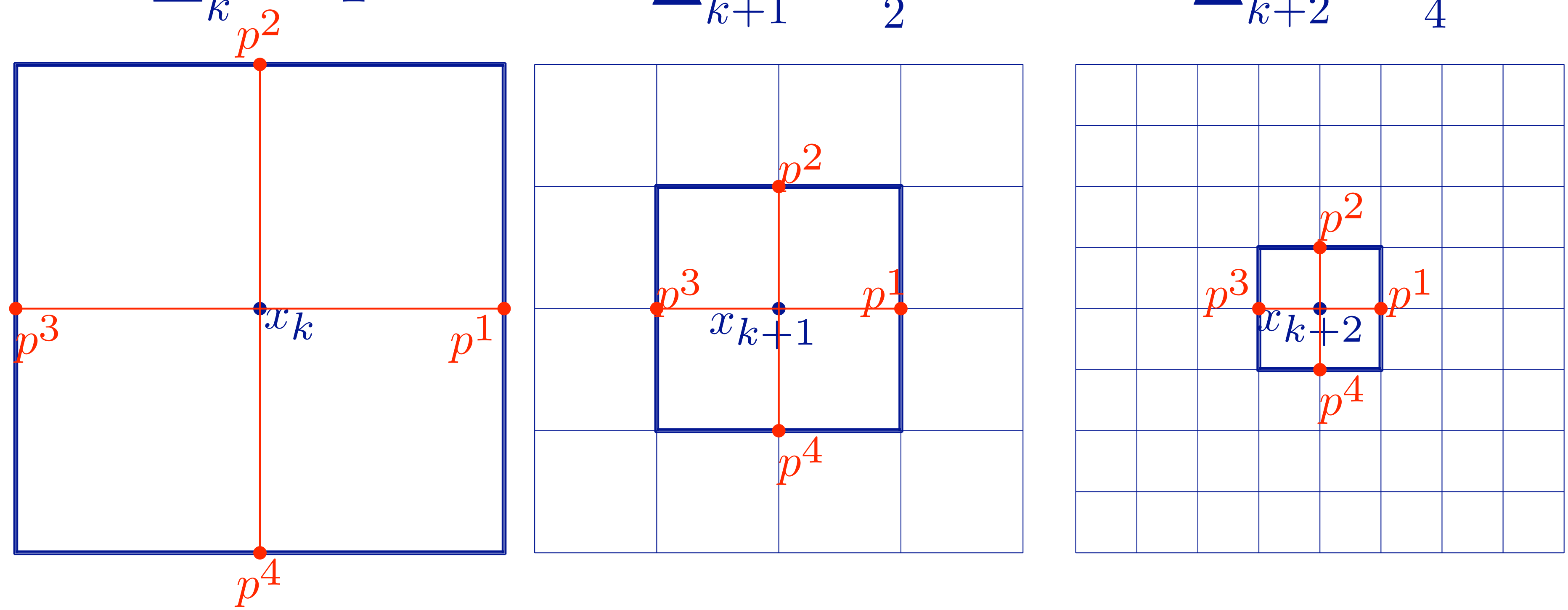
$2n$ points adjacent to x_k in M_k.



The Coordinate Search (CS) frame P_k

$P_k = \{x_k + \Delta_k^m d : d \in [I; -I]\}$;
$2n$ points adjacent to x_k in M_k.

$\Delta_k^m = 1$ $\quad$ $\Delta_{k+1}^m = \frac{1}{2}$ $\quad$ $\Delta_{k+2}^m = \frac{1}{4}$

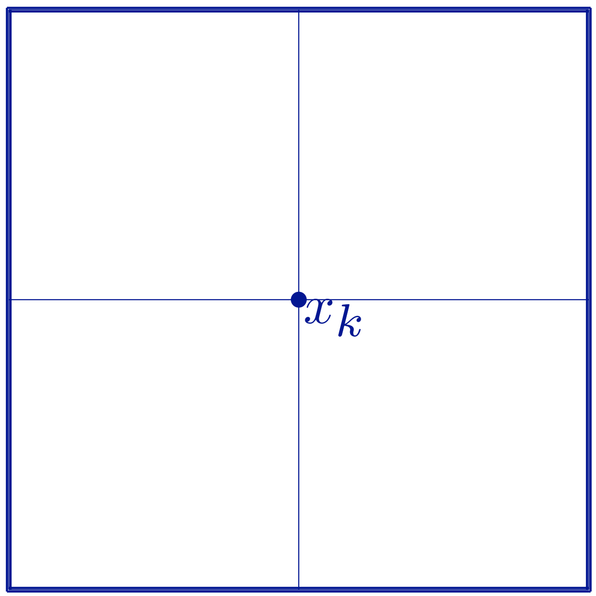


Always the same $2n = 4$ directions, regardless of Δ_k.

The GPS frame P_k

$P_k = \{x_k + \Delta_k^m d : d \in D_k \subset D\}$; points adjacent to x_k in M_k (wrt positive spanning set D_k).

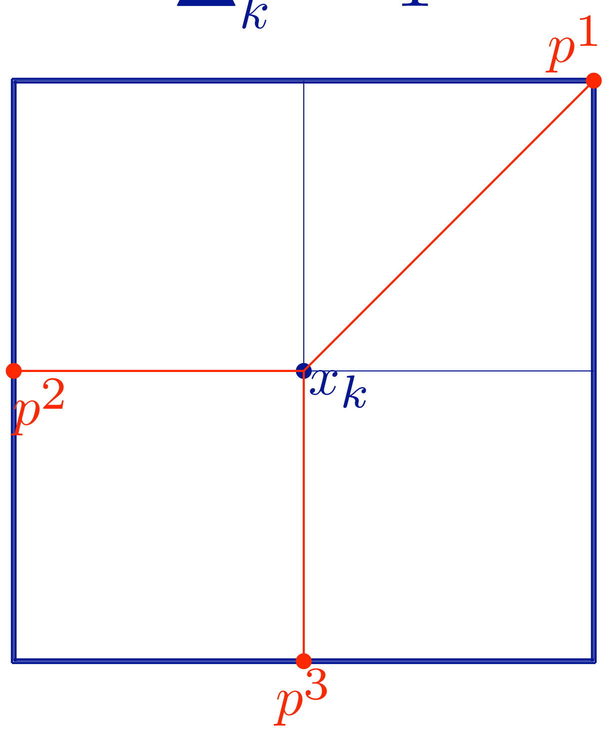
$\Delta_k^m = 1$



The GPS frame P_k

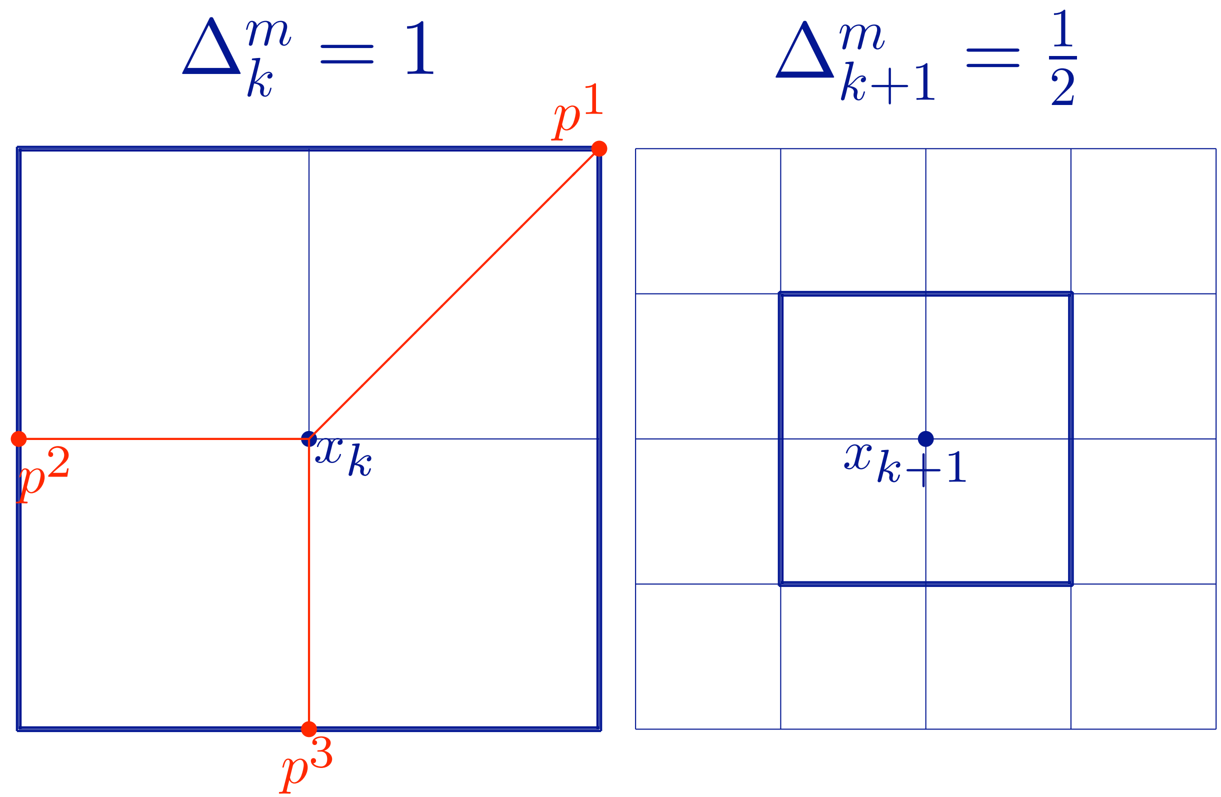
$P_k = \{x_k + \Delta_k^m d : d \in D_k \subset D\}$; points adjacent to x_k in M_k (wrt positive spanning set D_k).

$\Delta_k^m = 1$



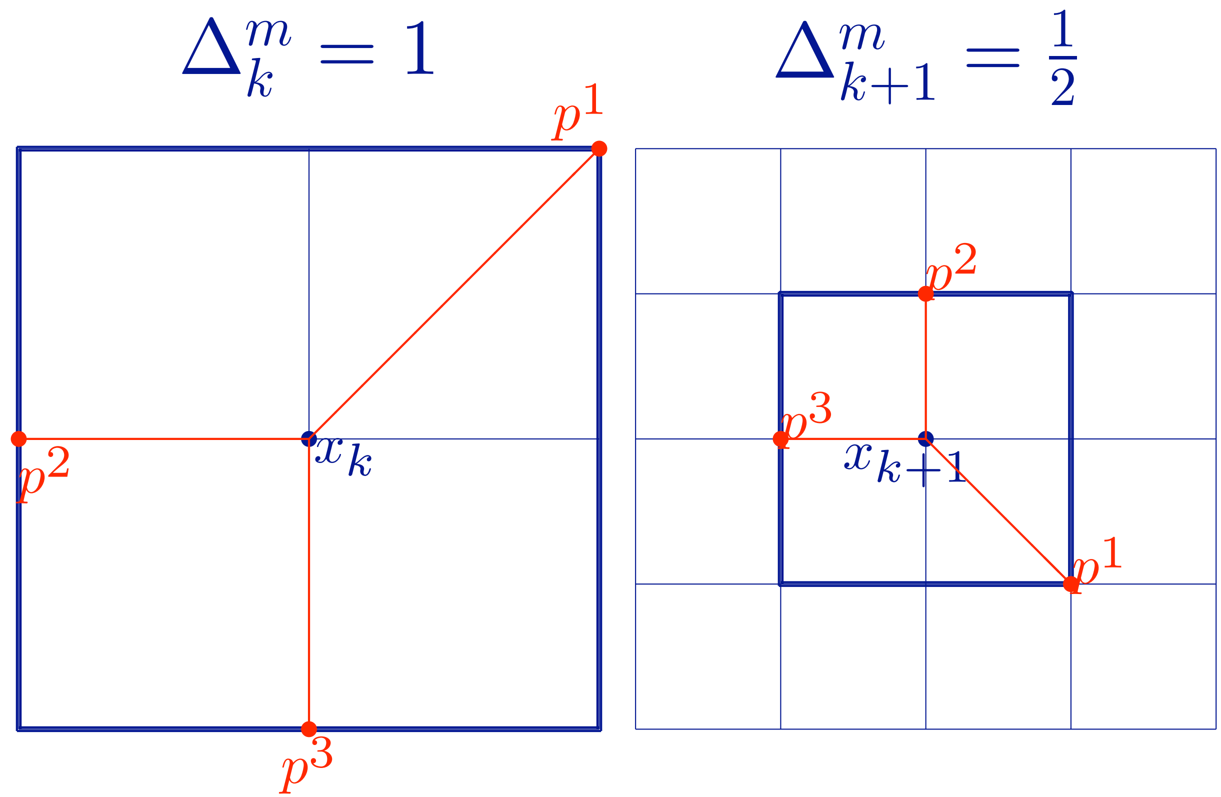
The GPS frame P_k

$P_k = \{x_k + \Delta_k^m d : d \in D_k \subset D\}$; points adjacent to x_k in M_k (wrt positive spanning set D_k).



The GPS frame P_k

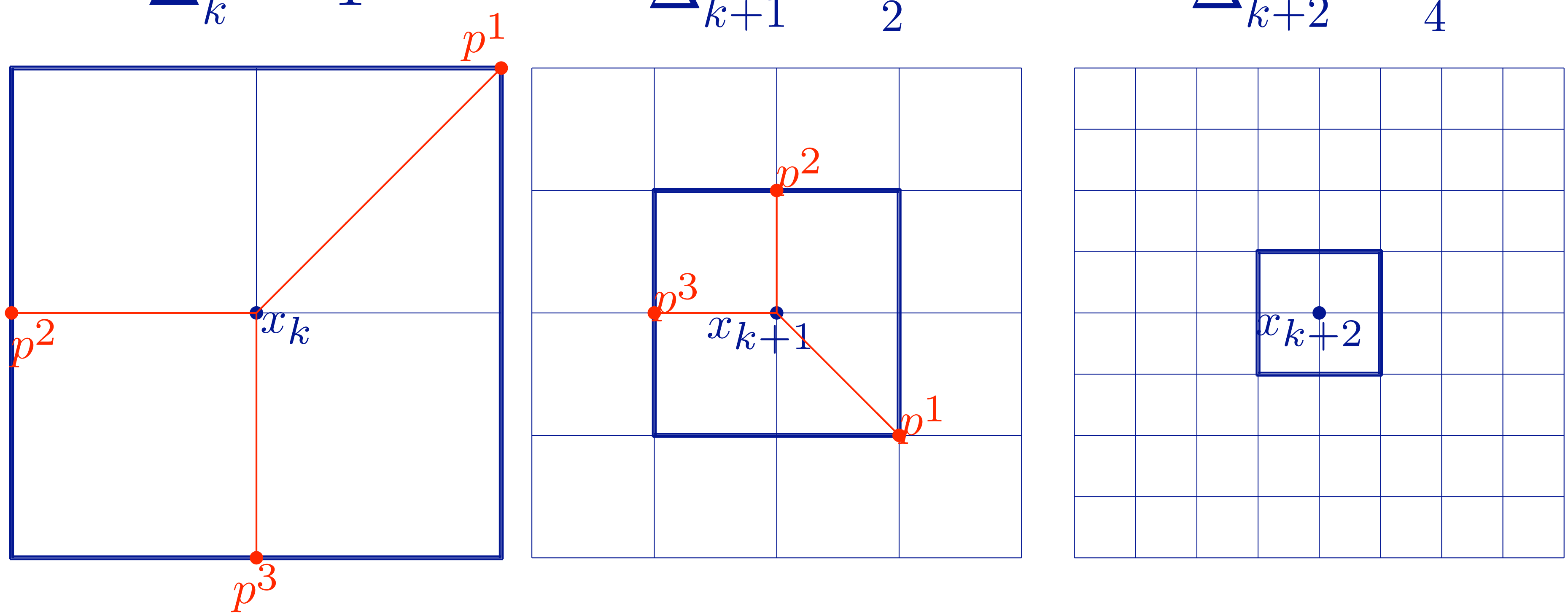
$P_k = \{x_k + \Delta_k^m d : d \in D_k \subset D\}$; points adjacent to x_k in M_k (wrt positive spanning set D_k).



