Optimization without derivatives in larger dimensions and across networks

Clément W. Royer (Université Paris Dauphine-PSL)

MIDAS Seminar Series - March 8, 2022

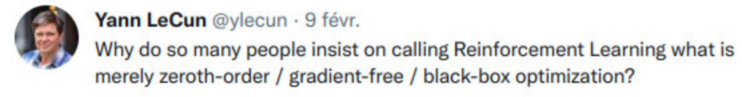
What I intend to talk about...

Optimization without derivatives, aka...

What I intend to talk about...

Optimization without derivatives, aka...

- **Derivative-free optimization (DFO);**
- **Zeroth-order optimization;**
- Black-box optimization;
- Simulation-based optimization;
- Hyperparameter tuning:
- **Reinforcement learning?**



Yann LeCun @ylecun · 9 févr.
Why do so many people insist on calling Reinforcement Learning what is merely zeroth-order / gradient-free / black-box optimization?

What I intend to talk about...

Optimization without derivatives, aka...

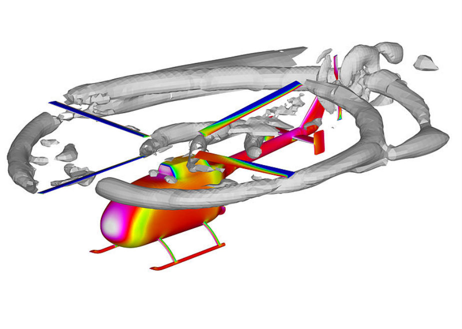
- Derivative-free optimization (DFO);
- Zeroth-order optimization;
- Black-box optimization;
- Simulation-based optimization;
- Hyperparameter tuning:
- Reinforcement learning?

Yann LeCun @ylecun · 9 févr.
Why do so many people insist on calling Reinforcement Learning what is merely zeroth-order / gradient-free / black-box optimization?

What I really intend to talk about

- My line of work (1+2);
- Hopefully relevant to others.

Classical DFO problem: Rotor helicopter design (Booker et al. 1998)

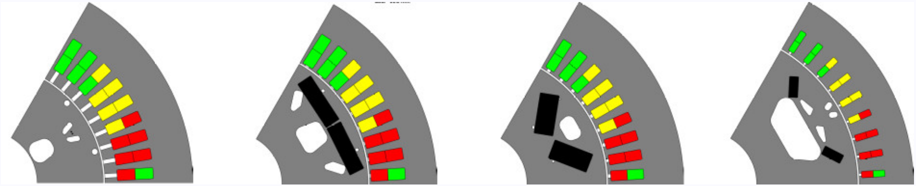


- About 30 parameters;
- 1 simulation: 2 weeks of computational fluid dynamics simulation;
- A simulation failed 60% of the time.

Ubiquitous in multidisciplinary optimization:

- Several codes interfaced;
- Numerical simulations;
- Large amount of calculation, possible failures.

Less classical example: Electrical engine design (D. Gaudrie, Stellantis)

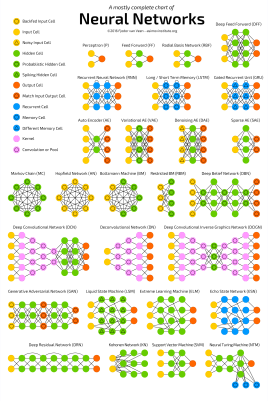


- About 50 continuous parameters;
- Multiobjective (3 functions), 6-dimensional constraint vector;
- Most points are infeasible!
- 1 simulation $\approx$ 5 minutes;
- Current practice: Run genetic algorithms **for 3 weeks**!

A modern challenge

Say you want to train a neural network...

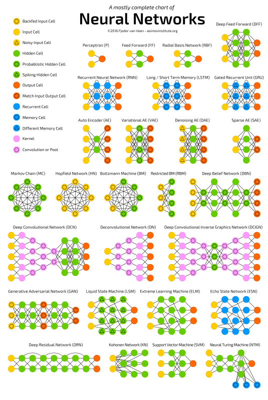
- What is your architecture? (Convolutional, Recurrent,etc)
- What is your training algorithm? (Adam, RMSProp, SG,etc)
- How do you choose your learning rate?



A modern challenge

Say you want to train a neural network...

- What is your architecture? (Convolutional, Recurrent,etc)
- What is your training algorithm? (Adam, RMSProp, SG,etc)
- How do you choose your learning rate?



Hyperparameter optimization

- Each change of hyperparameter involves another round of training (hours, days of CPU time + money!);
- Integer/Categorical/Continuous variables.

Goal: Reach automated ML!

Challenges for DFO

Scale up

- To millions of parameters? Maybe not...
- But a couple orders of magnitude may be helpful!

$\Rightarrow$ **Dimensionality reduction**.

Challenges for DFO

Scale up

- To millions of parameters? Maybe not...
- But a couple orders of magnitude may be helpful!

$\Rightarrow$ **Dimensionality reduction**.

Be data-oriented

- Expensive calculations involving massive amounts of data...
- ...possibly distributed on several memory nodes.

$\Rightarrow$ **Decentralized approaches**.

