Randomized Direct Search for Derivative-free Optimization

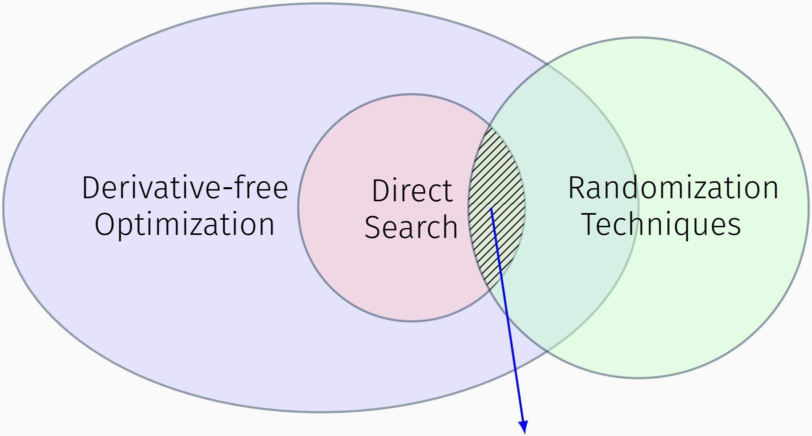
AMA613

Cunxin Huang
Co-supervised by Dr. Zaikun Zhang and Prof. Xiaojun Chen

September 18, 2023

Department of Applied Mathematics
The Hong Kong Polytechnic University

Big Picture



Our final goal in this talk
Randomized Direct Search

What is DFO and Why?

Derivative-free optimization (DFO)

- A branch of optimization
- Do not use derivatives (only use function evaluations)

What is DFO and Why?

Derivative-free optimization (DFO)

- A branch of optimization
- Do not use derivatives (only use function evaluations)

Cruel reality

- Derivatives are not available
- Problems are always noisy (finite difference loses effectiveness)
- Function evaluations are expensive (e.g.: PDE simulation)

What is DFO and Why?

Derivative-free optimization (DFO)

- A branch of optimization
- Do not use derivatives (only use function evaluations)

Cruel reality

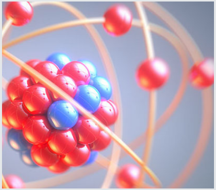
- Derivatives are not available
- Problems are always noisy (finite difference loses effectiveness)
- Function evaluations are expensive (e.g.: PDE simulation)

Example

$$x \in \mathbb{R}^n \longrightarrow \boxed{f : \mathbb{R}^n \to \mathbb{R}} \longrightarrow f(x)$$

Wide Application and Popularity

Applications



Nuclear Physics

Machine Learning

Cosmology

Wide Application and Popularity

- Powell's conjugate direction method (1964)

An efficient method for finding the minimum of a function of several variables without calculating derivatives
MJD Powell
The computer journal, 1964 • academic.oup.com
☆ Save Cite Cited by 5969 Related articles All 4 versions

- Nelder-Mead simplex method (1965)

A simplex method for function minimization
JA Nelder, R Mead - The computer journal, 1965 - academic.oup.com
☆ Save Cite Cited by 37851 Related articles All 16 versions Web of Science: 20486

Basic Assumptions

In this talk, we solve the unconstrained problem

$$\min_{x \in \mathbb{R}^n} f(x),$$

where

- ∇f is Lipschitz continuous with constant ν, cannot be evaluated,
- f is bounded below,
- the evaluation of f is expensive.

Basic Assumptions

In this talk, we solve the unconstrained problem

$$\min_{x \in \mathbb{R}^n} f(x),$$

where

- ∇f is Lipschitz continuous with constant ν, cannot be evaluated,
- f is bounded below,
- the evaluation of f is expensive.
  Number of function evaluations matters!

Basic Assumptions

In this talk, we solve the unconstrained problem

$$\min_{x \in \mathbb{R}^n} f(x),$$

where

- ∇f is Lipschitz continuous with constant ν, cannot be evaluated,
- f is bounded below,
- the evaluation of f is expensive.
  Number of function evaluations matters!

The fewer, the better.