The GPS frame P_k

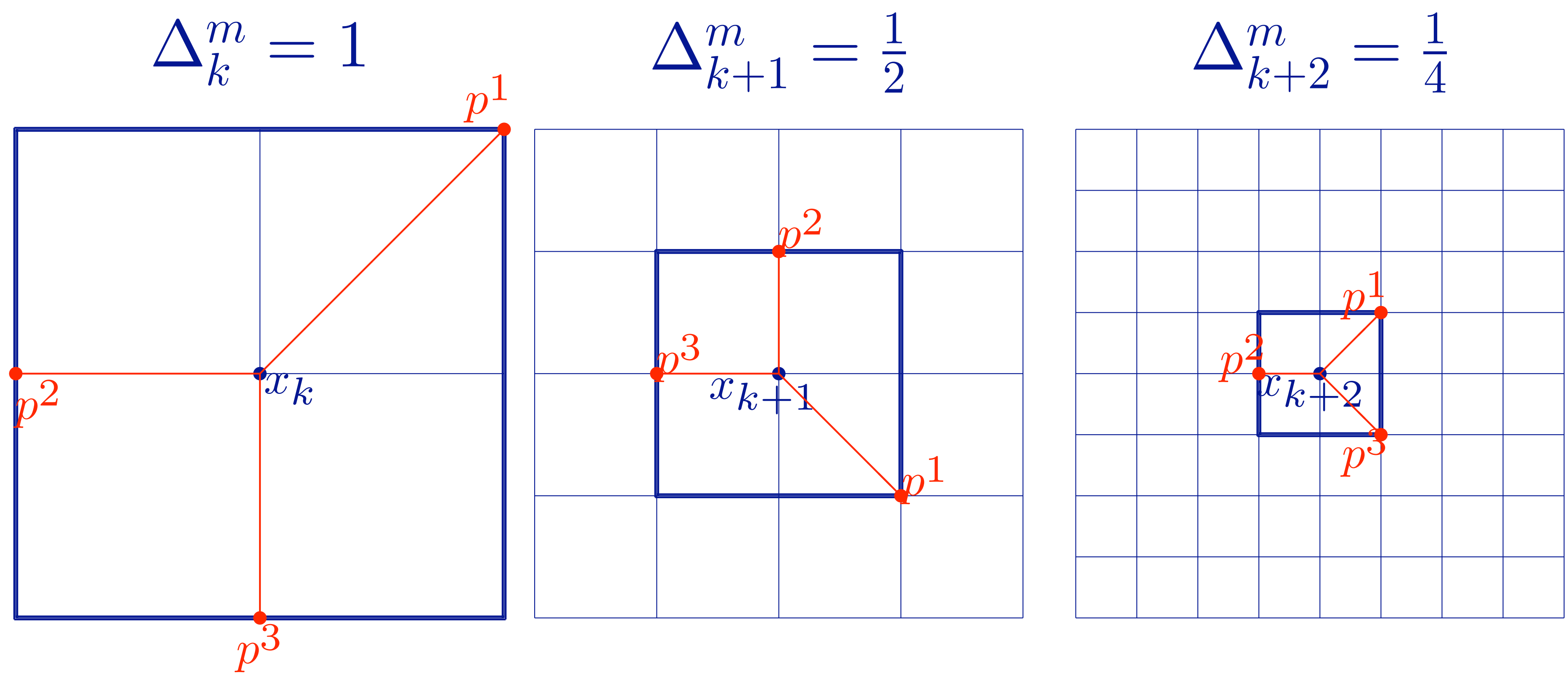
$P_k = \{x_k + \Delta_k^m d : d \in D_k \subset D\}$; points adjacent to x_k in M_k (wrt positive spanning set D_k).

$\Delta_k^m = 1$ $\Delta_{k+1}^m = \frac{1}{2}$ $\Delta_{k+2}^m = \frac{1}{4}$



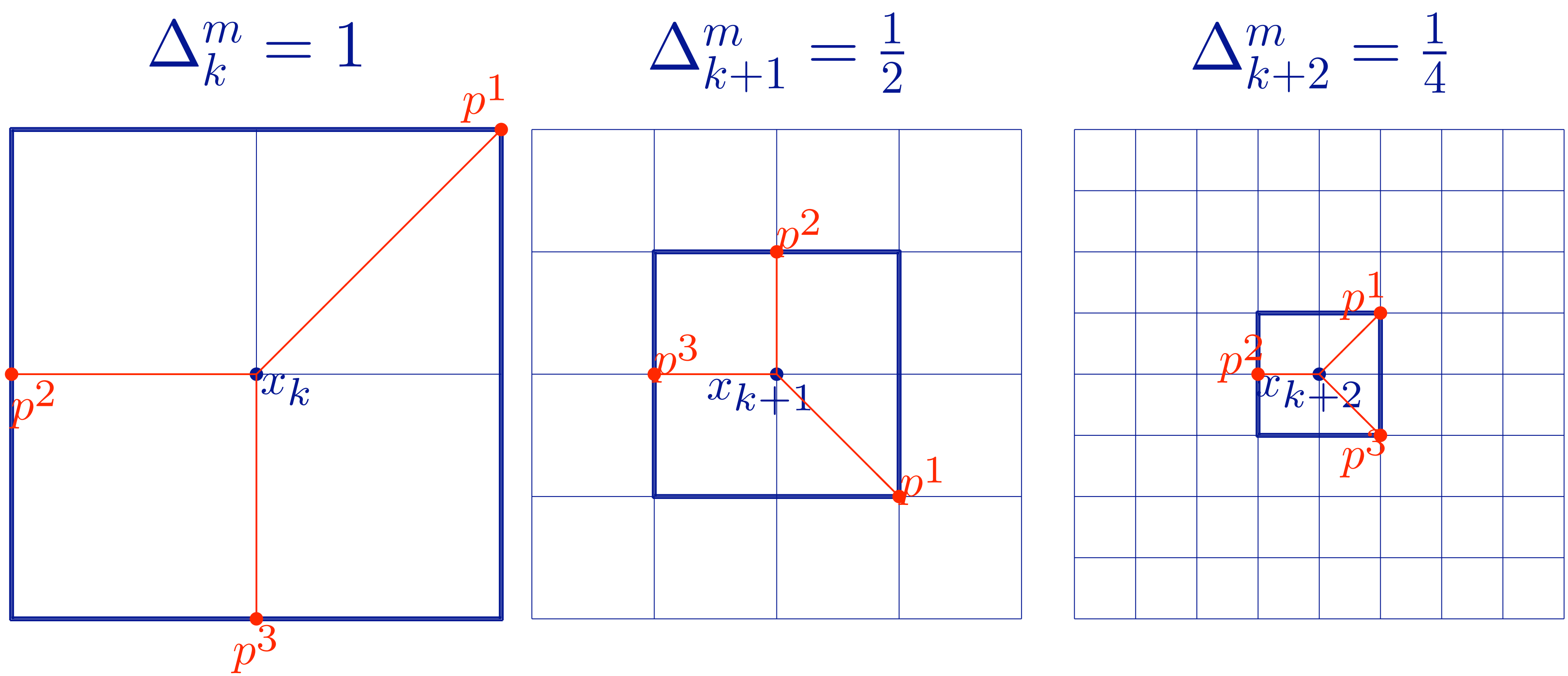
The GPS frame P_k

$P_k = \{x_k + \Delta_k^m d : d \in D_k \subset D\}$; points adjacent to x_k in M_k (wrt positive spanning set D_k).



The GPS frame P_k

$P_k = \{x_k + \Delta_k^m d : d \in D_k \subset D\}$; points adjacent to x_k in M_k (wrt positive spanning set D_k).



Here, only 14 different ways of selecting D_k, regardless of Δ_k.

Outline

- Statement of the optimization problem
- The GPS and MADS algorithm classes
- GPS theory and limiting examples
- The MADS algorithm class
  - An implementable MADS instance
  - MADS theory
  - Numerical results
- Discussion

Convergence results – unconstrained GPS

If all iterates are in a compact set, then $\liminf_k \Delta_k^m = 0$

Convergence results – unconstrained GPS

If all iterates are in a compact set, then $\liminf_{k} \Delta_k^m = 0$

- The mesh is refined only at a minimal frame center.

Convergence results – unconstrained GPS

If all iterates are in a compact set, then $\liminf_{k} \Delta_k^m = 0$

- The mesh is refined only at a minimal frame center.
- There is a limit point $\hat{x}$ of a subsequence $\{x_k\}_{k \in K}$ of minimal frame centers with $\{\Delta_k^m\}_{k \in K} \to 0$.

Convergence results – unconstrained GPS

If all iterates are in a compact set, then $\liminf_k \Delta_k^m = 0$

- The mesh is refined only at a minimal frame center.
- There is a limit point $\hat{x}$ of a subsequence $\{x_k\}_{k \in K}$ of minimal frame centers with $\{\Delta_k^m\}_{k \in K} \to 0$.
  $\{x_k\}_{k \in K}$ is called a refining subsequence.

Convergence results – unconstrained GPS

If all iterates are in a compact set, then $\liminf_k \Delta_k^m = 0$

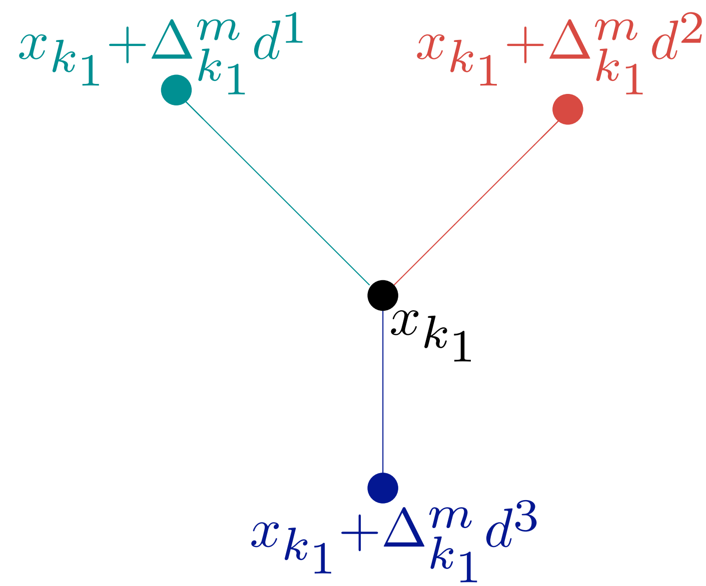
- The mesh is refined only at a minimal frame center.
- There is a limit point $\hat{x}$ of a subsequence $\{x_k\}_{k \in K}$ of minimal frame centers with $\{\Delta_k^m\}_{k \in K} \to 0$.
  $\{x_k\}_{k \in K}$ is called a refining subsequence.
- $f(x_k) \leq f(x_k + \Delta_k^m d) \; \forall \; d \in D_k \subset D$ with $k \in K$.

Convergence results – unconstrained GPS

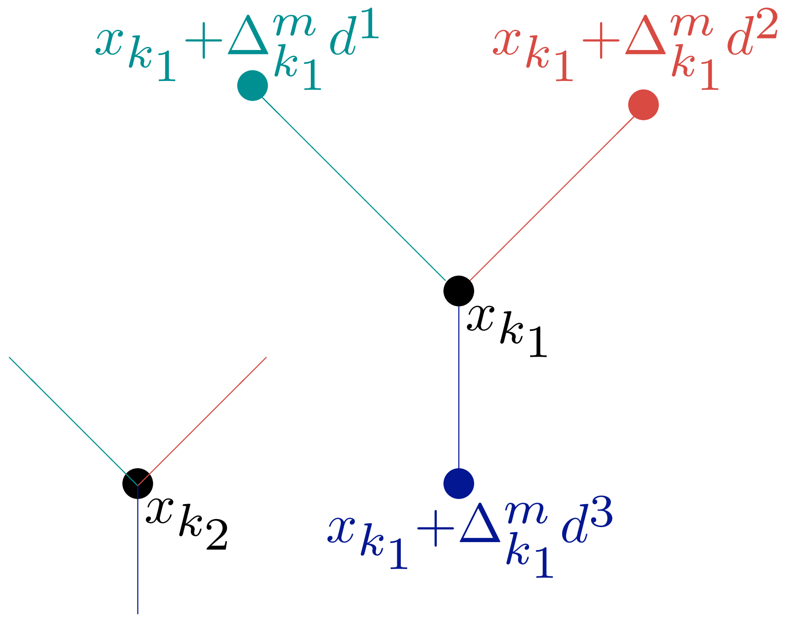
If all iterates are in a compact set, then $\liminf_k \Delta_k^m = 0$

- The mesh is refined only at a minimal frame center.
- There is a limit point $\hat{x}$ of a subsequence $\{x_k\}_{k \in K}$ of minimal frame centers with $\{\Delta_k^m\}_{k \in K} \to 0$.
  $\{x_k\}_{k \in K}$ is called a refining subsequence.
- $f(x_k) \leq f(x_k + \Delta_k^m d)\ \forall\ d \in D_k \subset D$ with $k \in K$.
  Let $\hat{D} \subseteq D$ be the set of POLL directions used infinitely often in the refining subsequence.
  $\hat{D}$ is the set of refining direction.