Roadmap

1. DFO and direct search
2. Direct search and reduced dimensions
3. Decentralizing direct search

Roadmap

Outline

Introductory assumptions and definitions

$$\text{minimize}_{\boldsymbol{x} \in \mathbb{R}^n} \quad f(\boldsymbol{x}).$$

Assumptions

- f bounded below;
- f continuously differentiable (nonconvex).

Blackbox/Derivative-free setup

- **Derivatives unavailable for algorithmic use.**
- Only access to values of f or stochastic estimates.
- f depends on expensive simulations/procedures.

Introductory assumptions and definitions

$$\text{minimize}_{\boldsymbol{x} \in \mathbb{R}^n} \quad f(\boldsymbol{x}).$$

Assumptions

- f bounded below;
- f continuously differentiable (nonconvex).

Blackbox/Derivative-free setup

- **Derivatives unavailable for algorithmic use.**
- Only access to values of f ~~or stochastic estimates~~.
- f depends on expensive simulations/procedures.

Complexity in blackbox optimization

My goal as a derivative-free/blackbox optimizer

Develop algorithms with controlled

- Number of calls to f;
- Dependency on n.

Complexity in blackbox optimization

My goal as a derivative-free/blackbox optimizer

Develop algorithms with controlled

- Number of calls to f;
- Dependency on n.

For this talk

Given $\epsilon \in (0,1)$ and, bound the number of **function evaluations** needed by a method to reach $\boldsymbol{x}$ such that

$$\|\nabla f(\boldsymbol{x})\| \leq \epsilon,$$

deterministically or in expectation/probability.

Complexity in blackbox optimization

My goal as a derivative-free/blackbox optimizer

Develop algorithms with controlled

- Number of calls to f;
- Dependency on n.

For this talk

Given $\epsilon \in (0,1)$ and, bound the number of **function evaluations** needed by a method to reach $\boldsymbol{x}$ such that

$$\|\nabla f(\boldsymbol{x})\| \leq \epsilon,$$

deterministically or in expectation/probability.

Focus: dependency w.r.t. n.

Two paradigms in derivative-free optimization

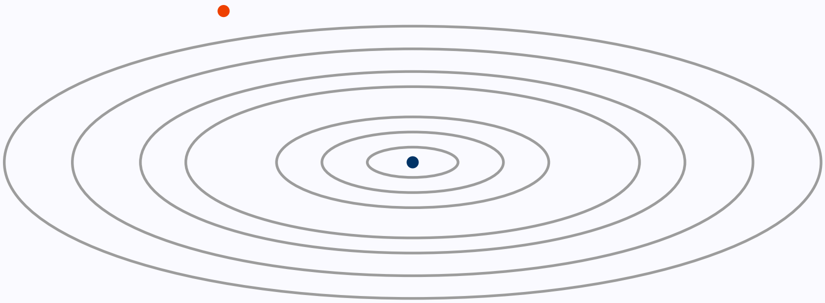
Model-based

- Build a model of the objective;
- Response surface, surrogate modeling, etc.

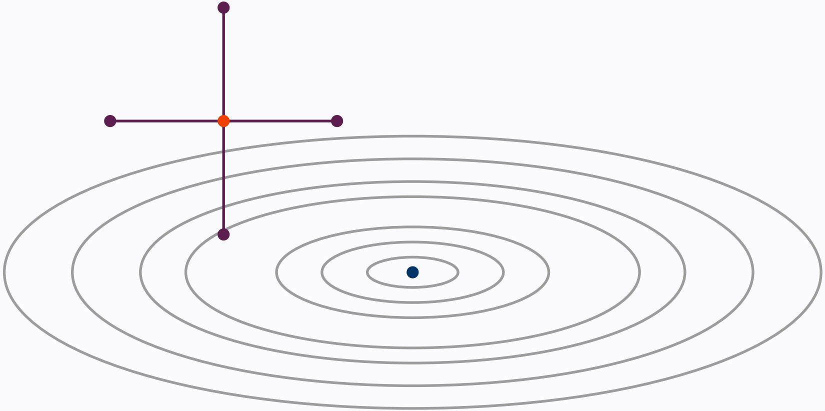
Direct search

- Sample along appropriate directions;
- Zeroth-order, random search, etc.