Direct-search Methods

Direct-search methods

- Sample points based on a finite direction set
- Make decisions by simple comparison (no explicit models)

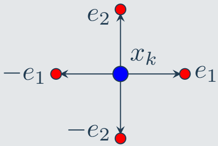
Direct-search Methods

Direct-search methods

- Sample points based on a finite direction set
- Make decisions by simple comparison (no explicit models)

Example

$$D = \{e_1, -e_1, e_2, -e_2\}$$



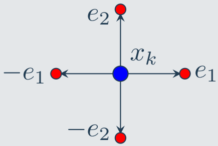
Direct-search Methods

Direct-search methods

- Sample points based on a finite direction set
- Make decisions by simple comparison (no explicit models)

Example

$$D = \{e_1, -e_1, e_2, -e_2\}$$



In this talk, direction sets only contain unit vectors in $\mathbb{R}^n$.

Algorithm 1: Direct Search with sufficient decrease

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, \infty)$, $0 < \theta < 1 \leq \gamma$.

for $k = 0, 1, \ldots$ **do**

Algorithm 1: Direct Search with sufficient decrease

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, \infty)$, $0 < \theta < 1 \leq \gamma$.

for $k = 0, 1, \ldots$ **do**

  Select $D_k \subset \mathbb{R}^n$.

Framework of Direct Search

Algorithm 1: Direct Search with sufficient decrease

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, \infty)$, $0 < \theta < 1 \leq \gamma$.

for $k = 0, 1, \ldots$ **do**

  Select $D_k \subset \mathbb{R}^n$.

  if $f(x_k + \alpha_k d) < f(x_k) - \alpha_k^2$ for some $d \in D_k$ **then**

Framework of Direct Search

Algorithm 1: Direct Search with sufficient decrease

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, \infty)$, $0 < \theta < 1 \leq \gamma$.

for $k = 0, 1, \ldots$ **do**

  Select $D_k \subset \mathbb{R}^n$.

  if $f(x_k + \alpha_k d) < f(x_k) - \alpha_k^2$ for some $d \in D_k$ **then**

    Expand step size, and move to that point

Framework of Direct Search

Algorithm 1: Direct Search with sufficient decrease

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, \infty)$, $0 < \theta < 1 \leq \gamma$.

for $k = 0, 1, \ldots$ **do**

  Select $D_k \subset \mathbb{R}^n$.

  if $f(x_k + \alpha_k d) < f(x_k) - \alpha_k^2$ for some $d \in D_k$ **then**

    Expand step size, and move to that point

    Set $\alpha_{k+1} = \gamma\alpha_k$ and $x_{k+1} = x_k + \alpha_k d$.

  else

Framework of Direct Search

Algorithm 1: Direct Search with sufficient decrease

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, \infty)$, $0 < \theta < 1 \leq \gamma$.

for $k = 0, 1, \dots$ **do**

  Select $D_k \subset \mathbb{R}^n$.

  if $f(x_k + \alpha_k d) < f(x_k) - \alpha_k^2$ for some $d \in D_k$ **then**

    Expand step size, and move to that point

    Set $\alpha_{k+1} = \gamma\alpha_k$ and $x_{k+1} = x_k + \alpha_k d$.

  else

    Shrink step size, and stand still

Algorithm 1: Direct Search with sufficient decrease

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, \infty)$, $0 < \theta < 1 \leq \gamma$.

for $k = 0, 1, \ldots$ **do**

  Select $D_k \subset \mathbb{R}^n$.

  if $f(x_k + \alpha_k d) < f(x_k) - \alpha_k^2$ for some $d \in D_k$ **then**

    Expand step size, and move to that point

    Set $\alpha_{k+1} = \gamma \alpha_k$ and $x_{k+1} = x_k + \alpha_k d$.

  else

    Shrink step size, and stand still

    Set $\alpha_{k+1} = \theta \alpha_k$ and $x_{k+1} = x_k$.

Framework of Direct Search

Algorithm 1: Direct Search with sufficient decrease

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, \infty)$, $0 < \theta < 1 \leq \gamma$.

for $k = 0, 1, \ldots$ **do**

  Select $D_k \subset \mathbb{R}^n$.

  if $f(x_k + \alpha_k d) < f(x_k) - \alpha_k^2$ for some $d \in D_k$ **then**

    Expand step size, and move to that point

    Set $\alpha_{k+1} = \gamma\alpha_k$ and $x_{k+1} = x_k + \alpha_k d$.

  else

    Shrink step size, and stand still

    Set $\alpha_{k+1} = \theta\alpha_k$ and $x_{k+1} = x_k$.