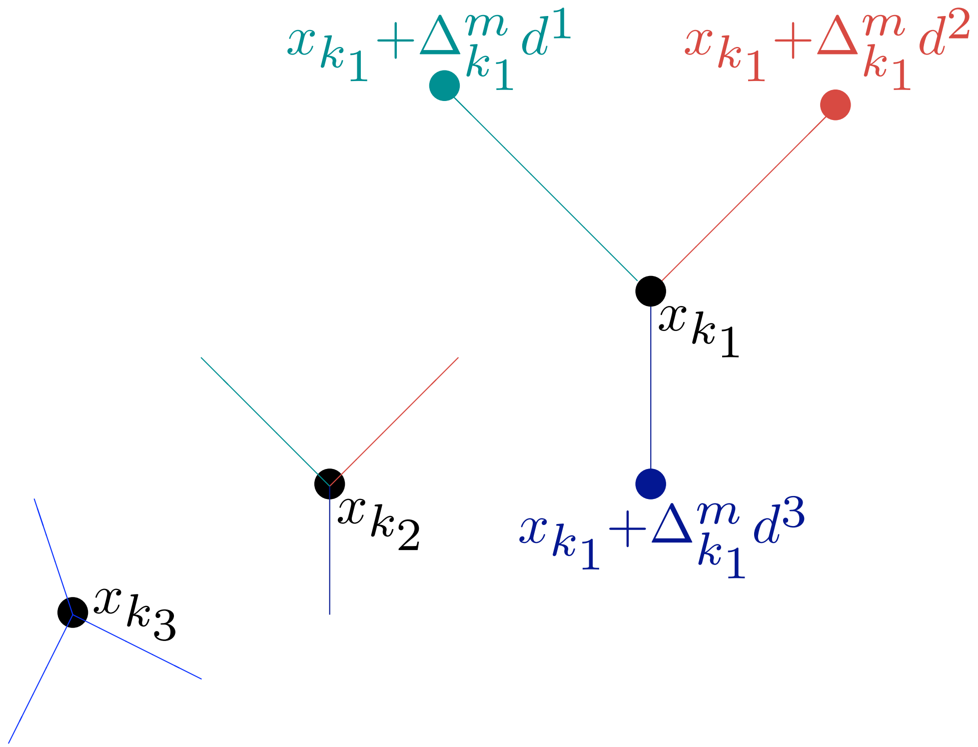
Set of refining directions $\hat{D}$



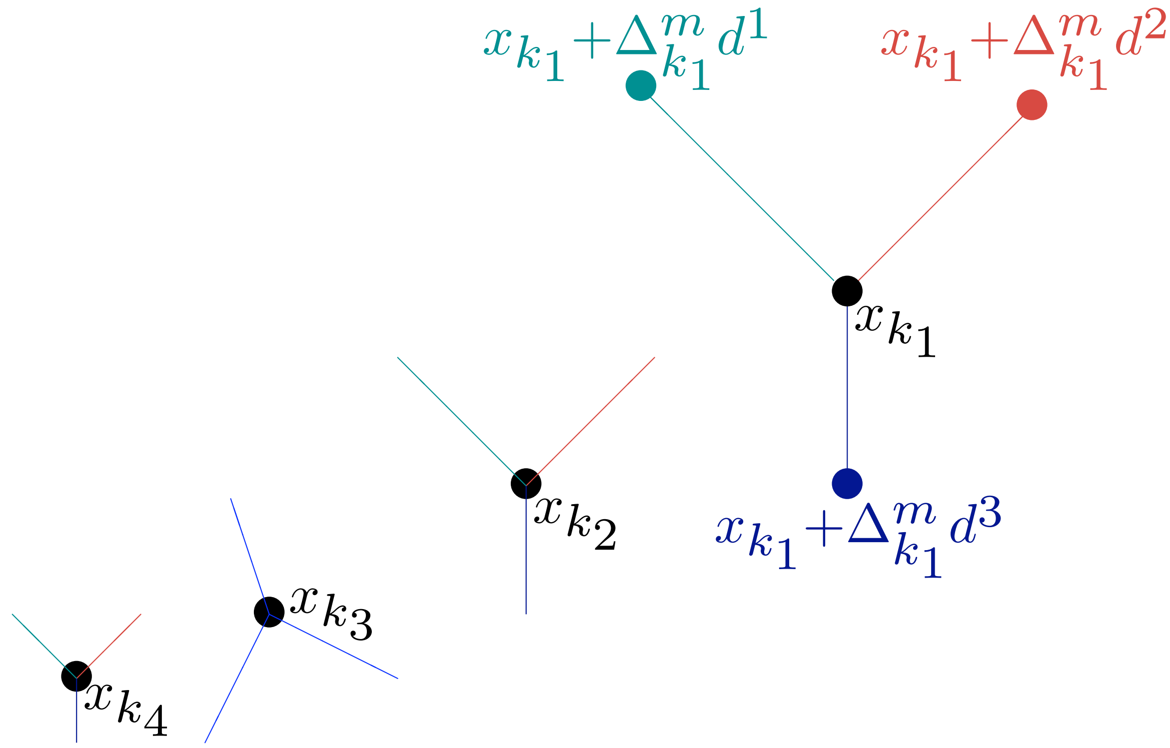
Set of refining directions $\hat{D}$



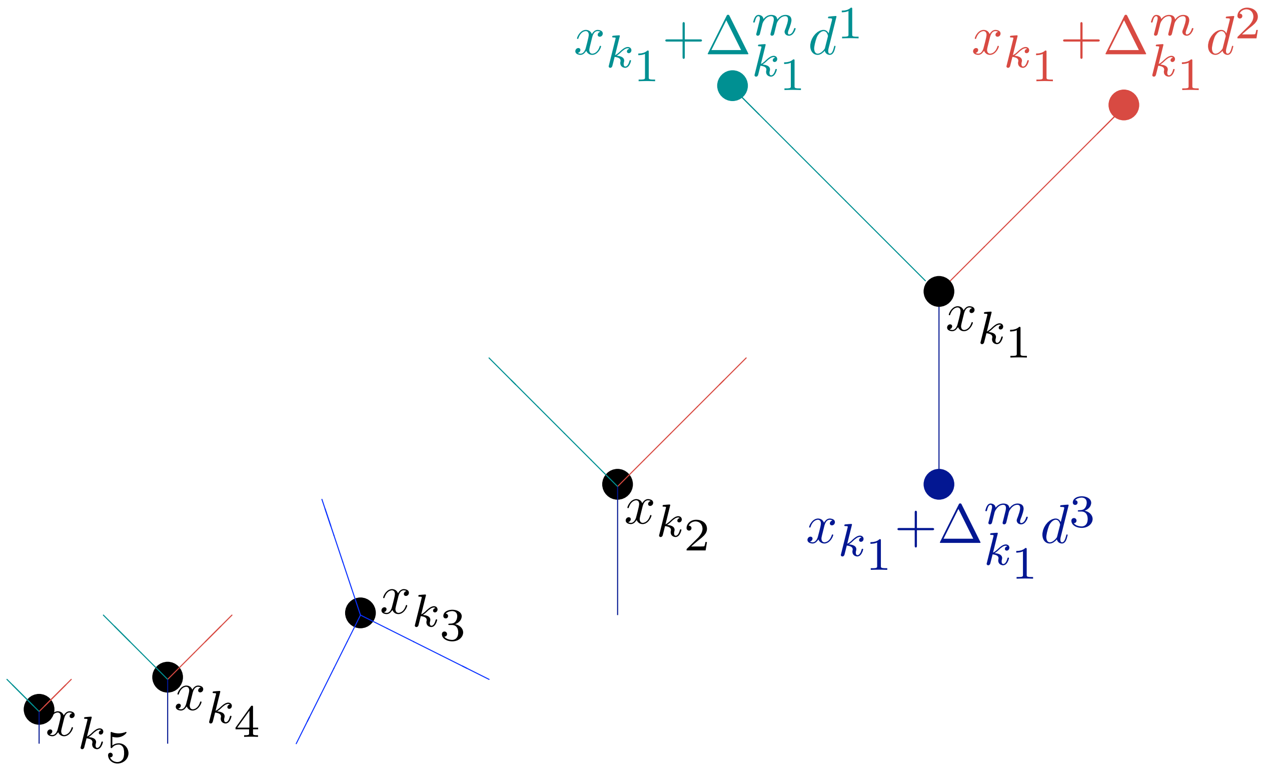
Set of refining directions $\hat{D}$



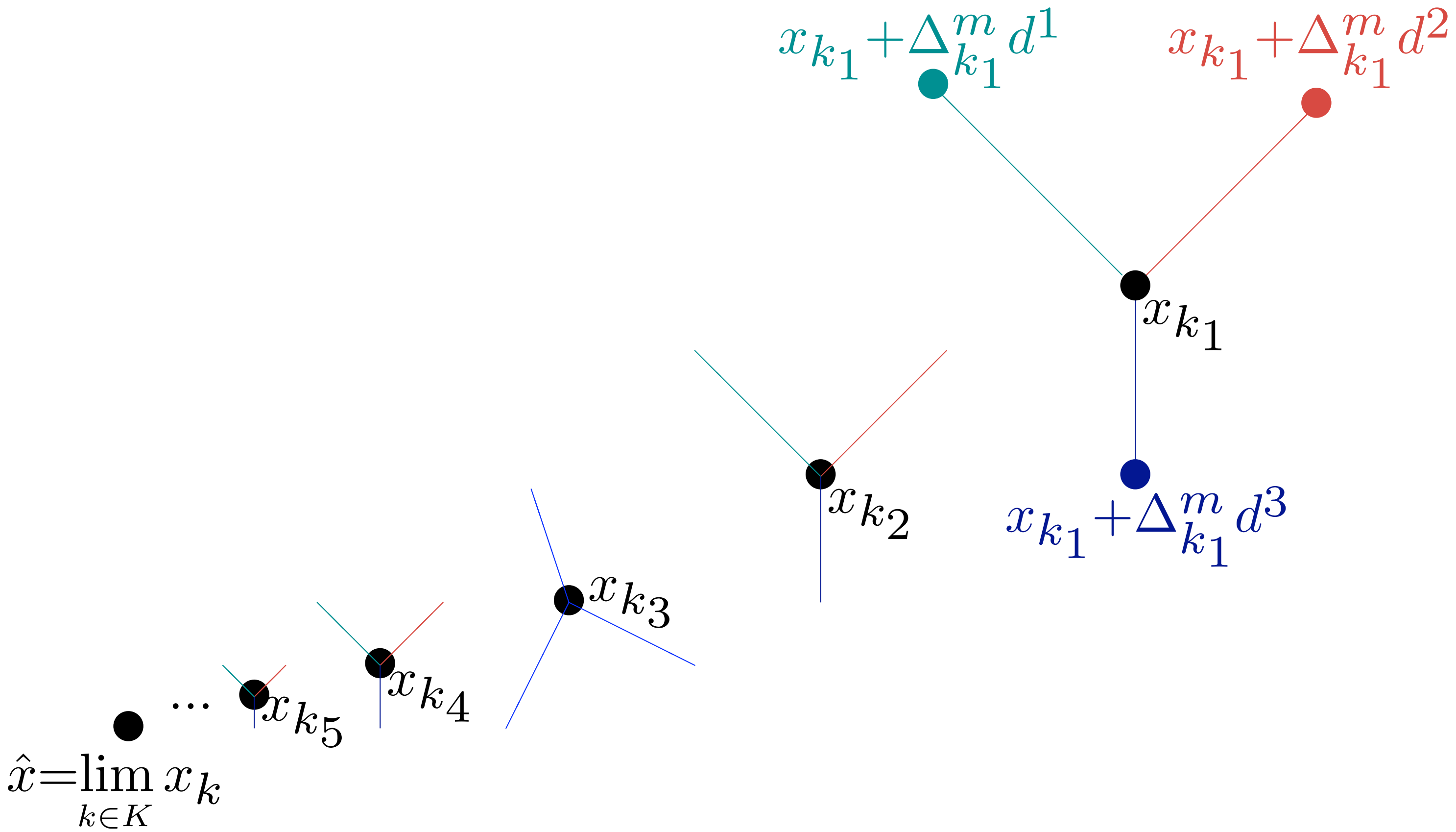
Set of refining directions $\hat{D}$



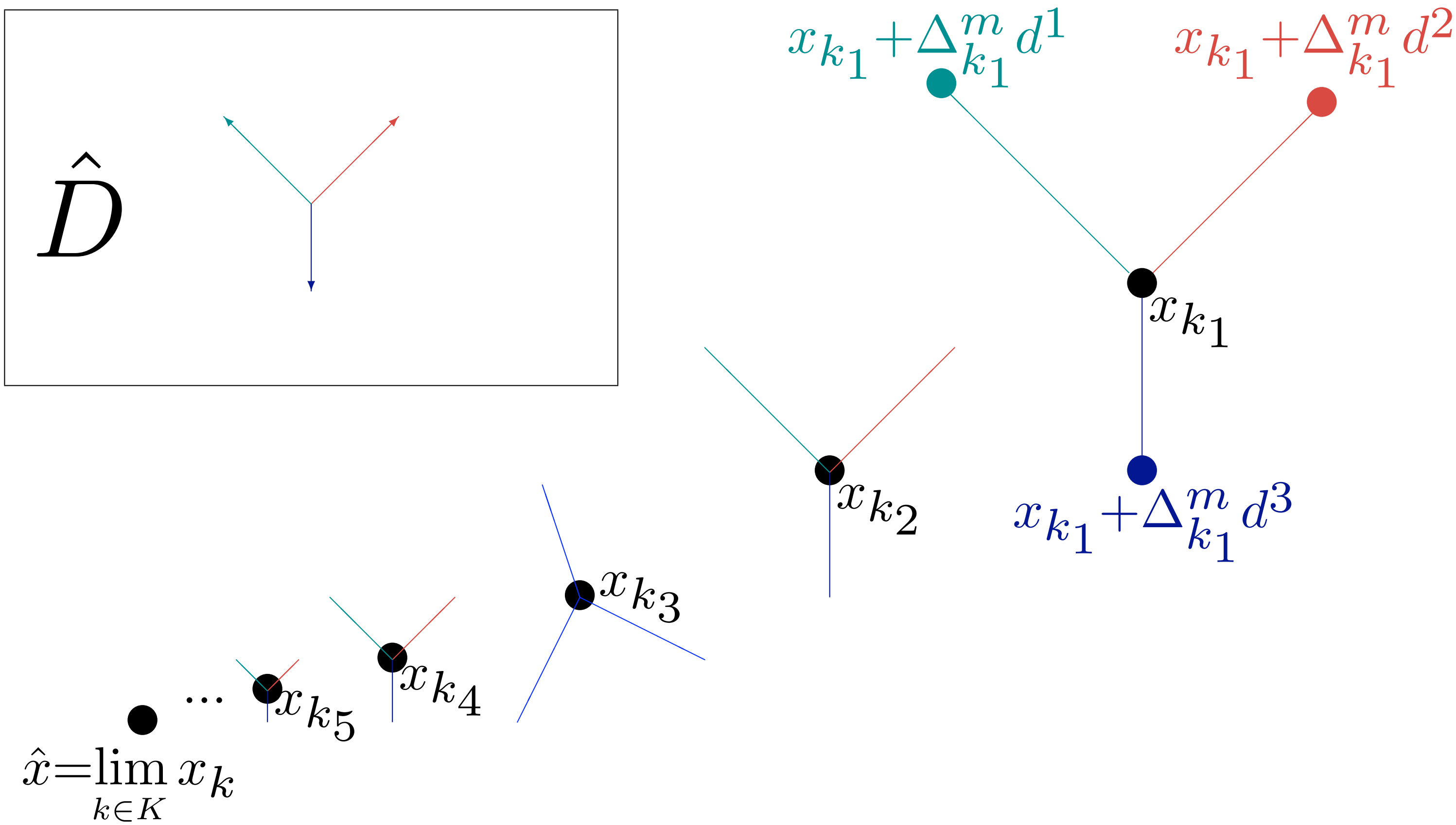
Set of refining directions $\hat{D}$



Set of refining directions $\hat{D}$



Set of refining directions $\hat{D}$



Convergence results – unconstrained GPS

If all iterates are in a compact set, then for any refining subsequence (with limit $\hat{x}$ and refining directions $\hat{D}$)

Convergence results – unconstrained GPS

If all iterates are in a compact set, then for any refining subsequence (with limit $\hat{x}$ and refining directions $\hat{D}$)

- f is Lipschitz near $\hat{x}$ $\Rightarrow f^\circ(\hat{x}; d) \geq 0$ for every $d \in \hat{D}$.