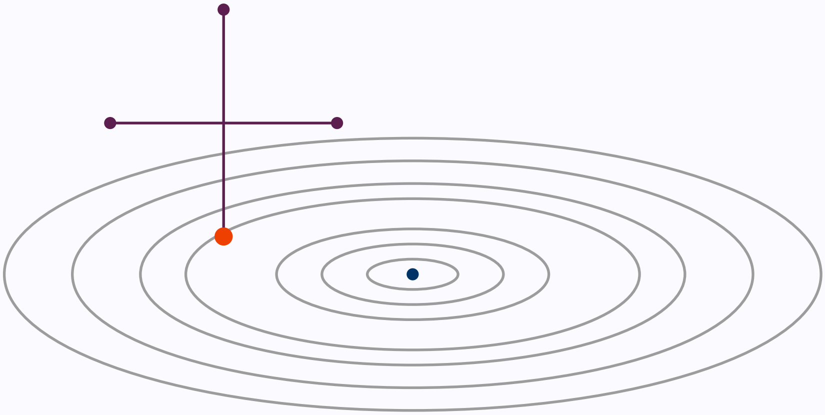
An example of DS : Coordinate Search

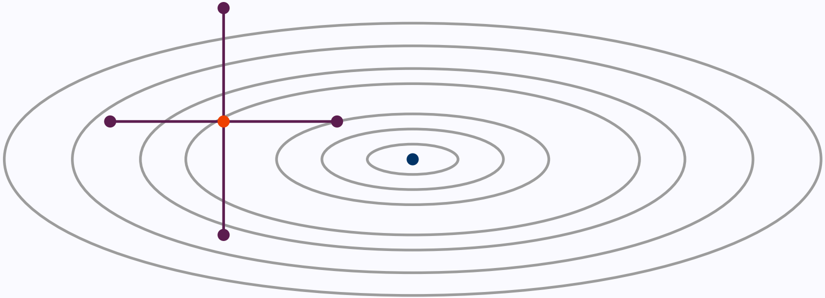


An example of DS : Coordinate Search

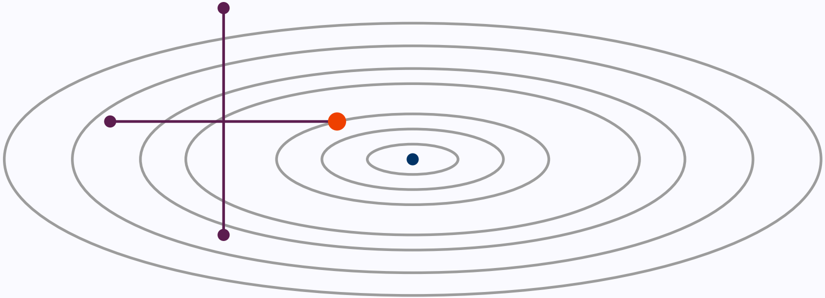


An example of DS : Coordinate Search

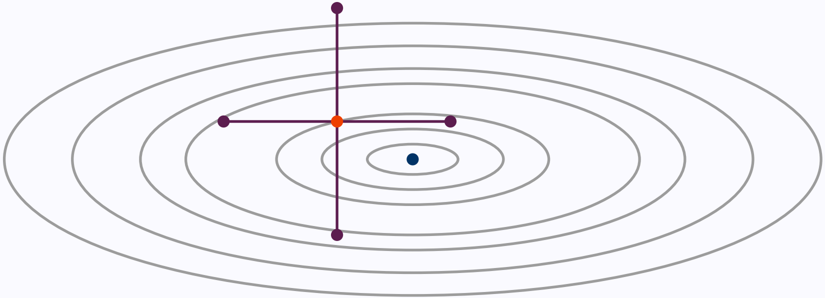




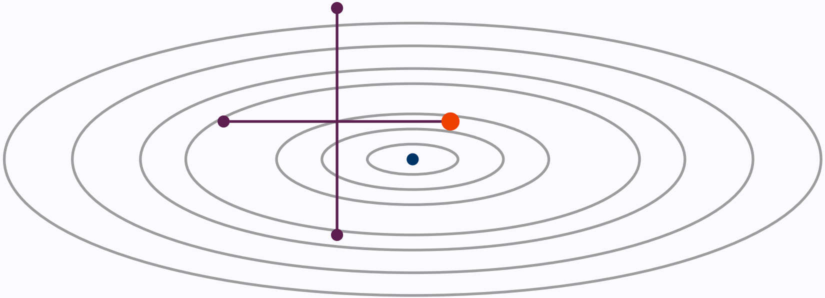
An example of DS : Coordinate Search



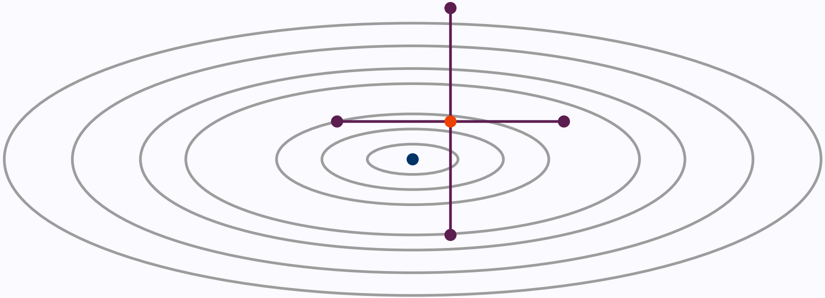
An example of DS : Coordinate Search



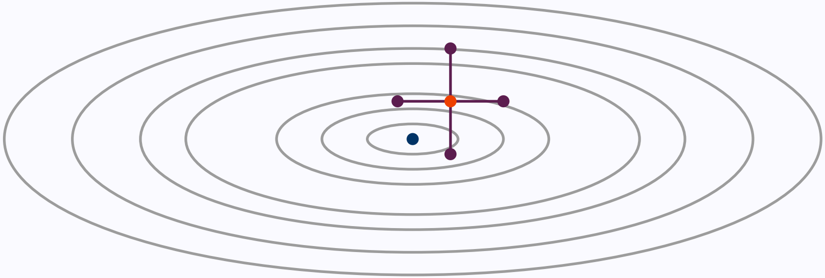
An example of DS : Coordinate Search



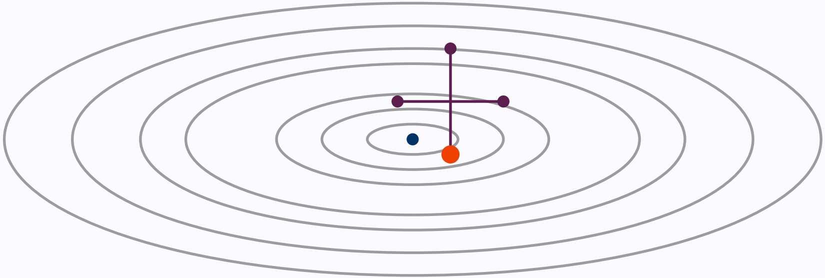
An example of DS : Coordinate Search



An example of DS : Coordinate Search



An example of DS : Coordinate Search



A simple direct-search framework

Inputs: $\boldsymbol{x}_0 \in \mathbb{R}^n$ $0 < \theta < 1 \leq \gamma$, $\alpha_0 > 0$.

Iteration k: Given $(\boldsymbol{x}_k, \alpha_k)$,

- Choose a set $\mathcal{D}_k \subset \mathbb{R}^n$ of m vectors.
- If $\exists$ $\boldsymbol{d}_k \in \mathcal{D}_k$ such that

$$f(\boldsymbol{x}_k + \alpha_k \, \boldsymbol{d}_k) < f(\boldsymbol{x}_k) - \alpha_k^2 \|\boldsymbol{d}_k\|^2$$

  set $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$, $\alpha_{k+1} := \gamma \alpha_k$.
- Otherwise, set $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k$, $\alpha_{k+1} := \theta \alpha_k$.

A simple direct-search framework

Inputs: $\boldsymbol{x}_0 \in \mathbb{R}^n$ $0 < \theta < 1 \leq \gamma$, $\alpha_0 > 0$.

Iteration k**:** Given $(\boldsymbol{x}_k, \alpha_k)$,

- Choose a set $\mathcal{D}_k \subset \mathbb{R}^n$ of m vectors.
- If $\exists$ $\boldsymbol{d}_k \in \mathcal{D}_k$ such that

$$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k) < f(\boldsymbol{x}_k) - \alpha_k^2 \|\boldsymbol{d}_k\|^2$$

  set $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$, $\alpha_{k+1} := \gamma \alpha_k$.