Everything almost clear, except "select D_k".

How to Select a “Good” Direction Set?

Question 1: what is a “good” D_k?

How to Select a "Good" Direction Set?

Question 1: what is a "good" D_k?

Include at least one descent direction, i.e., $\exists d \in D_k$ s.t., $-g_k^\top d > 0$, where we use g_k to denote $\nabla f(x_k)$.

How to Select a "Good" Direction Set?

Question 1: what is a "good" D_k?

Include at least one descent direction, i.e., $\exists d \in D_k$ s.t., $-g_k^\top d > 0$, where we use g_k to denote $\nabla f(x_k)$.

Question 2: how to choose a "good" D_k when we don't know g_k?

How to Select a "Good" Direction Set?

Question 1: what is a "good" D_k?

Include at least one descent direction, i.e., $\exists d \in D_k$ s.t., $-g_k^\top d > 0$, where we use g_k to denote $\nabla f(x_k)$.

Question 2: how to choose a "good" D_k when we don't know g_k?

Natural idea: choose a D_k s.t.,

$$\forall v \in \mathbb{R}^n,\ \exists d \in D_k \text{ satisfying } d^\top v > 0.$$

How to Select a "Good" Direction Set?

Question 1: what is a "good" D_k?

Include at least one descent direction, i.e., $\exists d \in D_k$ s.t., $-g_k^\top d > 0$, where we use g_k to denote $\nabla f(x_k)$.

Question 2: how to choose a "good" D_k when we don't know g_k?

Natural idea: choose a D_k s.t.,

$$\forall v \in \mathbb{R}^n, \exists d \in D_k \text{ satisfying } d^\top v > 0.$$

Positive Spanning Set (PSS)

How to Select a “Good” Direction Set?

Question 1: what is a “good” D_k?

Include at least one descent direction, i.e., $\exists d \in D_k$ s.t., $-g_k^\top d > 0$, where we use g_k to denote $\nabla f(x_k)$.

Question 2: how to choose a “good” D_k when we don't know g_k?

Natural idea: choose a D_k s.t.,

$$\forall v \in \mathbb{R}^n,\ \exists d \in D_k \text{ satisfying } d^\top v > 0.$$

Positive Spanning Set (PSS)