Convergence results – unconstrained GPS

If all iterates are in a compact set, then for any refining subsequence (with limit $\hat{x}$ and refining directions $\hat{D}$)

- f is Lipschitz near $\hat{x}$ $\Rightarrow f^\circ(\hat{x}; d) \geq 0$ for every $d \in \hat{D}$.

  this says that the Clarke derivatives are non-negative on a finite set of directions that positively span $\mathbb{R}^n$.

  $$f^\circ(\hat{x}; d) := \limsup_{y \to \hat{x}, t \downarrow 0} \frac{f(y + td) - f(y)}{t} \geq \lim_{k \in K} \frac{f(x_k + \Delta_k d) - f(x_k)}{\Delta_k}$$

Convergence results – unconstrained GPS

If all iterates are in a compact set, then for any refining subsequence (with limit $\hat{x}$ and refining directions $\hat{D}$)

- f is Lipschitz near $\hat{x}$ $\Rightarrow f^\circ(\hat{x}; d) \geq 0$ for every $d \in \hat{D}$.

  this says that the Clarke derivatives are non-negative on a finite set of directions that positively span $\mathbb{R}^n$.

  $$f^\circ(\hat{x}; d) := \limsup_{y \to \hat{x}, t \downarrow 0} \frac{f(y + td) - f(y)}{t} \geq \lim_{k \in K} \frac{f(x_k + \Delta_k d) - f(x_k)}{\Delta_k}$$

- f is regular at $\hat{x}$ $\Rightarrow f'(\hat{x}; d) \geq 0$ for every $d \in \hat{D}$.

Convergence results – unconstrained GPS

If all iterates are in a compact set, then for any refining subsequence (with limit $\hat{x}$ and refining directions $\hat{D}$)

- f is Lipschitz near $\hat{x}$ $\Rightarrow f^\circ(\hat{x};d) \geq 0$ for every $d \in \hat{D}$.

  this says that the Clarke derivatives are non-negative on a finite set of directions that positively span $\mathbb{R}^n$.

  $$f^\circ(\hat{x};d) := \limsup_{y \to \hat{x}, t \downarrow 0} \frac{f(y+td) - f(y)}{t} \geq \lim_{k \in K} \frac{f(x_k + \Delta_k d) - f(x_k)}{\Delta_k}$$

- f is regular at $\hat{x}$ $\Rightarrow f'(\hat{x};d) \geq 0$ for every $d \in \hat{D}$.

- f is strictly differentiable at $\hat{x}$ $\Rightarrow \nabla f(\hat{x}) = 0$.

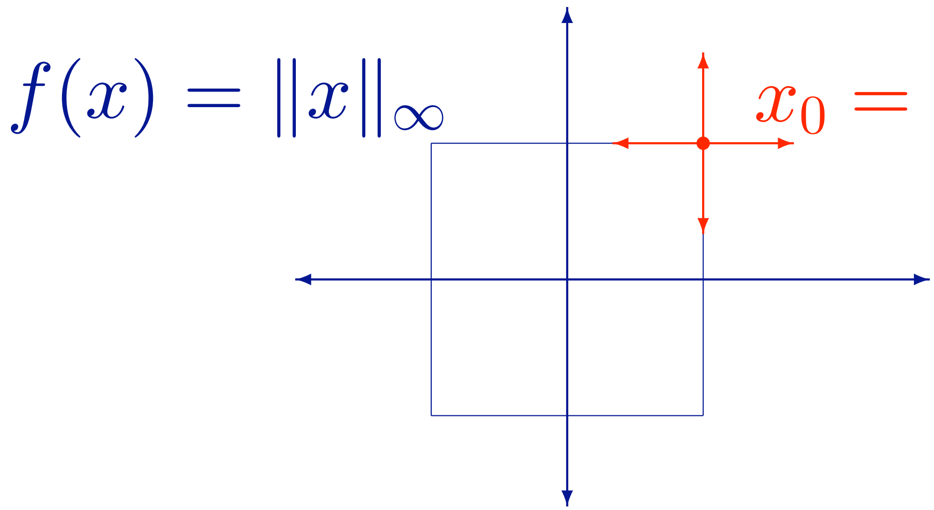
Limitations of GPS

GPS methods are directional, so the restriction to a finite set of directions is a big limitation, particularly when dealing with nonlinear constraints.

Limitations of GPS

GPS methods are directional, so the restriction to a finite set of directions is a big limitation, particularly when dealing with nonlinear constraints.

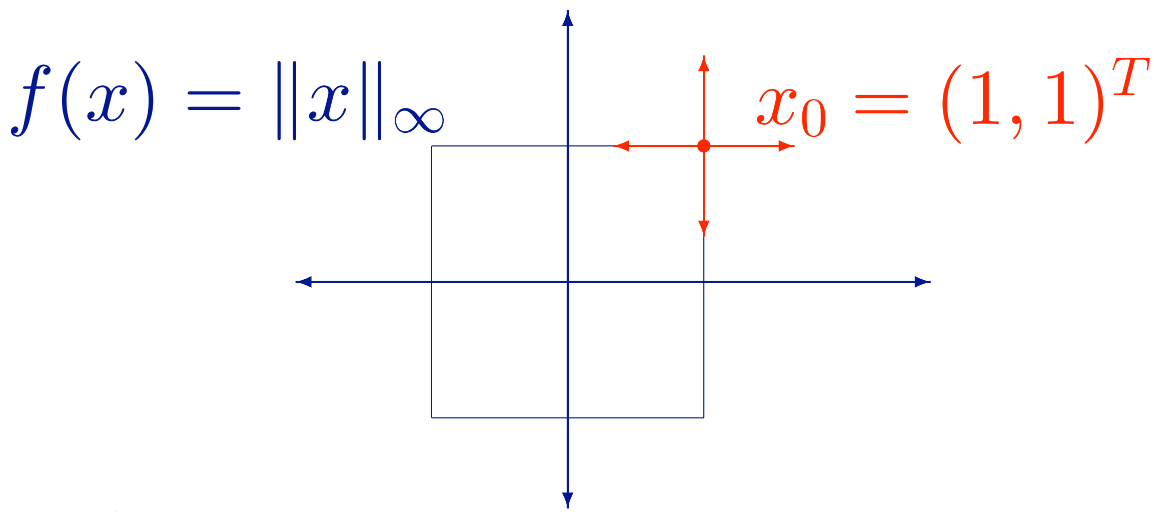
GPS with empty SEARCH: The iterates stall at x_0.



Limitations of GPS

GPS methods are directional, so the restriction to a finite set of directions is a big limitation, particularly when dealing with nonlinear constraints.

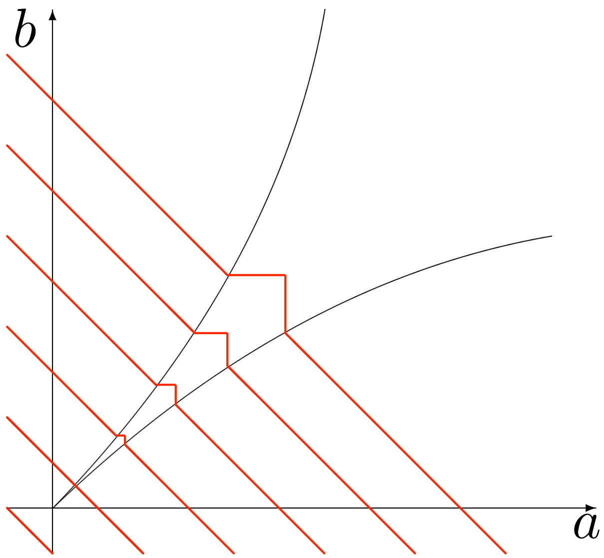
GPS with empty SEARCH: The iterates stall at x_0.



Even with a C^1 function, GPS may generate infinitely many limit points, some of them non-stationary.

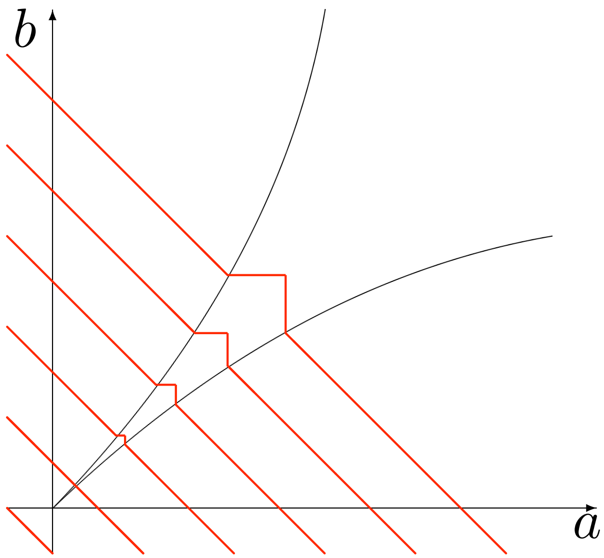
GPS convergence to a bad solution

Level Sets

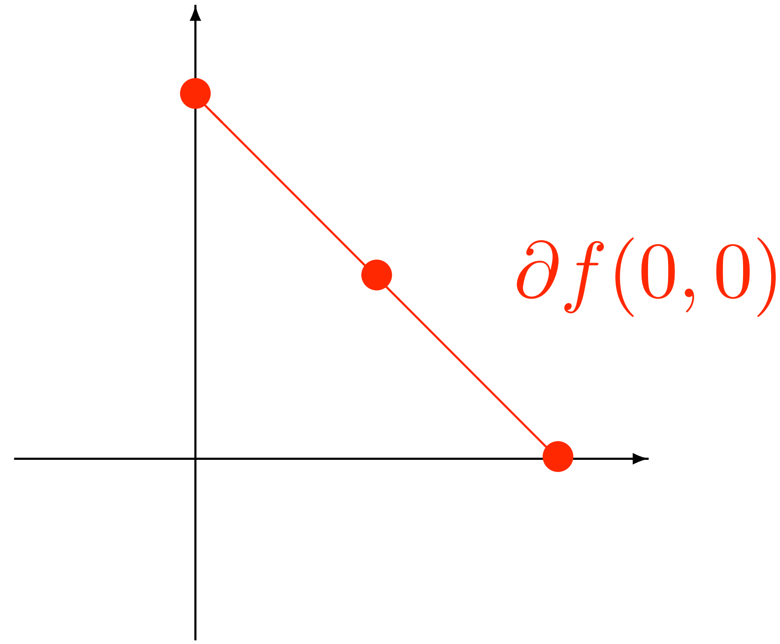


GPS convergence to a bad solution

Level Sets



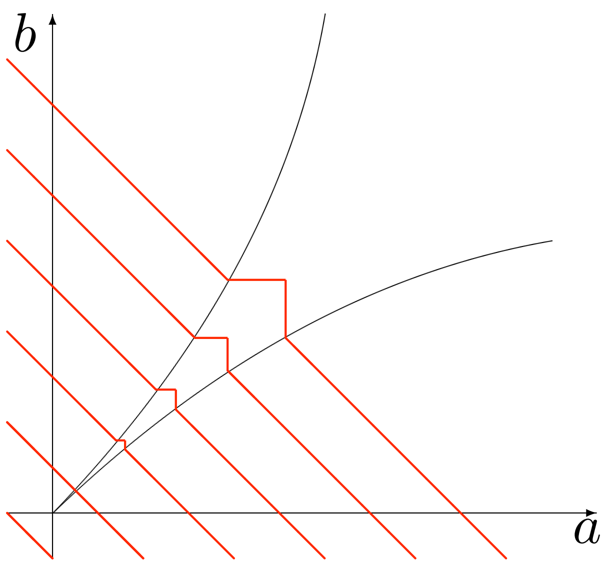
$(0,0) \notin \partial f(0,0)$

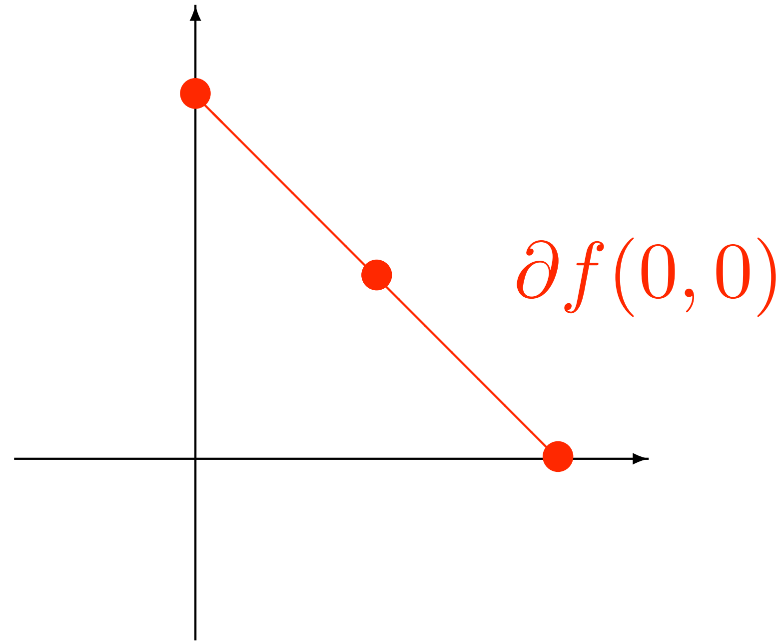


GPS convergence to a bad solution

Level Sets

$(0,0) \notin \partial f(0,0)$





GPS iterates – with a bad strategy – converge to the origin, where the gradient exists and is nonzero (f is differentiable at $(0,0)$ but not strictly differentiable).

Outline

- Statement of the optimization problem
- The GPS and MADS algorithm classes
- GPS theory and limiting examples
- The MADS algorithm class
  - An implementable MADS instance
  - MADS theory
  - Numerical results
- Discussion

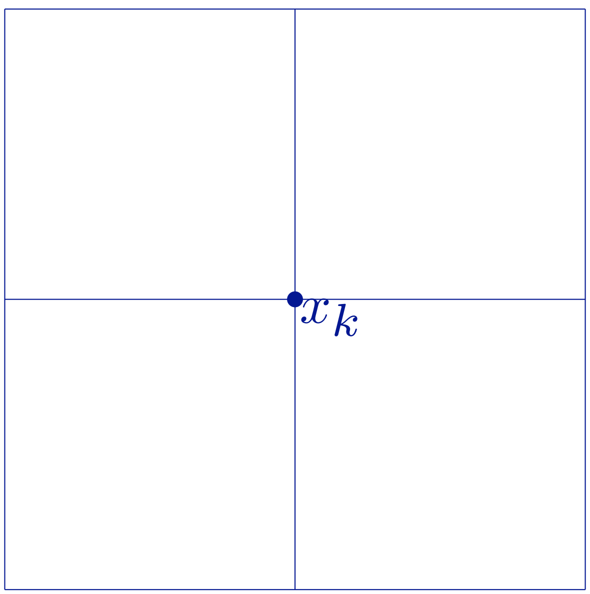
The MADS frames

In addition to the mesh size parameter Δ_k^m, we introduce the poll size parameter Δ_k^p.

The MADS frames

In addition to the mesh size parameter Δ_k^m, we introduce the poll size parameter Δ_k^p. Ex : $\Delta_k^p = n\sqrt{\Delta_k^m}$.