- Otherwise, set $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k$, $\alpha_{k+1} := \theta \alpha_k$.

A simple direct-search framework

Inputs: $\boldsymbol{x}_0 \in \mathbb{R}^n$ $0 < \theta < 1 \leq \gamma$, $\alpha_0 > 0$.

Iteration k**:** Given $(\boldsymbol{x}_k, \alpha_k)$,

- **Choose a set $\mathcal{D}_k \subset \mathbb{R}^n$ of m vectors.**
- If $\exists$ $\boldsymbol{d}_k \in \mathcal{D}_k$ such that

$$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k) < f(\boldsymbol{x}_k) - \alpha_k^2 \|\boldsymbol{d}_k\|^2$$

  set $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$, $\alpha_{k+1} := \gamma \alpha_k$.
- Otherwise, set $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k$, $\alpha_{k+1} := \theta \alpha_k$.

Which vectors should we use?

Choosing $\mathcal{D}_k$

A measure of set quality

The set $\mathcal{D}_k$ is called κ-descent for f at $\boldsymbol{x}_k$ if

$$\max_{\boldsymbol{d} \in \mathcal{D}_k} \frac{-\boldsymbol{d}^{\mathrm{T}} \nabla f(\boldsymbol{x}_k)}{\|\boldsymbol{d}\| \|\nabla f(\boldsymbol{x}_k)\|} \ \geq \ \kappa \in (0, 1].$$

Choosing $\mathcal{D}_k$

A measure of set quality

The set $\mathcal{D}_k$ is called κ-descent for f at $\boldsymbol{x}_k$ if

$$\max_{\boldsymbol{d} \in \mathcal{D}_k} \frac{-\boldsymbol{d}^\mathrm{T} \nabla f(\boldsymbol{x}_k)}{\|\boldsymbol{d}\| \|\nabla f(\boldsymbol{x}_k)\|} \ \geq \ \kappa \in (0, 1].$$

- Guaranteed when $\mathcal{D}_k$ is a Positive Spanning Set (PSS);
- $\mathcal{D}_k$ PSS $\Rightarrow |\mathcal{D}_k| \geq n + 1$;
- Ex) $\mathcal{D}_\oplus := \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_n, \text{-}\boldsymbol{e}_1, \dots, \text{-}\boldsymbol{e}_n\}$ is always $\frac{1}{\sqrt{n}}$-descent.

Key convergence arguments in direct search

Assumption: For every k, $\mathcal{D}_k$ is κ-descent and contains m unit directions.