Example

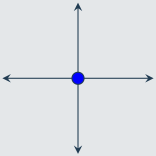


Illustration of How Direct Search Works

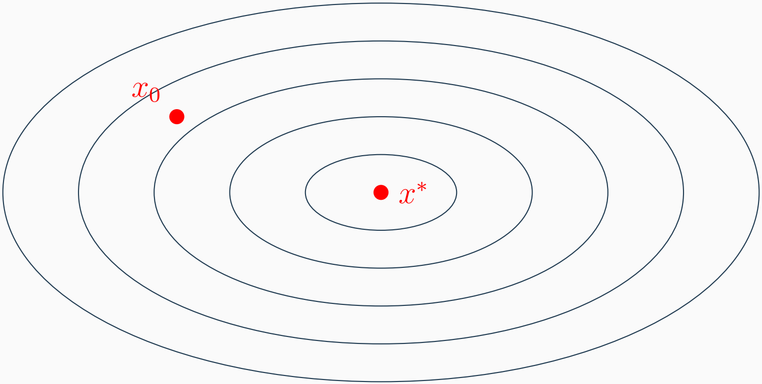


Illustration of How Direct Search Works

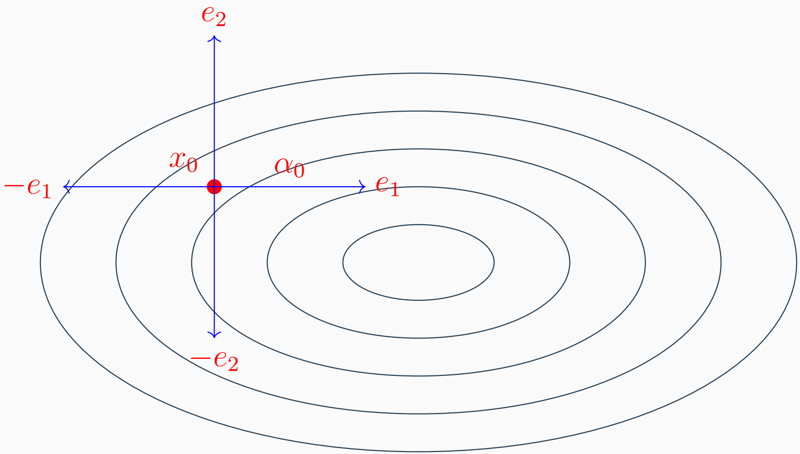


Illustration of How Direct Search Works

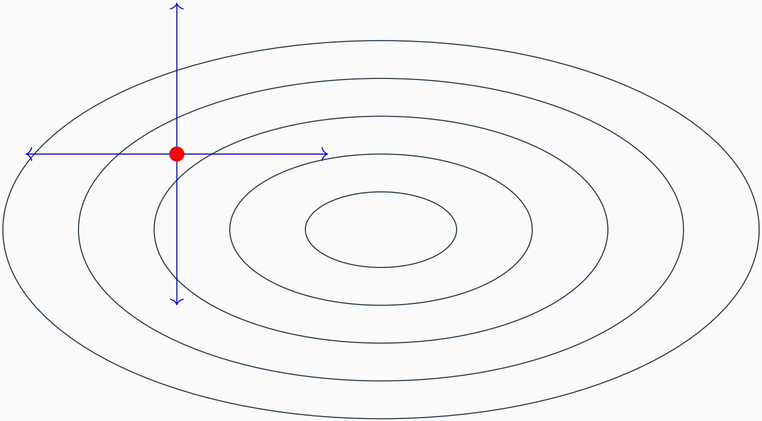


Illustration of How Direct Search Works

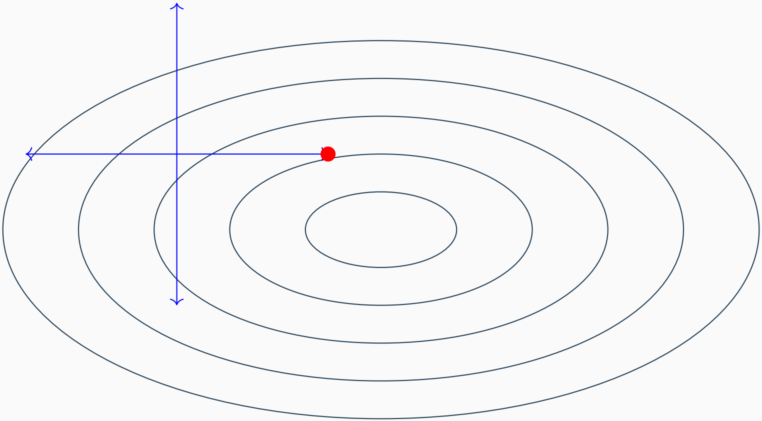


Illustration of How Direct Search Works

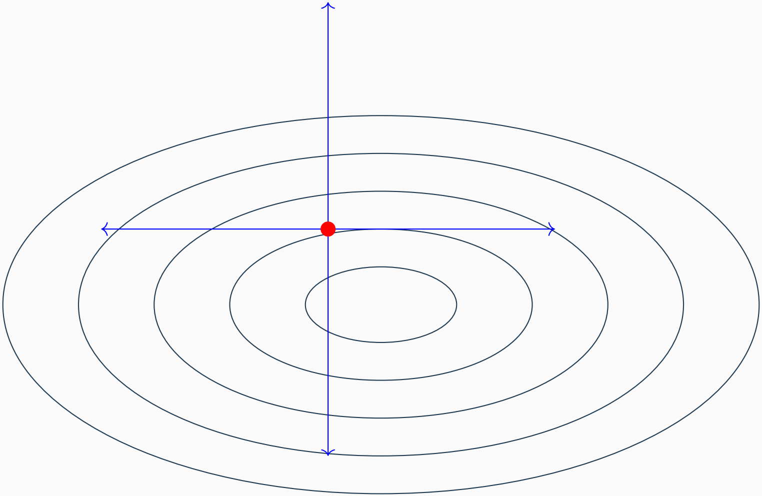


Illustration of How Direct Search Works

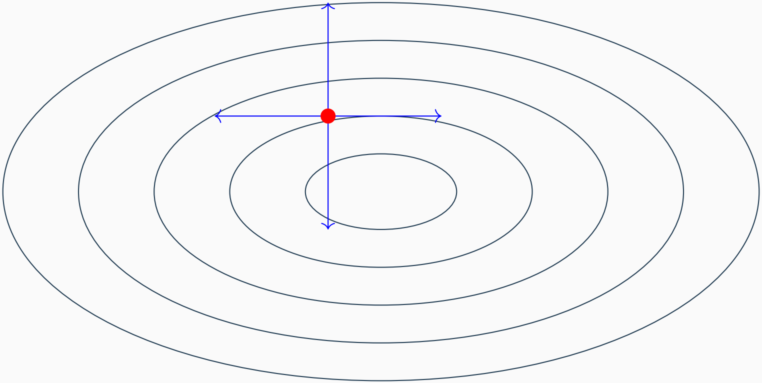


Illustration of How Direct Search Works

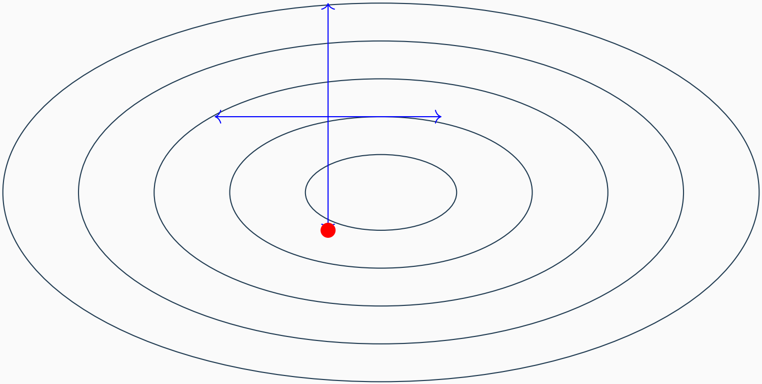


Illustration of How Direct Search Works

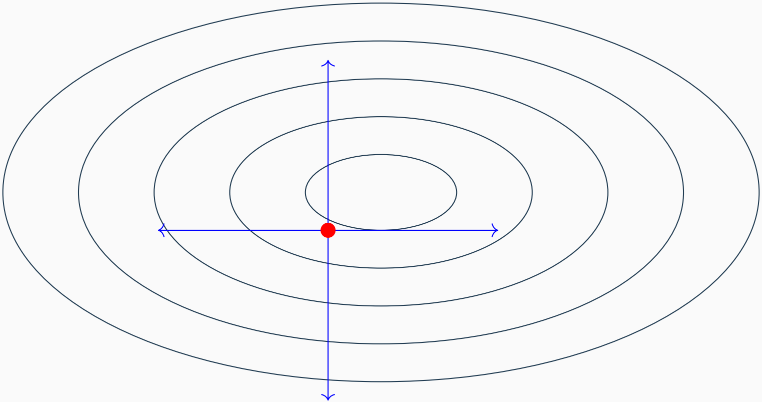


Illustration of How Direct Search Works