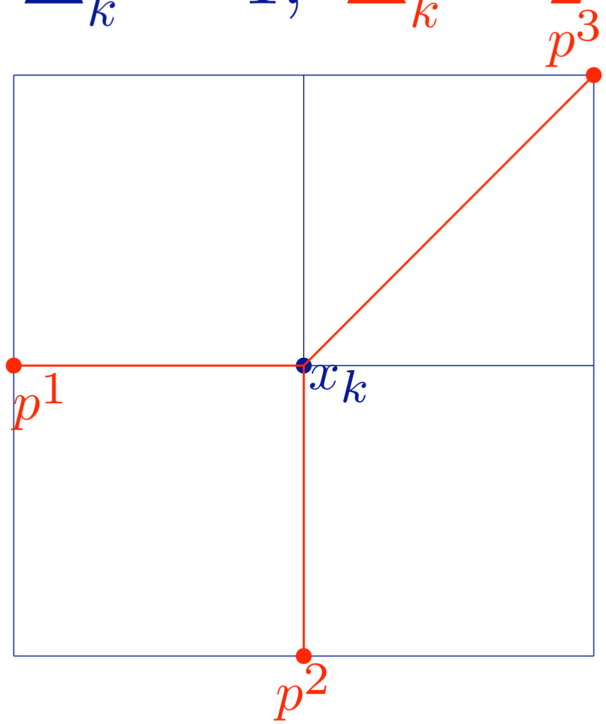
$\Delta_k^m = 1,\ \Delta_k^p = 2$



The MADS frames

In addition to the mesh size parameter Δ_k^m, we introduce the poll size parameter Δ_k^p. Ex : $\Delta_k^p = n\sqrt{\Delta_k^m}$.

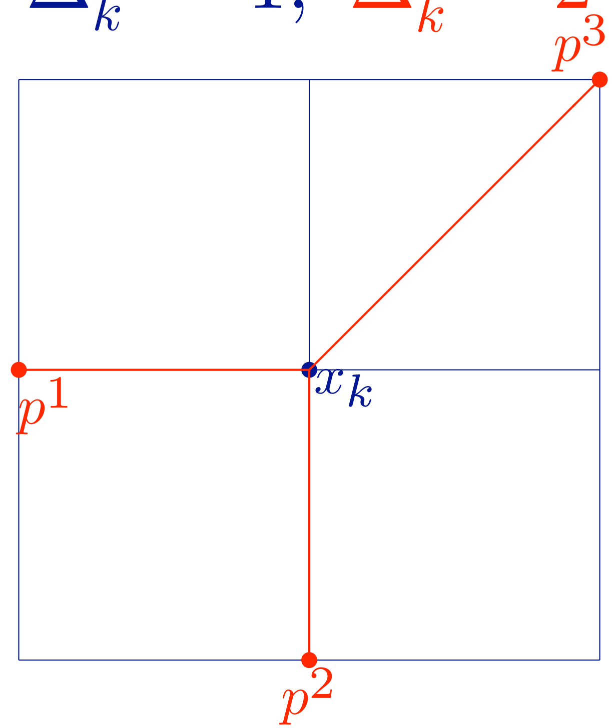
$\Delta_k^m = 1, \ \Delta_k^p = 2$

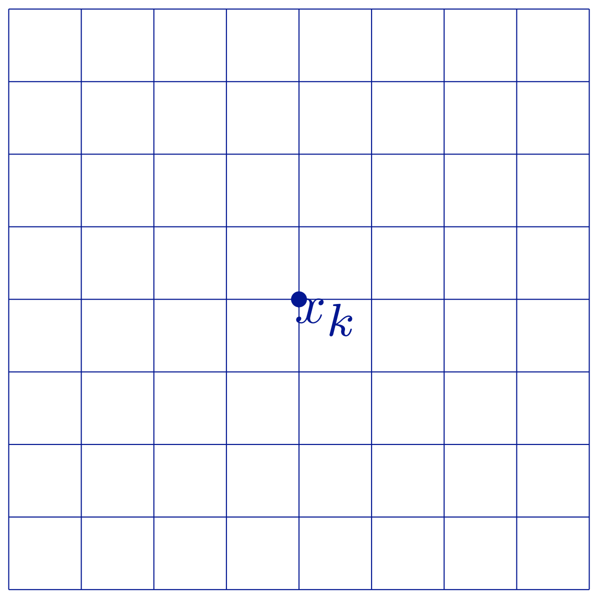


The MADS frames

In addition to the mesh size parameter Δ_k^m, we introduce the poll size parameter Δ_k^p. Ex : $\Delta_k^p = n\sqrt{\Delta_k^m}$.

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

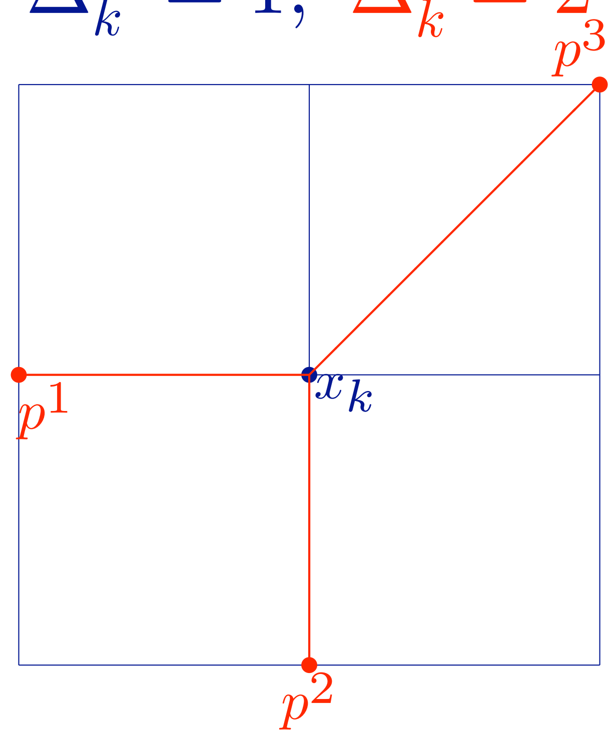




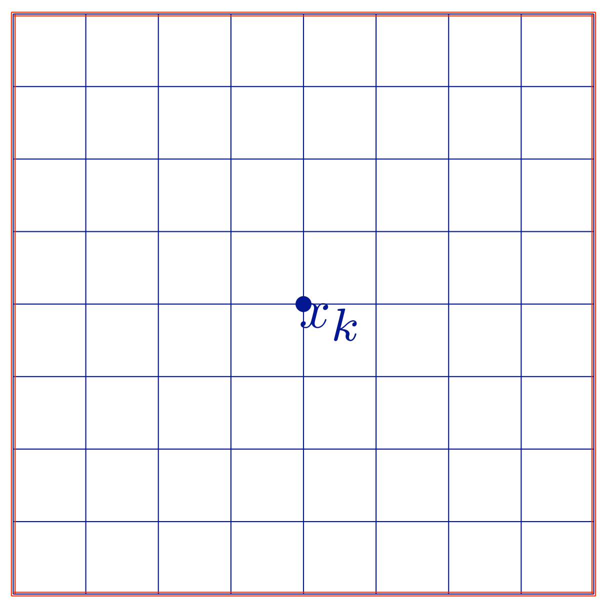
The MADS frames

In addition to the mesh size parameter Δ_k^m, we introduce the poll size parameter Δ_k^p. Ex : $\Delta_k^p = n\sqrt{\Delta_k^m}$.

$\Delta_k^m = 1,\ \Delta_k^p = 2$



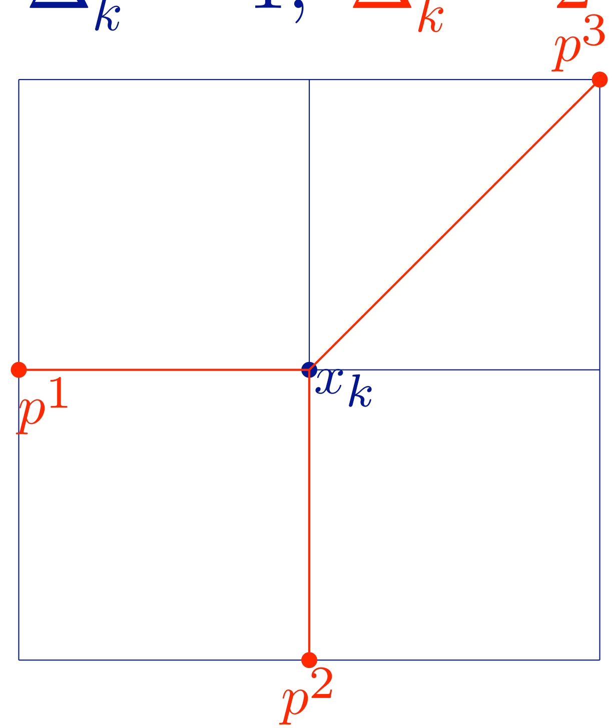
$\Delta_k^m = \frac{1}{4}, \Delta_k^p = 1$



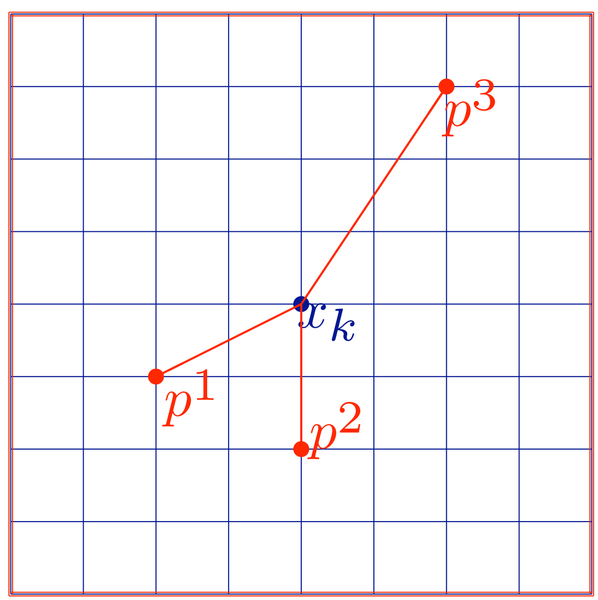
The MADS frames

In addition to the mesh size parameter Δ_k^m, we introduce the poll size parameter Δ_k^p. Ex : $\Delta_k^p = n\sqrt{\Delta_k^m}$.

$\Delta_k^m = 1,\ \Delta_k^p = 2$



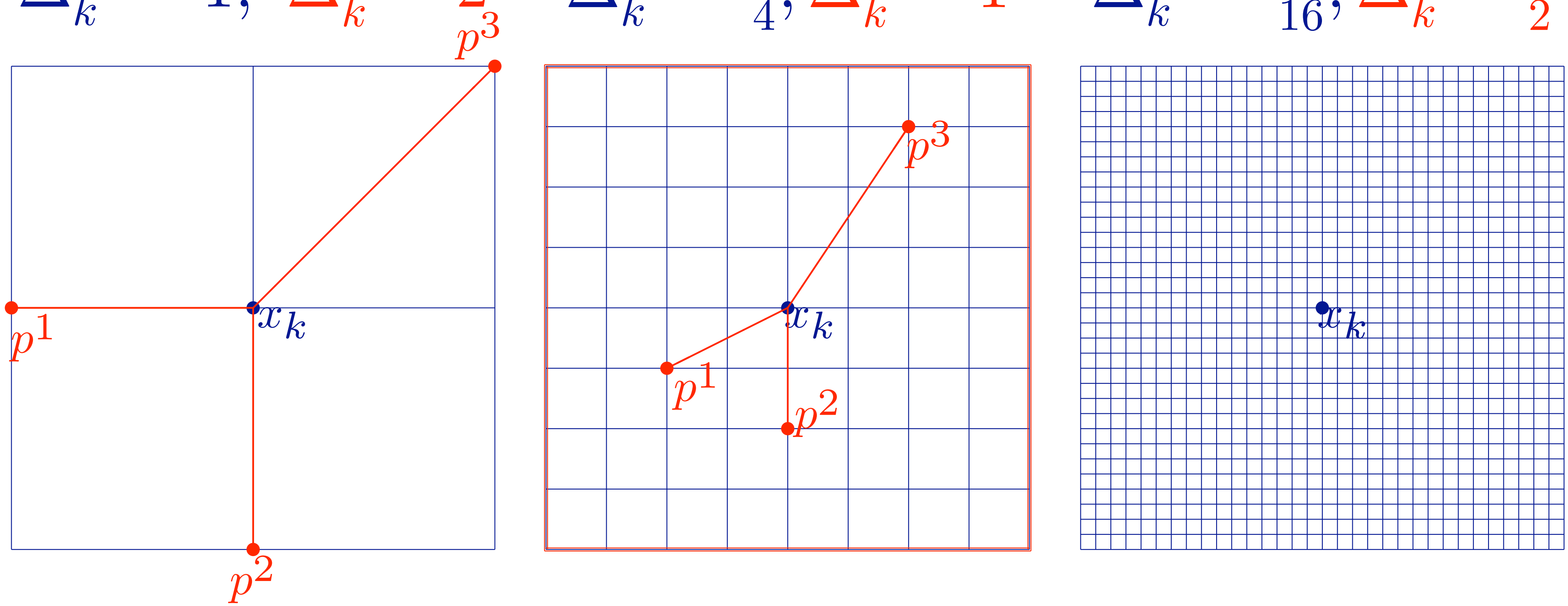
$\Delta_k^m = \frac{1}{4}, \Delta_k^p = 1$



The MADS frames

In addition to the mesh size parameter Δ_k^m, we introduce the poll size parameter Δ_k^p. Ex : $\Delta_k^p = n\sqrt{\Delta_k^m}$.

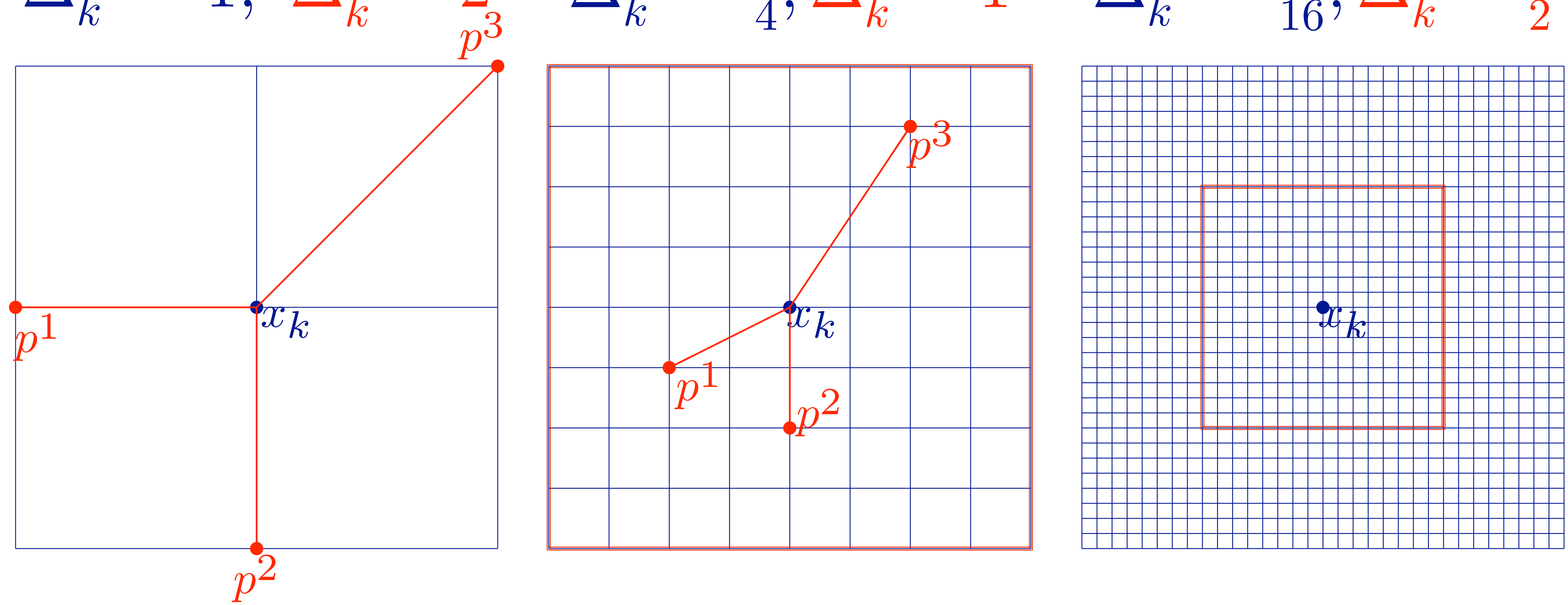
$\Delta_k^m = 1,\ \Delta_k^p = 2$ $\quad \Delta_k^m = \frac{1}{4}, \Delta_k^p = 1$ $\quad \Delta_k^m = \frac{1}{16}, \Delta_k^p = \frac{1}{2}$



The MADS frames

In addition to the mesh size parameter Δ_k^m, we introduce the poll size parameter Δ_k^p. Ex : $\Delta_k^p = n\sqrt{\Delta_k^m}$.