Small step size $\Rightarrow$ Success

If

$$\alpha_k < \mathcal{O}\left(\kappa \|\nabla f(\boldsymbol{x}_k)\|\right),$$

then $\boldsymbol{x}_{k+1} \neq \boldsymbol{x}_k$ and $\alpha_{k+1} \geq \alpha_k$.

Key convergence arguments in direct search

Assumption: For every k, $\mathcal{D}_k$ is κ-descent and contains m unit directions.

Small step size ⇒ Success

If

$$\alpha_k < \mathcal{O}\left(\kappa \|\nabla f(\boldsymbol{x}_k)\|\right),$$

then $\boldsymbol{x}_{k+1} \neq \boldsymbol{x}_k$ and $\alpha_{k+1} \geq \alpha_k$.

Step size goes to zero

Independently of the κ-descent property,

$$\exists \beta \in (0, \infty), \qquad \sum_{k=0}^{\infty} \alpha_k^2 < \beta < \infty \quad \left(\Rightarrow \lim_{k \to \infty} \alpha_k = 0 \right).$$

Worst-case complexity in deterministic direct search

Assumption: For every k, $\mathcal{D}_k$ is κ-descent and contains m unit directions.

Theorem

Let $\epsilon \in (0,1)$ and N_ϵ be the number of function evaluations needed to reach $\boldsymbol{x}_k$ such that $\|\nabla f(\boldsymbol{x}_k)\| \le \epsilon$. Then,

$$N_\epsilon \;\le\; \mathcal{O}\left(m\,\kappa^{-2}\,\epsilon^{-2}\right).$$

Worst-case complexity in deterministic direct search

Assumption: For every k, $\mathcal{D}_k$ is κ-descent and contains m unit directions.

Theorem

Let $\epsilon \in (0,1)$ and N_ϵ be the number of function evaluations needed to reach $\boldsymbol{x}_k$ such that $\|\nabla f(\boldsymbol{x}_k)\| \le \epsilon$. Then,

$$N_\epsilon \;\le\; \mathcal{O}\left(m\,\kappa^{-2}\,\epsilon^{-2}\right).$$

- Unit norm can be replaced by bounded norm.
- Choosing $\mathcal{D}_k = \mathcal{D}_\oplus$, one has $\kappa = \frac{1}{\sqrt{n}}$, $m = 2n$, and the bound becomes
$$N_\epsilon \;\le\; \mathcal{O}\left(n^2\,\epsilon^{-2}\right).$$
- Optimal in the power of n for **deterministic** direct-search algorithms.

Outline

1. DFO and direct search
   - Deterministic direct search
   - Direct search based on probabilistic descent
2. Direct search and reduced dimensions
3. Decentralizing direct search

A probabilistic property

Deterministic descent

The set $\mathcal{D}_k$ is κ-descent for $(f, \boldsymbol{x}_k)$ if

$$\max_{\boldsymbol{d} \in \mathcal{D}_k} \frac{-\nabla f(\boldsymbol{x}_k)^\top \boldsymbol{d}}{\|\nabla f(\boldsymbol{x}_k)\| \|\boldsymbol{d}\|} \ \geq \ \kappa \in (0, 1].$$

A probabilistic property

Deterministic descent

The set $\mathcal{D}_k$ is κ-descent for $(f, \boldsymbol{x}_k)$ if

$$\max_{\boldsymbol{d} \in \mathcal{D}_k} \frac{-\nabla f(\boldsymbol{x}_k)^\top \boldsymbol{d}}{\|\nabla f(\boldsymbol{x}_k)\| \|\boldsymbol{d}\|} \geq \kappa \in (0, 1].$$

Probabilistic descent

The sequence $\{\mathcal{D}_k\}$ is said to be (p, κ)-descent if:

$$\begin{aligned} \mathbb{P}\left(\mathcal{D}_0 \ \kappa\text{-descent}\ \right) &\geq p \\ \forall k \geq 1, \quad \mathbb{P}\left(\mathcal{D}_k \ \kappa\text{-descent}\ \middle|\ \mathcal{D}_0, \dots, \mathcal{D}_{k-1}\right) &\geq p, \end{aligned}$$

Key arguments in probabilistic direct search (1/2)

Assumption: For every k, $\mathcal{D}_k$ contains m unit directions.

Small step size ⇒ Success

If $\mathcal{D}_k$ is κ-descent and

$$\alpha_k < \mathcal{O}\left(\kappa \|\nabla f(\boldsymbol{x}_k)\|\right).$$

then $\boldsymbol{x}_{k+1} \neq \boldsymbol{x}_k$ and $\alpha_{k+1} \geq \alpha_k$.

Step size goes to zero

For all realizations of the method,

$$\sum_{k=0}^{\infty} \alpha_k^2 < \infty$$

Key arguments for probabilistic direct search (2/2)

A useful bound

Let $z_k = 1(\mathcal{D}_k\ \kappa\text{-descent})$. For all realizations of the algorithm, one has

$$\sum_{\ell=0}^{k-1} z_\ell \ \leq\ \mathcal{O}\left(\frac{1}{\kappa^2\ \left(\min_{0\leq\ell\leq k} \|\nabla f(\boldsymbol{x}_\ell)\|\right)^2}\right) + p_0\, k,$$

with $p_0 = \frac{1}{1+\mu}$, $\mu = \log_\theta(1/\gamma)$.