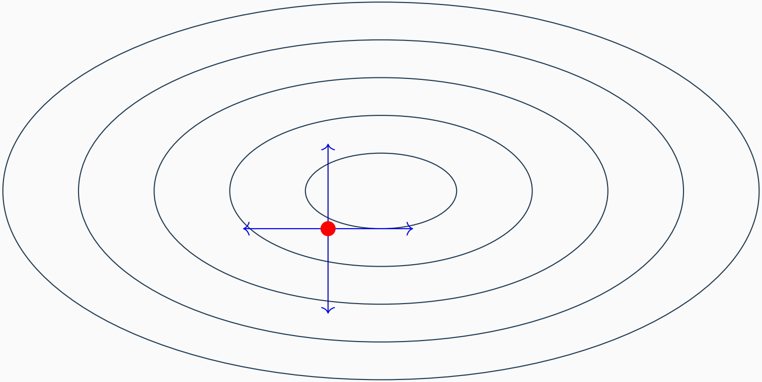


Illustration of How Direct Search Works

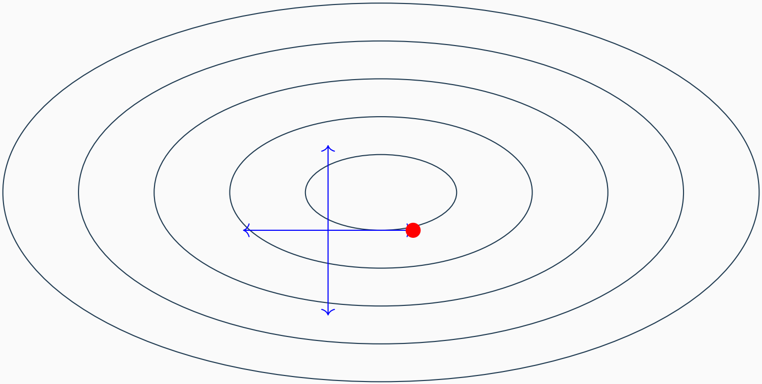
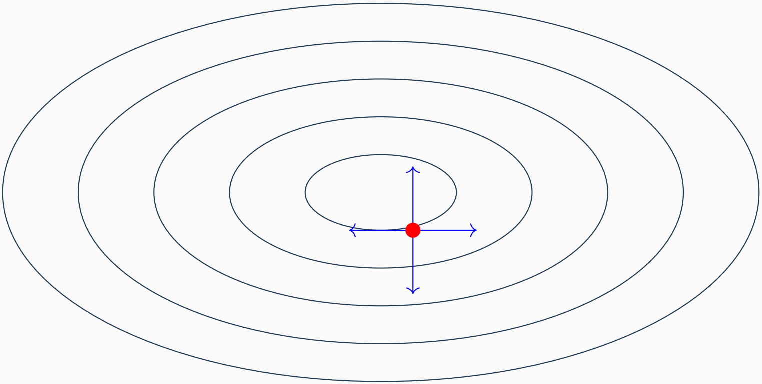


Illustration of How Direct Search Works



How “Good” is a PSS?

How "Good" is a PSS?

Cosine measure

Cosine measure for a finite set of nonzero vectors $D \subseteq \mathbb{R}^n$:

$$\mathrm{cm}(D) \;=\; \min_{v \in \mathbb{R}^n \setminus \{0\}} \mathrm{cm}(D, v) \;=\; \min_{v \in \mathbb{R}^n \setminus \{0\}} \max_{d \in D} \frac{d^\top v}{\|d\| \|v\|}.$$

How “Good” is a PSS?

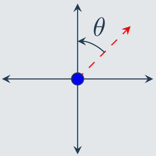
Cosine measure

Cosine measure for a finite set of nonzero vectors $D \subseteq \mathbb{R}^n$:

$$\mathrm{cm}(D) \;=\; \min_{v \in \mathbb{R}^n \setminus \{0\}} \mathrm{cm}(D, v) \;=\; \min_{v \in \mathbb{R}^n \setminus \{0\}} \max_{d \in D} \frac{d^\top v}{\|d\| \|v\|}.$$

Example

$$\mathrm{cm}(D) \;=\; \cos\theta \;=\; \frac{\sqrt{2}}{2}$$



Convergence of Deterministic Direct Search

Theorem

If we have

$$\mathrm{cm}(D_k) \;\geq\; \kappa > 0 \qquad \textit{for all } k \geq 0,$$

then

$$\liminf_{k\to\infty} \|g_k\| \;=\; 0.$$

Convergence of Deterministic Direct Search

Theorem

If we have

$$\mathrm{cm}(D_k) \;\geq\; \kappa > 0 \qquad \textit{for all } k \geq 0,$$

then

$$\liminf_{k\to\infty} \|g_k\| \;=\; 0.$$

Each direction set D_k should be a "good enough" PSS.

Randomization Techniques

Randomization Techniques

Bad news

If D is a PSS in $\mathbb{R}^n$, then $|D| \geq n + 1$.

Bad news

If D is a PSS in $\mathbb{R}^n$, then $|D| \geq n+1$.