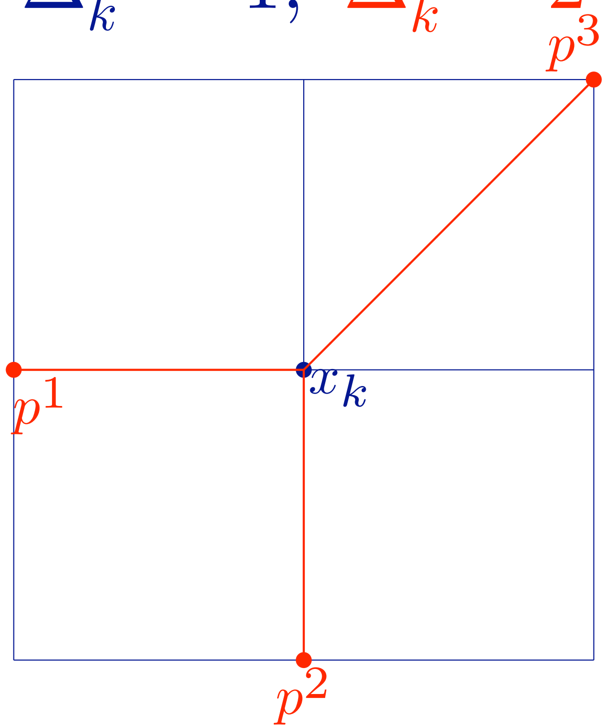
$\Delta_k^m = 1,\ \Delta_k^p = 2$ $\quad$ $\Delta_k^m = \frac{1}{4}, \Delta_k^p = 1$ $\quad$ $\Delta_k^m = \frac{1}{16}, \Delta_k^p = \frac{1}{2}$



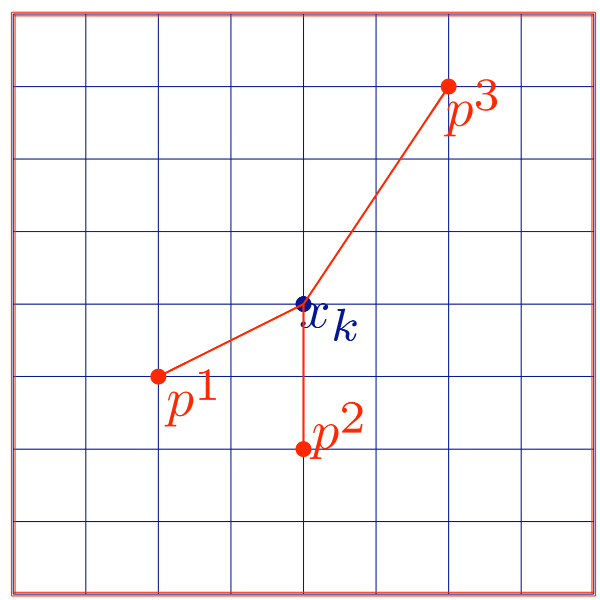
The MADS frames

In addition to the mesh size parameter Δ_k^m, we introduce the poll size parameter Δ_k^p. Ex : $\Delta_k^p = n\sqrt{\Delta_k^m}$.

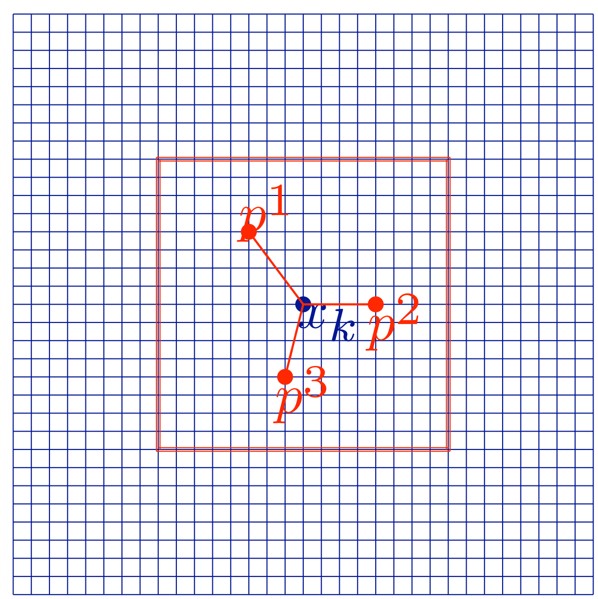
$\Delta_k^m = 1,\ \Delta_k^p = 2$



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



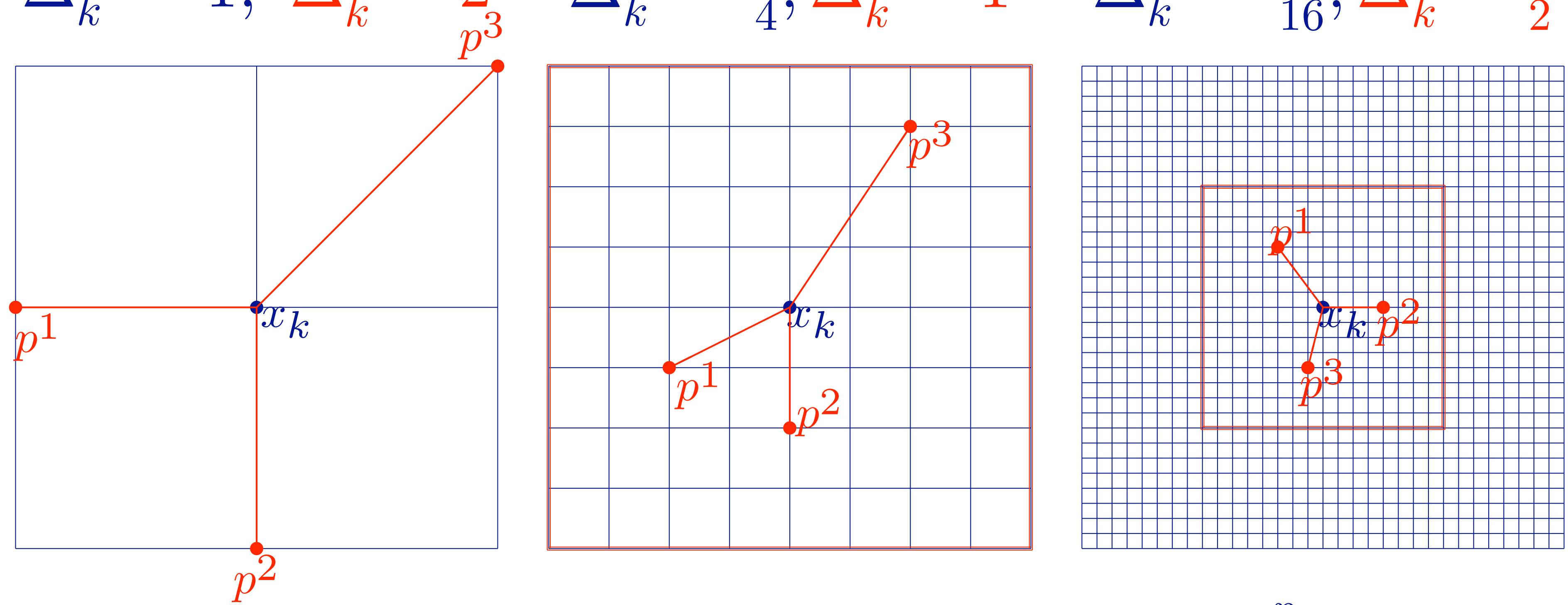
$\Delta_k^m = \frac{1}{16}, \Delta_k^p = \frac{1}{2}$



The MADS frames

In addition to the mesh size parameter Δ_k^m, we introduce the poll size parameter Δ_k^p. Ex : $\Delta_k^p = n\sqrt{\Delta_k^m}$.

$\Delta_k^m = 1, \ \Delta_k^p = 2$ $\quad$ $\Delta_k^m = \frac{1}{4}, \Delta_k^p = 1$ $\quad$ $\Delta_k^m = \frac{1}{16}, \Delta_k^p = \frac{1}{2}$



Number of ways of selecting D_k increases as Δ_k^p gets smaller.

Barrier approach to constraints

To enforce Ω constraints, replace f by a barrier objective

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

This is a standard construct in nonsmooth optimization.

Barrier approach to constraints

To enforce Ω constraints, replace f by a barrier objective

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

This is a standard construct in nonsmooth optimization.
Then apply the unconstrained algorithm to f_Ω.
This is NOT a standard construct in optimization algorithms.

Barrier approach to constraints

To enforce Ω constraints, replace f by a barrier objective

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

This is a standard construct in nonsmooth optimization.
Then apply the unconstrained algorithm to f_Ω.
This is NOT a standard construct in optimization algorithms.
Quality of the limit solution depends the local smoothness of f, not of f_Ω.

A MADS instance

NOTE: GPS = MADS with $\Delta_k^p = \Delta_k^m$.

A MADS instance

NOTE: GPS = MADS with $\Delta_k^p = \Delta_k^m$.

An implementable way to generate D_k:

A MADS instance

NOTE: GPS = MADS with $\Delta_k^p = \Delta_k^m$.

An implementable way to generate D_k:

- Let B be a lower triangular nonsingular random integer matrix.

A MADS instance

NOTE: GPS = MADS with $\Delta_k^p = \Delta_k^m$.

An implementable way to generate D_k:

- Let B be a lower triangular nonsingular random integer matrix.
- Randomly permute the lines of B

A MADS instance

NOTE: GPS = MADS with $\Delta_k^p = \Delta_k^m$.

An implementable way to generate D_k:

- Let B be a lower triangular nonsingular random integer matrix.
- Randomly permute the lines of B
- Complete to a positive basis
  - $D_k = [B; \ -B]$ (maximal positive basis $2n$ directions). or
  - $D_k = [B; \ -\sum_{b \in B} b]$ (minimal positive basis $n+1$ directions).
  - Use Luis' talk to order the poll directions

Dense polling directions

Theorem 1. *As* $k \to \infty$*, MADS's polling directions form a dense set in* $\mathbb{R}^n$ *(with probability 1).*

Dense polling directions

Theorem 1. *As $k \to \infty$, MADS's polling directions form a dense set in $\mathbb{R}^n$ (with probability 1).*

The ultimate goal is a way to be sure that the subset of refining directions $\hat{D}$ is dense.

Dense polling directions

Theorem 1. *As $k \to \infty$, MADS's polling directions form a dense set in $\mathbb{R}^n$ (with probability 1).*

The ultimate goal is a way to be sure that the subset of refining directions $\hat{D}$ is dense.

Then the barrier approach to constraints promises strong optimality under weak assumptions - the existence of a hypertangent vector, e.g., a vector that makes a negative inner product with all the active constraint gradients.

MADS convergence results

Let f be Lipschitz near a limit $\hat{x}$ of a refining sequence.

Theorem 2. *Suppose that $\hat{D}$ is dense in Ω.*

- *If either $\Omega = \mathbb{R}^n$, or $\hat{x} \in int(\Omega)$, then* $0 \in \partial f(\hat{x})$.

MADS convergence results

Let f be Lipschitz near a limit $\hat{x}$ of a refining sequence.

Theorem 2. *Suppose that $\hat{D}$ is dense in Ω.*

- *If either $\Omega = \mathbb{R}^n$, or $\hat{x} \in int(\Omega)$, then* $0 \in \partial f(\hat{x})$.

Theorem 3. *Suppose that $\hat{D}$ is dense in $T_\Omega^H(\hat{x}) \neq \emptyset$.*

- *Then $\hat{x}$ is a Clarke stationary point of f over Ω:*

MADS convergence results

Let f be Lipschitz near a limit $\hat{x}$ of a refining sequence.

Theorem 2. *Suppose that $\hat{D}$ is dense in Ω.*

- *If either $\Omega = \mathbb{R}^n$, or $\hat{x} \in int(\Omega)$, then* $0 \in \partial f(\hat{x})$.

Theorem 3. *Suppose that $\hat{D}$ is dense in $T_\Omega^H(\hat{x}) \neq \emptyset$.*

- *Then $\hat{x}$ is a Clarke stationary point of f over Ω:*

$$f^\circ(\hat{x}; v) \geq 0, \forall v \in T_\Omega^{Cl}(\hat{x}).$$

MADS convergence results

Let f be Lipschitz near a limit $\hat{x}$ of a refining sequence.

Theorem 2. *Suppose that $\hat{D}$ is dense in Ω.*

• *If either $\Omega = \mathbb{R}^n$, or $\hat{x} \in int(\Omega)$, then* $0 \in \partial f(\hat{x})$.

Theorem 3. *Suppose that $\hat{D}$ is dense in $T_\Omega^H(\hat{x}) \neq \emptyset$.*

• *Then $\hat{x}$ is a Clarke stationary point of f over Ω:*

$$f^\circ(\hat{x}; v) \geq 0, \forall v \in T_\Omega^{Cl}(\hat{x}).$$

• *In addition, it f is strictly differentiable at $\hat{x}$ and if Ω is regular at $\hat{x}$, then $\hat{x}$ is a contingent KKT stationary point of f over Ω*

MADS convergence results

Let f be Lipschitz near a limit $\hat{x}$ of a refining sequence.

Theorem 2. *Suppose that $\hat{D}$ is dense in Ω.*

• *If either $\Omega = \mathbb{R}^n$, or $\hat{x} \in int(\Omega)$, then* $0 \in \partial f(\hat{x})$.