Key arguments for probabilistic direct search (2/2)

A useful bound

Let $z_k = 1\,(\mathcal{D}_k\ \kappa\text{-descent})$. For all realizations of the algorithm, one has

$$\sum_{\ell=0}^{k-1} z_\ell \;\le\; \mathcal{O}\left(\frac{1}{\kappa^2\,\left(\min_{0\le\ell\le k}\|\nabla f(\boldsymbol{x}_\ell)\|\right)^2}\right) + p_0\,k,$$

with $p_0 = \frac{1}{1+\mu}$, $\mu = \log_\theta(1/\gamma)$.

- $\mathbb{P}(z_\ell = 1 | z_0,\dots,z_{\ell-1}) \ge p$ by assumption;
- $\{\sum_{\ell=0}^{k-1} z_\ell\}_k$ is a submartingale;
- As long as $p > p_0$, can relate the behavior of $\|\nabla f(\boldsymbol{x}_k)\|$ and that of $\sum_{\ell=0}^{k-1} z_\ell$.

Assumptions:

- $\{\mathcal{D}_k\}$ (p,κ)-descent, $p > p_0$.
- $\mathcal{D}_k$ contains m unit vectors.

Probabilistic worst-case complexity (Gratton et al, '15)

Let $\epsilon \in (0,1)$ and N_ϵ the number of function evaluations needed to have $\|\nabla f(x_k)\| \leq \epsilon$. Then

$$\mathbb{P}\left(N_\epsilon \leq \mathcal{O}\left(\frac{m\,\kappa^{-2}\epsilon^{-2}}{p-p_0}\right)\right) \geq 1 - \exp\left(-\mathcal{O}\left(\frac{p-p_0}{p}(\kappa\,\epsilon)^{-2}\right)\right).$$

Complexity results

Assumptions:

- $\{\mathcal{D}_k\}$ (p,κ)-descent, $p > p_0$.
- $\mathcal{D}_k$ contains m unit vectors.

Probabilistic worst-case complexity (Gratton et al, '15)

Let $\epsilon \in (0,1)$ and N_ϵ the number of function evaluations needed to have $\|\nabla f(x_k)\| \le \epsilon$. Then

$$\mathbb{P}\left(N_\epsilon \le \mathcal{O}\left(\frac{m\,\kappa^{-2}\epsilon^{-2}}{p-p_0}\right)\right) \ge 1 - \exp\left(-\mathcal{O}\left(\frac{p-p_0}{p}(\kappa\,\epsilon)^{-2}\right)\right).$$

Expected evaluation complexity

$$\mathbb{E}[N_\epsilon] \;\le\; \mathcal{O}\left(\frac{m\kappa^{-2}\epsilon^{-2}}{p-p_0}\right) + \mathcal{O}(m).$$

A practical (p, κ)-descent sequence

Using 2 directions uniformly distributed over the unit sphere.

- Defines a $(p, \tau/\sqrt{n})$-descent sequence, $p > 1/2$.
- Optimal (largest τ): Choose opposite directions!

A practical (p, κ)-descent sequence

Using 2 directions uniformly distributed over the unit sphere.

- Defines a $(p, \tau/\sqrt{n})$-descent sequence, $p > 1/2$.
- Optimal (largest τ): Choose opposite directions!

Complexity bound

- Deterministic: $m = \mathcal{O}(n) \Rightarrow \mathcal{O}(n^2\,\epsilon^{-2})$.
- Probabilistic $m = \mathcal{O}(1) \Rightarrow \mathcal{O}(n\,\epsilon^{-2})$.

$\Rightarrow$ Factor n improvement at the iteration level.

Not the only game in town

Gaussian smoothing approach: Draw $\boldsymbol{u}_k \sim \mathcal{N}(0, \boldsymbol{I})$ and use

$$\frac{f(\boldsymbol{x}+\mu\boldsymbol{u}) - f(\boldsymbol{x})}{\mu}\boldsymbol{u} \quad \text{or} \quad \frac{f(\boldsymbol{x}+\mu\boldsymbol{u}) - f(\boldsymbol{x}-\mu\boldsymbol{u})}{\mu}\boldsymbol{u}.$$

Random gradient-free method (Nesterov and Spokoiny 2017),
Stochastic three-point method (Bergou et al, 2020).
$\Rightarrow$ Both achieve an $\mathcal{O}(n\epsilon^{-2})$ bound with predefined stepsizes.

Not the only game in town

Gaussian smoothing approach: Draw $\boldsymbol{u}_k \sim \mathcal{N}(0, \boldsymbol{I})$ and use

$$\frac{f(\boldsymbol{x}+\mu\boldsymbol{u})-f(\boldsymbol{x})}{\mu}\boldsymbol{u} \quad \text{or} \quad \frac{f(\boldsymbol{x}+\mu\boldsymbol{u})-f(\boldsymbol{x}-\mu\boldsymbol{u})}{\mu}\boldsymbol{u}.$$

Random gradient-free method (Nesterov and Spokoiny 2017),
Stochastic three-point method (Bergou et al, 2020).
$\Rightarrow$ Both achieve an $\mathcal{O}(n\epsilon^{-2})$ bound with predefined stepsizes.