Number of function evaluations ensuring $\|g_k\| \leq \varepsilon$:

$$\mathcal{O}(n^2 \varepsilon^{-2})$$

Bad news

If D is a PSS in $\mathbb{R}^n$, then $|D| \geq n+1$.

Number of function evaluations ensuring $\|g_k\| \leq \varepsilon$:

$$\mathcal{O}(n^2\varepsilon^{-2}) \; ? \Rightarrow \mathcal{O}(n\varepsilon^{-2})$$

Randomization Techniques

Bad news

If D is a PSS in $\mathbb{R}^n$, then $|D| \geq n+1$.

Number of function evaluations ensuring $\|g_k\| \leq \varepsilon$:

$$\mathcal{O}(n^2\varepsilon^{-2}) \ ? \Rightarrow \mathcal{O}(n\varepsilon^{-2})$$

Idea

Reduce $|D|$ from $\mathcal{O}(n)$ to $\mathcal{O}(1)$ by randomization techniques

Algorithm 2: Randomized Direct Search

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, \infty)$, $0 < \theta < 1 \leq \gamma$.

for $k = 0, 1, \dots$ **do**

  Randomly generate $\mathfrak{D}_k \subset \mathbb{R}^n$.

  if $f(x_k + \alpha_k d) < f(x_k) - \rho(\alpha_k)$ for some $d \in \mathfrak{D}_k$ **then**

    Set $\alpha_{k+1} = \gamma\alpha_k$ and $x_{k+1} = x_k + \alpha_k d$.

  else

    Set $\alpha_{k+1} = \theta\alpha_k$ and $x_{k+1} = x_k$.

Algorithm 2: Randomized Direct Search

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, \infty)$, $0 < \theta < 1 \leq \gamma$.

for $k = 0, 1, \ldots$ **do**

  Randomly generate $\mathfrak{D}_k \subset \mathbb{R}^n$.

  if $f(x_k + \alpha_k d) < f(x_k) - \rho(\alpha_k)$ for some $d \in \mathfrak{D}_k$ **then**

    Set $\alpha_{k+1} = \gamma\alpha_k$ and $x_{k+1} = x_k + \alpha_k d$.

  else

    Set $\alpha_{k+1} = \theta\alpha_k$ and $x_{k+1} = x_k$.

All the randomness comes from $\mathfrak{D}_k$

Randomization Techniques

Algorithm 2: Randomized Direct Search

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0, \infty)$, $0 < \theta < 1 \le \gamma$.

for $k = 0, 1, \ldots$ **do**

  Randomly generate $\mathfrak{D}_k \subset \mathbb{R}^n$.

  if $f(x_k + \alpha_k d) < f(x_k) - \rho(\alpha_k)$ for some $d \in \mathfrak{D}_k$ **then**

    Set $\alpha_{k+1} = \gamma\alpha_k$ and $x_{k+1} = x_k + \alpha_k d$.

  else

    Set $\alpha_{k+1} = \theta\alpha_k$ and $x_{k+1} = x_k$.

All the randomness comes from $\mathfrak{D}_k$

Notations for random variables or random vectors

$D_k \Rightarrow \mathfrak{D}_k$, $d \Rightarrow \mathfrak{d}$, $x_k \Rightarrow X_k$, $\alpha_k \Rightarrow A_k$, $g_k \Rightarrow G_k$

Assumptions on Randomness

Actually, what we need is not $\mathrm{cm}(D_k) \geq \kappa$ but $\mathrm{cm}(D_k, -G_k) \geq \kappa$.

Assumptions on Randomness

Actually, what we need is not $\mathrm{cm}(D_k) \geq \kappa$ but $\mathrm{cm}(D_k, -G_k) \geq \kappa$.