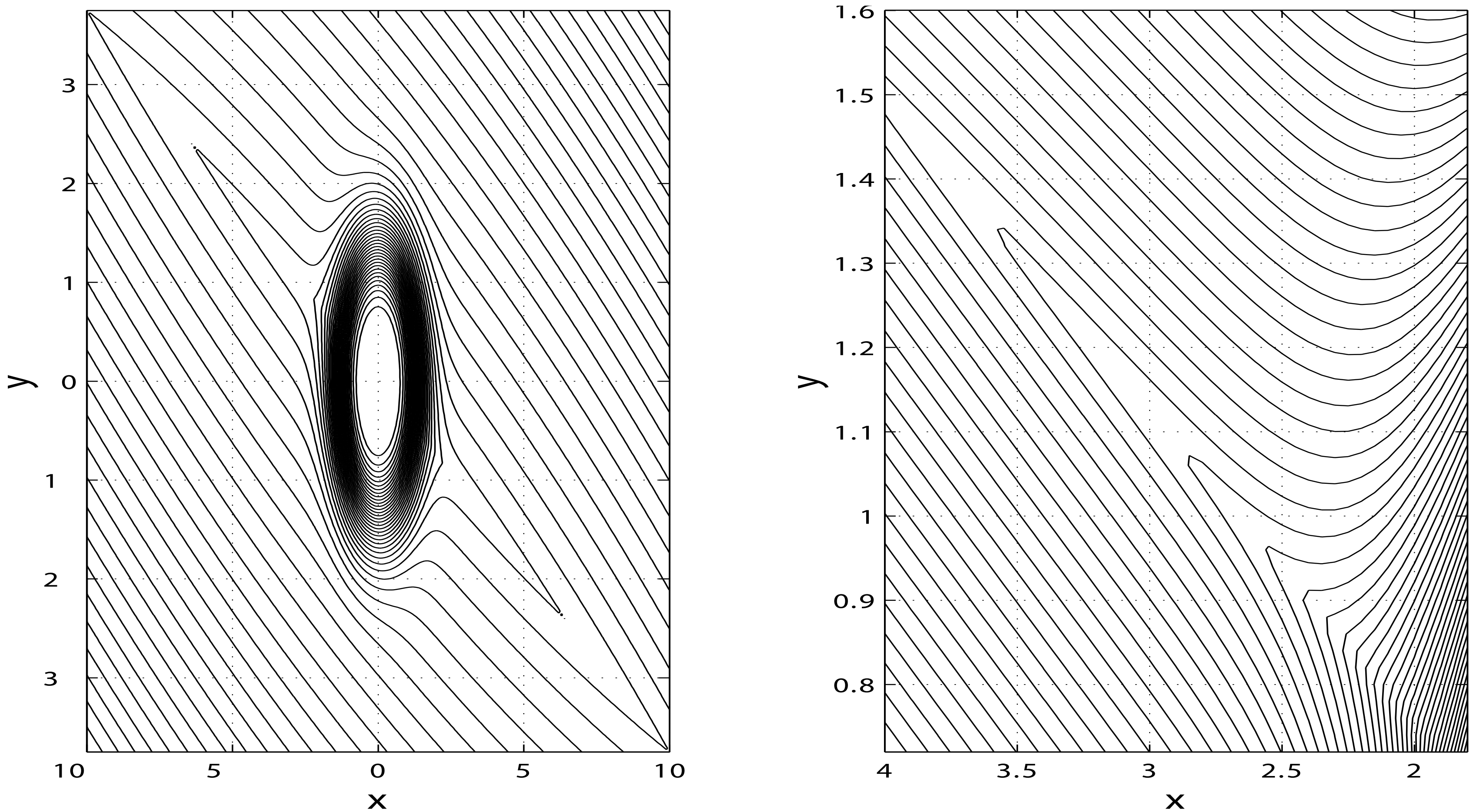
Theorem 3. *Suppose that $\hat{D}$ is dense in $T_\Omega^H(\hat{x}) \neq \emptyset$.*

• *Then $\hat{x}$ is a Clarke stationary point of f over Ω:*

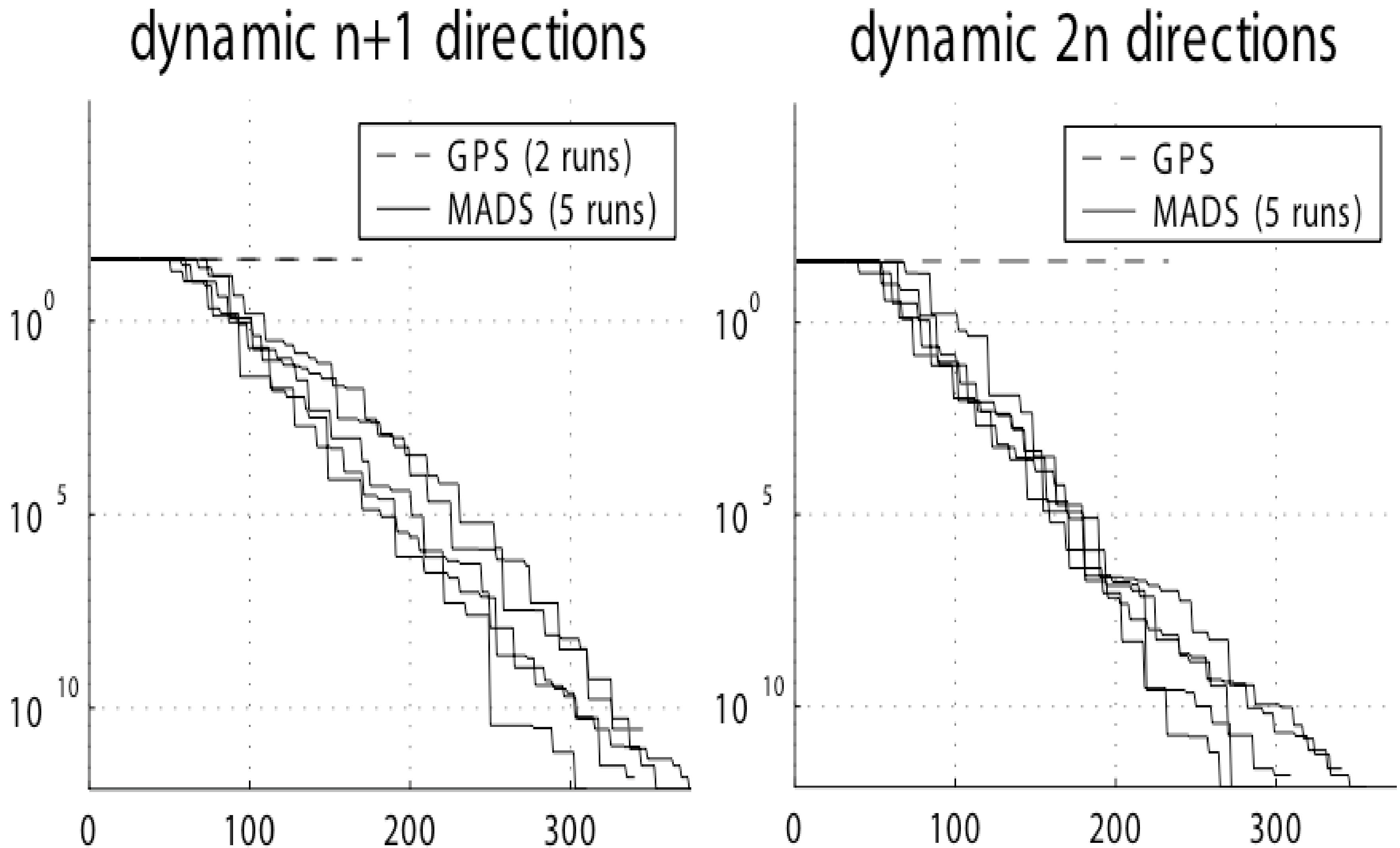
$$f^\circ(\hat{x}; v) \geq 0, \forall v \in T_\Omega^{Cl}(\hat{x}).$$

• *In addition, it f is strictly differentiable at $\hat{x}$ and if Ω is regular at $\hat{x}$, then $\hat{x}$ is a contingent KKT stationary point of f over Ω :* $-\nabla f(\hat{x})^T v \leq 0, \forall v \in T_\Omega^{Co}(\hat{x})$.

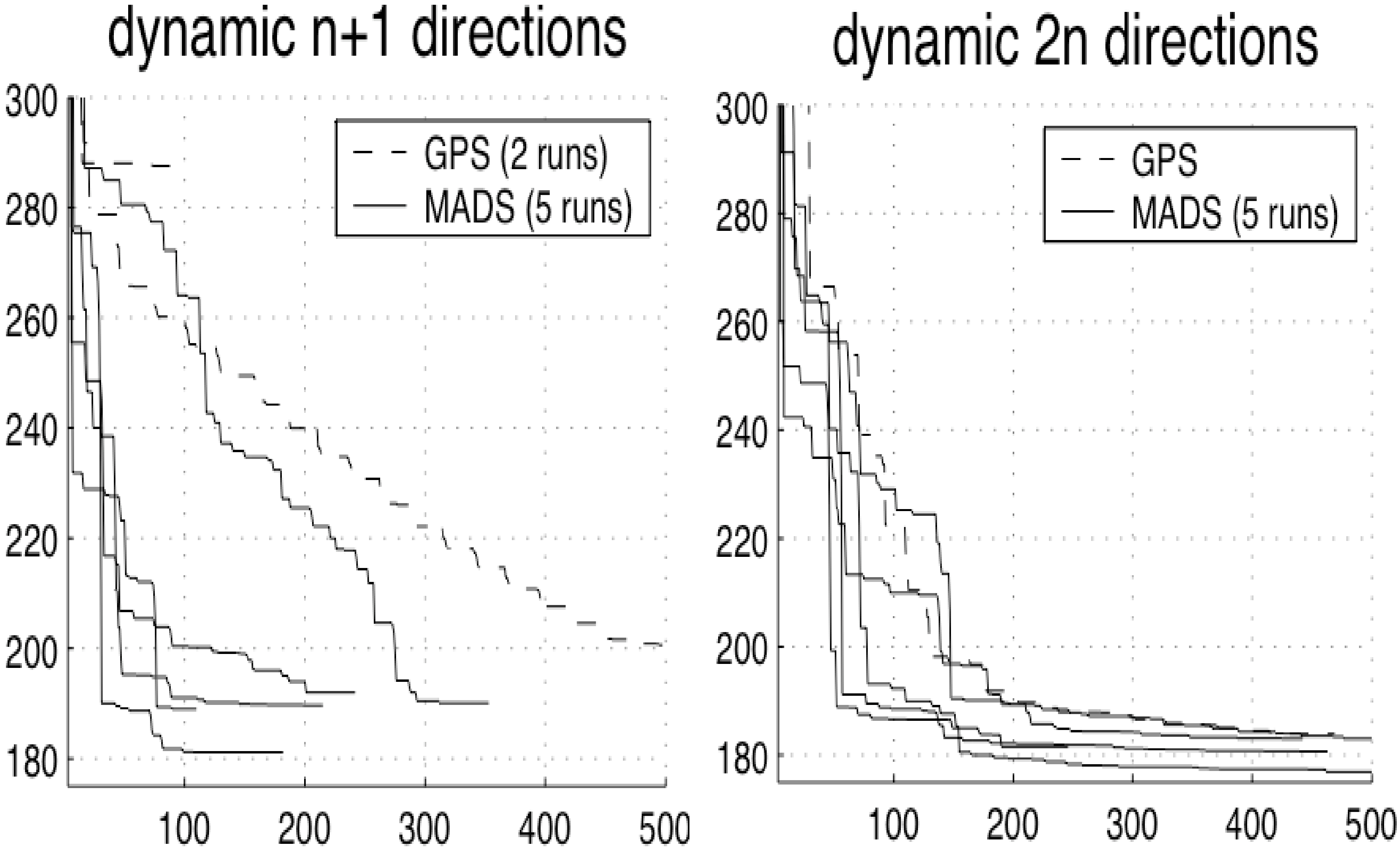
A problem for which GPS stagnates



Our results


dynamic n+1 directions
GPS (2 runs)
MADS (5 runs)
10^0
10^5
10^10
0
100
200
300
dynamic 2n directions
GPS
MADS (5 runs)
10^0
10^5
10^10
0
100
200
300

Results for a chemE pid problem



Constrained optimization

A disk constrained problem

$$\begin{aligned} \min_{x,y} \quad & x + y \\ \text{s.t.} \quad & x^2 + y^2 \leq 6 \end{aligned}$$

How hard can that be?

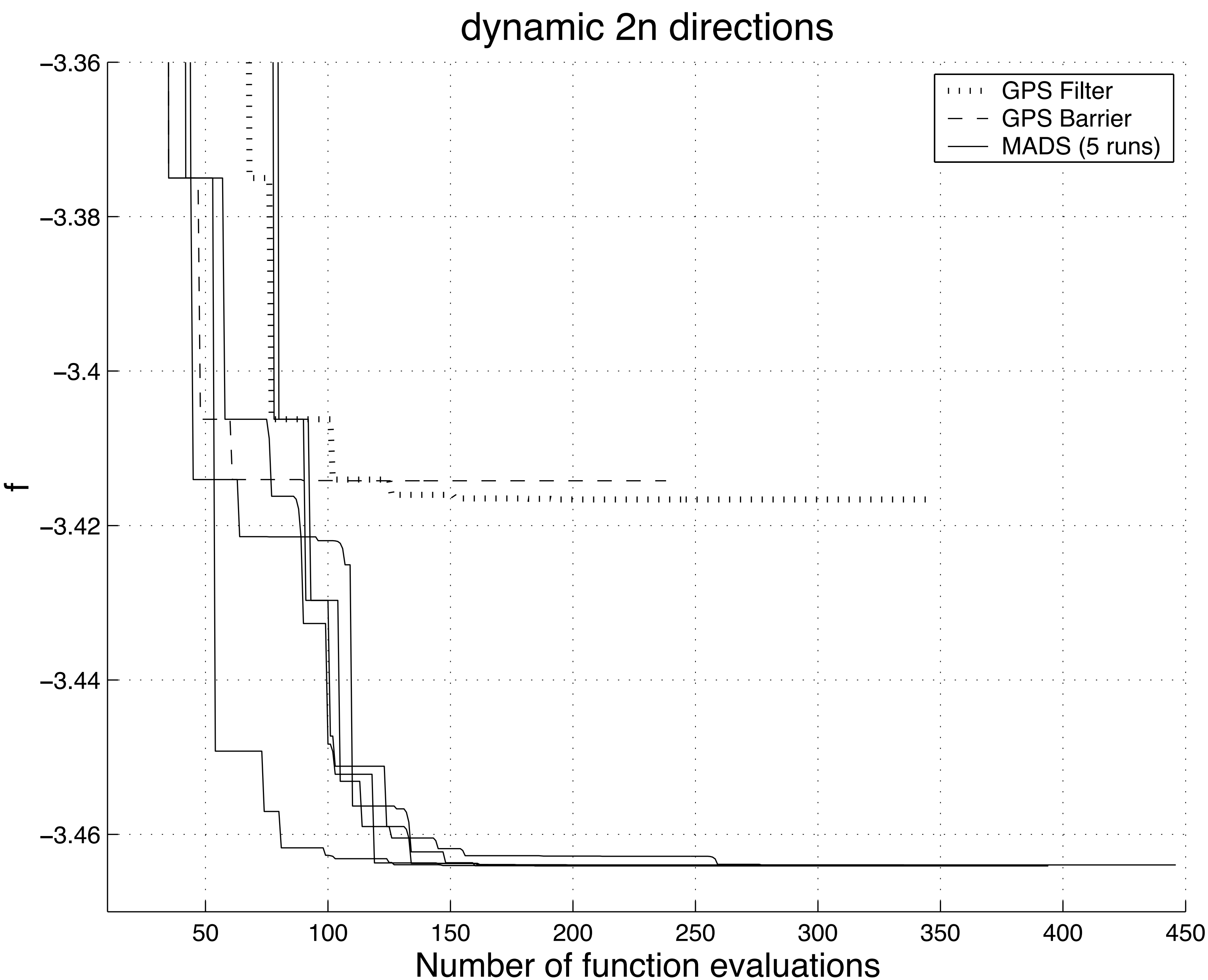
Constrained optimization

A disk constrained problem

$$\min_{x,y} \quad x + y$$
$$\text{s.t.} \quad x^2 + y^2 \leq 6$$

How hard can that be?