- Gaussian directions are not always bounded $\Rightarrow$ Probabilistic analysis does not apply.
- Same complexity but different directions $\Rightarrow$ Can we provide a unified framework?

Outline

Back to original direct search

Recall: Classical direct search

- Set $\mathcal{D}_k \subset \mathbb{R}^n$, $|\mathcal{D}_k| = m$, $\mathrm{cm}(\mathcal{D}_k) \geq \kappa$;
- Complexity:

$$\mathcal{O}(m\kappa^{-2}\,\epsilon^{-2}).$$

  - m may not depend on n (probabilistic)
  - ...but κ depends on n (approximate $\nabla f(\boldsymbol{x}_k) \in \mathbb{R}^n$).

Meanwhile...

- Random embeddings (Cartis et al 2020, 2021);
- Random subspaces (Gratton et al, Kozak et al. 2021).

Reduce the dependency on n by working on low dimensions.

A new start (with Lindon Roberts)

Idea

- Consider a random subspace of dimension $r \leq n$;
- Use a PSS to approximate the projected gradient in the subspace;
- Guarantee sufficient gradient information **in probability**.

What it brings us

- Handle **unbounded directions**;
- Revisit the opposite uniform directions choice;
- Generalize the analysis to other settings, e.g. Gaussian.

Algorithm

Inputs: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\alpha_0 > 0$.

Iteration k: Given $(\boldsymbol{x}_k, \alpha_k)$,

- Choose $\boldsymbol{P}_k \in \mathbb{R}^{r \times n}$ **at random**.
- Choose $\mathcal{D}_k \subset \mathbb{R}^r$ having m vectors.
- If $\exists\ \boldsymbol{d}_k \in \mathcal{D}_k$ such that

$$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{P}_k^{\mathrm{T}} \boldsymbol{d}_k) < f(\boldsymbol{x}_k) - \alpha_k^2 \|\boldsymbol{P}_k^{\mathrm{T}} \boldsymbol{d}_k\|^2,$$

  set $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k \boldsymbol{P}_k^{\mathrm{T}} \boldsymbol{d}_k$, $\alpha_{k+1} := \gamma \alpha_k$.
- Otherwise, set $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k$, $\alpha_{k+1} := \theta \alpha_k$.

Probabilistic properties

New polling sets

$$\left\{\boldsymbol{P}_k^{\mathrm{T}}\boldsymbol{d} \;\middle|\; \boldsymbol{d} \in \mathcal{D}_k\right\} \subset \mathbb{R}^n.$$

- $\boldsymbol{P}_k \in \mathbb{R}^{r \times n}$: Maps onto r-dimensional subspace;
- $\mathcal{D}_k$: Direction set in the subspace.

What do we want?

- Preserve information while applying $\boldsymbol{P}_k/\boldsymbol{P}_k^{\mathrm{T}}$.
- Approximate $-\boldsymbol{P}_k\nabla f(\boldsymbol{x}_k)$ using $\mathcal{D}_k$.

Probabilistic properties for $\boldsymbol{P}_k$

$\boldsymbol{P}_k$ is $(\eta, \sigma, P_{\max})$-well aligned for $(f, \boldsymbol{x}_k)$ if

$$\left\{ \begin{array}{rcl} \|\boldsymbol{P}_k \nabla f(\boldsymbol{x}_k)\| & \geq & \eta \|\nabla f(\boldsymbol{x}_k)\|, \\ \sigma_{\min}(\boldsymbol{P}_k) & \geq & \sigma, \\ \sigma_{\max}(\boldsymbol{P}_k) & \leq & P_{\max}. \end{array} \right.$$

Probabilistic properties for $\boldsymbol{P}_k$

$\boldsymbol{P}_k$ is $(\eta, \sigma, P_{\max})$-well aligned for $(f, \boldsymbol{x}_k)$ if

$$\left\{ \begin{array}{rcl} \|\boldsymbol{P}_k \nabla f(\boldsymbol{x}_k)\| & \geq & \eta \|\nabla f(\boldsymbol{x}_k)\|, \\ \sigma_{\min}(\boldsymbol{P}_k) & \geq & \sigma, \\ \sigma_{\max}(\boldsymbol{P}_k) & \leq & P_{\max}. \end{array} \right.$$

Ex) $\boldsymbol{P}_k = \boldsymbol{I} \in \mathbb{R}^{n \times n}$ *is* $(1, 1, 1)$*-well aligned.*

Probabilistic properties for $\boldsymbol{P}_k$

$\boldsymbol{P}_k$ is $(\eta, \sigma, P_{\max})$-well aligned for $(f, \boldsymbol{x}_k)$ if

$$\left\{ \begin{array}{rcl} \|\boldsymbol{P}_k \nabla f(\boldsymbol{x}_k)\| & \geq & \eta \|\nabla f(\boldsymbol{x}_k)\|, \\ \sigma_{\min}(\boldsymbol{P}_k) & \geq & \sigma, \\ \sigma_{\max}(\boldsymbol{P}_k) & \leq & P_{\max}. \end{array} \right.$$