Definition (p-probabilistically κ-descent)

$(\mathfrak{D}_k)_{k\geq 0}$ is said to be p-probabilistically κ-descent, if

$$\mathbb{P}\left(\mathrm{cm}(\mathfrak{D}_k, -G_k) \geq \kappa \mid \mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1}\right) \;\geq\; p \quad \text{for each } k \geq 0.$$

Intuition:

each $\mathfrak{D}_k$ is “good enough with lower-bounded probability”,

no matter what happened before

Convergence of Probabilistic Direct Search

Theorem (Gratton et al. 2015)

If $(\mathfrak{D}_k)_{k\geq 0}$ *is* p*-probabilistically* κ*-descent with*

$$p = \log\theta / \log(\gamma^{-1}\theta),$$

then

$$\mathbb{P}\left(\liminf_{k\to\infty} \|G_k\| = 0\right) \;=\; 1.$$

Convergence of Probabilistic Direct Search

Theorem (Gratton et al. 2015)

If $(\mathfrak{D}_k)_{k\geq 0}$ *is* p*-probabilistically* κ*-descent with*

$$p = \log\theta / \log(\gamma^{-1}\theta),$$

then

$$\mathbb{P}\left(\liminf_{k\to\infty} \|G_k\| = 0\right) \;=\; 1.$$

Complexity: $\mathcal{O}(n\varepsilon^{-2})$ with overwhelmingly high probability

Typical Choice in Practice

Corollary (Gratton et al. 2015)

If $\mathfrak{D}_k = \{\mathfrak{d}_1, \ldots, \mathfrak{d}_m\}$, where $\mathfrak{d}_1, \ldots, \mathfrak{d}_m$ are independent random vectors uniformly distributed on the unit sphere in $\mathbb{R}^n$, then the algorithm is convergent if

$$m \; > \; \log_2\left(1 - \frac{\log\theta}{\log\gamma}\right).$$

Typical Choice in Practice

Corollary (Gratton et al. 2015)

If $\mathfrak{D}_k = \{\mathfrak{d}_1, \ldots, \mathfrak{d}_m\}$, where $\mathfrak{d}_1, \ldots, \mathfrak{d}_m$ are independent random vectors uniformly distributed on the unit sphere in $\mathbb{R}^n$, then the algorithm is convergent if

$$m \; > \; \log_2\left(1 - \frac{\log\theta}{\log\gamma}\right).$$

A natural question arises:

$$\text{what if } m \; < \; \log_2\left(1 - \frac{\log\theta}{\log\gamma}\right)?$$

Typical Choice in Practice

Corollary (Gratton et al. 2015)

If $\mathfrak{D}_k = \{\mathfrak{d}_1, \ldots, \mathfrak{d}_m\}$, where $\mathfrak{d}_1, \ldots, \mathfrak{d}_m$ are independent random vectors uniformly distributed on the unit sphere in $\mathbb{R}^n$, then the algorithm is convergent if

$$m \; > \; \log_2 \left(1 - \frac{\log \theta}{\log \gamma} \right).$$

A natural question arises:

$$\text{what if } m \; < \; \log_2 \left(1 - \frac{\log \theta}{\log \gamma} \right)?$$

"We must know. We will know."

Takeaways

- Try DFO solvers when you are dealing with tough problems!
- Try randomization techniques to reduce complexity!
- Try scanning the following two QR codes!

My Personal Website

Video on DFO

Thank you!

References I

- Biviano, A. et al. (2013). "CLASH-VLT: the mass, velocity-anisotropy, and pseudo-phase-space density profiles of the $z = 0.44$ galaxy cluster MACS J1206.2-0847". *A&A* 558, A1:1–A1:22.
- Conn, A. R., Scheinberg, K., and Vicente, L. N. (2009). *Introduction to Derivative-Free Optimization*. Vol. 8. MOS-SIAM Ser. Optim. Philadelphia: SIAM.
- Fermi, E. and Metropolis, N. (1952). *Numerical solution of a minimum problem*. Tech. rep. Alamos National Laboratory, Los Alamos, USA.
- Ghanbari, H. and Scheinberg, K. (2017). "Black-box optimization in machine learning with trust region based derivative free algorithm". *arXiv:1703.06925*.

References II

- Gratton, S. et al. (2015). “Direct search based on probabilistic descent”. *SIAM J. Optim.* 25, pp. 1515–1541.
- Nelder, J. A. and Mead, R. (1965). “A simplex method for function minimization”. *Comput. J.* 7, pp. 308–313.
- Powell, M. J. D. (1964). “An efficient method for finding the minimum of a function of several variables without calculating derivatives”. *Comput. J.* 7, pp. 155–162.