Very hard for GPS and filter-GPS with the standard 2n directions with an empty SEARCH


dynamic 2n directions
−3.36
−3.38
−3.4
−3.42
−3.44
−3.46
f
50
100
150
200
250
300
350
400
450
Number of function evaluations
GPS Filter
GPS Barrier
MADS (5 runs)

Parameter fit in a rheology problem

Rheology is a branch of mechanics that studies properties of materials which determine their response to mechanical force.

MODEL :

Viscosity η of a material can be modelled as a function of the shear rate $\dot{\gamma}_i$:

$$\eta(\dot{\gamma}) \;=\; \eta_0(1 + \lambda^2\dot{\gamma}^2)^{\frac{\beta-1}{2}}$$

A parameter fit problem.

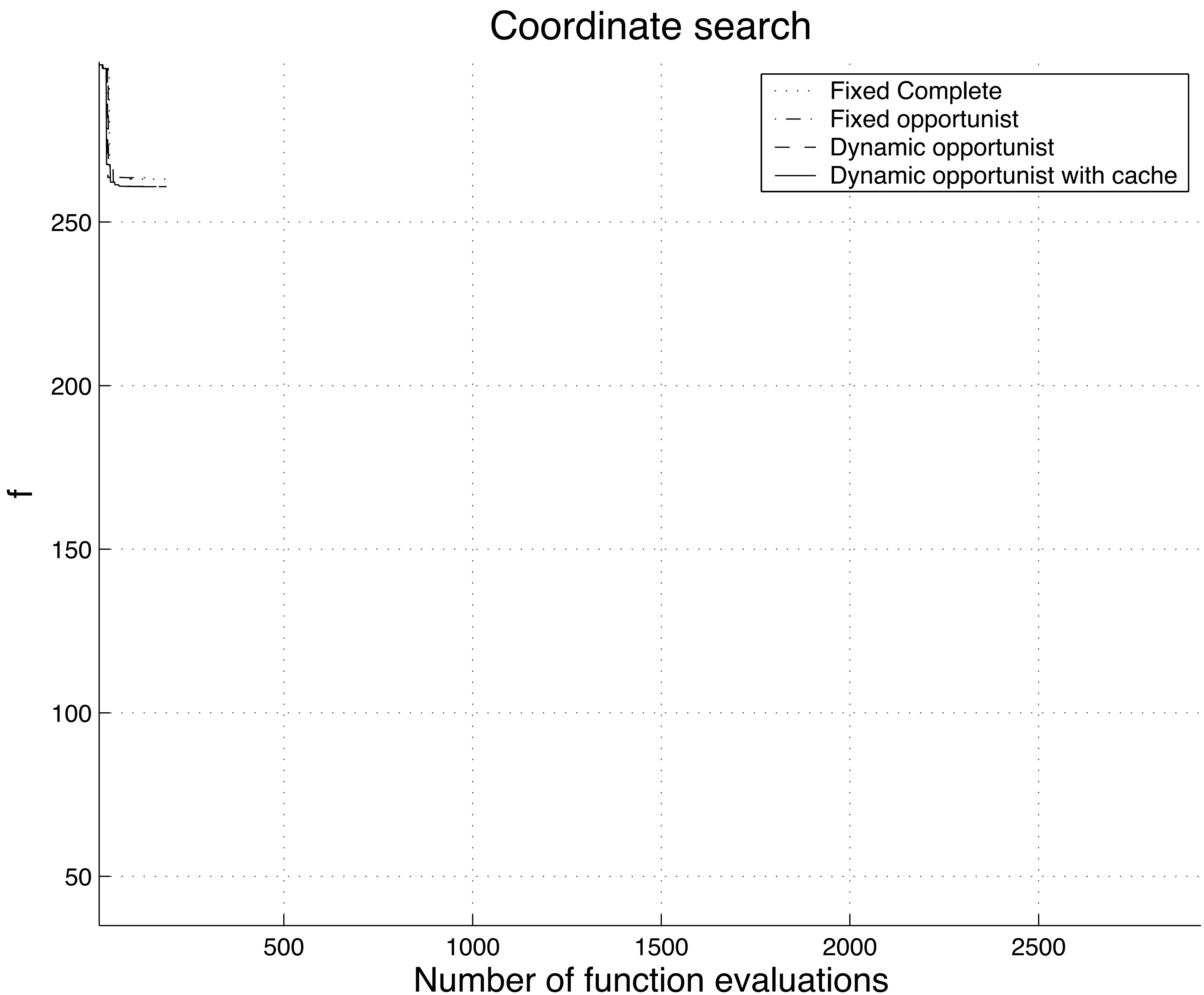
Observation i	Strain rate $\dot{\gamma}_i$ (s^{-1})	Viscosity η_i $(Pa \cdot s)$
1	0.0137	3220
2	0.0274	2190
3	0.0434	1640
4	0.0866	1050
5	0.137	766
6	0.274	490
7	0.434	348
8	0.866	223
9	1.37	163
10	2.74	104
11	4.34	76.7
12	5.46	68.1
13	6.88	58.2

The unconstrained optimization problem :

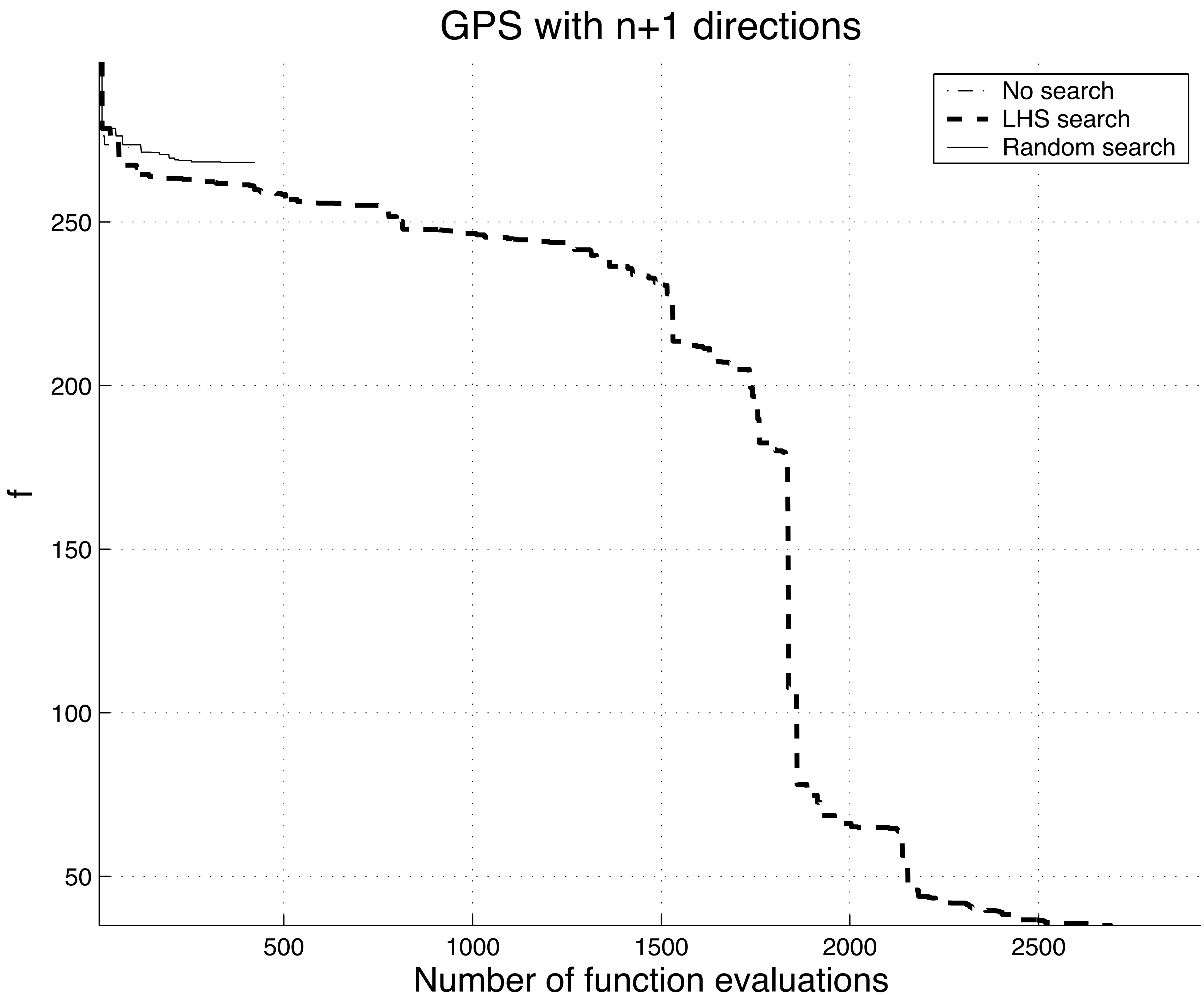
$$\min_{\eta_0,\lambda,\beta} \; g(\eta_0, \lambda, \beta)$$

with

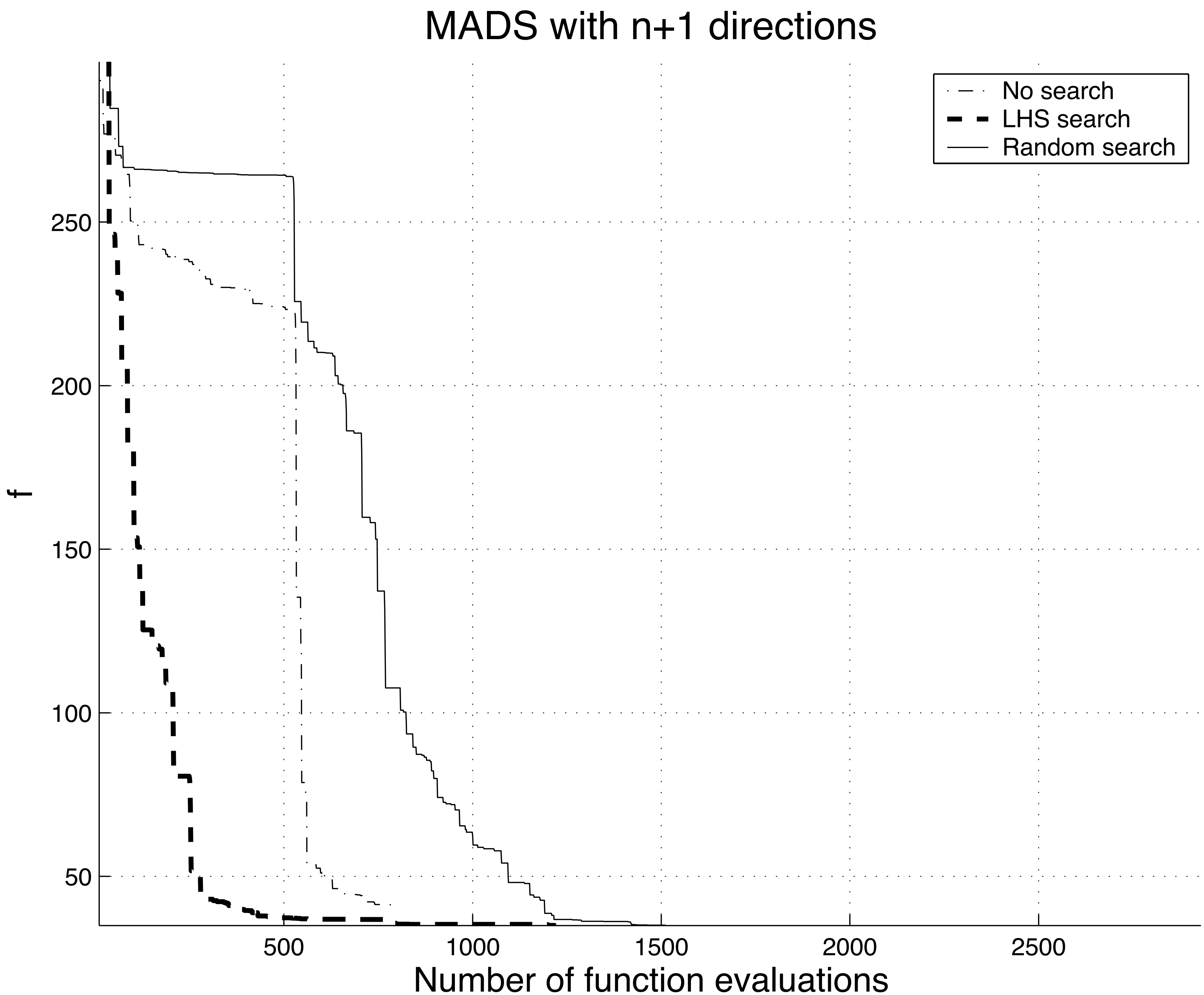
$$g = \sum_{i=1}^{13} |\eta(\dot{\gamma}) - \eta_i|$$


Coordinate search
Fixed Complete
Fixed opportunist
Dynamic opportunist
Dynamic opportunist with cache
250
200
150
100
50
f
500
1000
1500
2000
2500
Number of function evaluations

Rice 2004


GPS with n+1 directions
No search
LHS search
Random search
f
250
200
150
100
50
500
1000
1500
2000
2500
Number of function evaluations

Rice 2004


MADS with n+1 directions
No search
LHS search
Random search
f
250
200
150
100
50
500
1000
1500
2000
2500
Number of function evaluations

Rice 2004

Discussion

- MADS variant looks good as a 1st try

Discussion

- MADS variant looks good as a 1st try,
  and is more general than the instance shown here.

Discussion

- MADS variant looks good as a 1st try,
  and is more general than the instance shown here.

- Numerically, randomness is a blessing and a curse.

Discussion

- MADS variant looks good as a 1st try,
  and is more general than the instance shown here.

- Numerically, randomness is a blessing and a curse.

- MADS can handle oracular or yes/no constraints.

Discussion

- MADS variant looks good as a 1st try,
  and is more general than the instance shown here.
- Numerically, randomness is a blessing and a curse.
- MADS can handle oracular or yes/no constraints.
- The underlying mesh is finer in MADS than in GPS :
  Good for general searches and surrogates.

Discussion

- MADS variant looks good as a 1st try, and is more general than the instance shown here.
- Numerically, randomness is a blessing and a curse.
- MADS can handle oracular or yes/no constraints.
- The underlying mesh is finer in MADS than in GPS : Good for general searches and surrogates.
- MADS is the result of nonsmooth analysis pointing up the weaknesses in GPS.

Later today...

- This is a meeting about surrogates,
  but I did not talk about surrogates...
  Alison will present in the next talk the use of a surrogate in a specific mechanical engineering problem using GPS/MADS.

Later today...

- This is a meeting about surrogates,
  but I did not talk about surrogates...
  Alison will present in the next talk the use of a surrogate in a specific mechanical engineering problem using GPS/MADS.

- MADS replaces GPS in our NOMADm and NOMAD softwares. Gilles and Mark will present a demo of these sofwares after lunch.