Ex) $\boldsymbol{P}_k = \boldsymbol{I} \in \mathbb{R}^{n \times n}$ *is* $(1,1,1)$*-well aligned.*

Probabilistic version

$\{\boldsymbol{P}_k\}$ is $(q, \eta, \sigma, P_{\max})$-well aligned if:

$$\begin{array}{rcl} \mathbb{P}\left(\boldsymbol{P}_0 \ (q, \eta, \sigma, P_{\max})\text{-well aligned }\right) & \geq & q \\ \forall k \geq 1, \quad \mathbb{P}\left((q, \eta, \sigma, P_{\max})\text{-well aligned } \mid \boldsymbol{P}_0, \mathcal{D}_0, \dots, \boldsymbol{P}_{k-1}, \mathcal{D}_{k-1}\right) & \geq & q, \end{array}$$

Probabilistic properties for $\mathcal{D}_k$

Deterministic descent

The set $\mathcal{D}_k$ is $(\kappa, d_{\max})$-descent for $(f, \boldsymbol{x}_k)$ if

$$\begin{cases} \max_{\boldsymbol{d} \in \mathcal{D}_k} \frac{-\boldsymbol{d}^{\mathrm{T}} \boldsymbol{P}_k \nabla f(\boldsymbol{x}_k)}{\|\boldsymbol{d}\| \|\boldsymbol{P}_k \nabla f(\boldsymbol{x}_k)\|} \geq \kappa, \\ \forall \boldsymbol{d} \in \mathcal{D}_k, \quad d_{\max}^{-1} \leq \|\boldsymbol{d}\| \leq d_{\max}. \end{cases}$$

Probabilistic properties for $\mathcal{D}_k$

Deterministic descent

The set $\mathcal{D}_k$ is $(\kappa, d_{\max})$-descent for $(f, \boldsymbol{x}_k)$ if

$$\begin{cases} \max_{\boldsymbol{d} \in \mathcal{D}_k} \frac{-\boldsymbol{d}^{\mathrm{T}} \boldsymbol{P}_k \nabla f(\boldsymbol{x}_k)}{\|\boldsymbol{d}\| \|\boldsymbol{P}_k \nabla f(\boldsymbol{x}_k)\|} \geq \kappa, \\ \forall \boldsymbol{d} \in \mathcal{D}_k, \quad d_{\max}^{-1} \leq \|\boldsymbol{d}\| \leq d_{\max}. \end{cases}$$

Ex) $D_{\oplus}$ is $(\frac{1}{\sqrt{n}}, 1)$-descent.

Probabilistic properties for $\mathcal{D}_k$

Deterministic descent

The set $\mathcal{D}_k$ is $(\kappa, d_{\max})$-descent for $(f, \boldsymbol{x}_k)$ if

$$\begin{cases} \max_{\boldsymbol{d} \in \mathcal{D}_k} \frac{-\boldsymbol{d}^{\mathrm{T}} \boldsymbol{P}_k \nabla f(\boldsymbol{x}_k)}{\|\boldsymbol{d}\| \|\boldsymbol{P}_k \nabla f(\boldsymbol{x}_k)\|} \geq \kappa, \\ \forall \boldsymbol{d} \in \mathcal{D}_k, \quad d_{\max}^{-1} \leq \|\boldsymbol{d}\| \leq d_{\max}. \end{cases}$$

Ex) $D_{\oplus}$ is $(\frac{1}{\sqrt{n}}, 1)$-descent.

Probabilistic descent sets

$\{\mathcal{D}_k\}$ is $(p, \kappa, d_{\max})$-descent if:

$$\begin{aligned} \mathbb{P}\left(\mathcal{D}_0 \ (\kappa, d_{\max})\text{-descent} \mid \boldsymbol{P}_0\right) &\geq p \\ \forall k \geq 1, \quad \mathbb{P}\left(\mathcal{D}_k \ (\kappa, d_{\max})\text{-descent} \mid \boldsymbol{P}_0, \mathcal{D}_0, \ldots, \boldsymbol{P}_{k-1}, \mathcal{D}_{k-1}, \boldsymbol{P}_k\right) &\geq p, \end{aligned}$$

Key arguments

Small step size + Good $\boldsymbol{P}_k/\mathcal{D}_k \Rightarrow$ Success

If $\boldsymbol{P}_k$ is $(\eta, \sigma, P_{\max})$-well aligned, $\mathcal{D}_k$ is $(\kappa, d_{\max})$-descent, and

$$\alpha_k < \mathcal{O}\left(\frac{\kappa\eta}{P_{\max}^2 d_{\max}^3}\|\nabla f(\boldsymbol{x}_k)\|\right).$$

then $\boldsymbol{x}_{k+1} \neq \boldsymbol{x}_k$ and $\alpha_{k+1} \geq \alpha_k$.

Key arguments

Small step size + Good $\mathbf{P}_k/\mathcal{D}_k \Rightarrow$ Success

If $\mathbf{P}_k$ is $(\eta, \sigma, P_{\max})$-well aligned, $\mathcal{D}_k$ is $(\kappa, d_{\max})$-descent, and

$$\alpha_k < \mathcal{O}\left(\frac{\kappa\eta}{P_{\max}^2 d_{\max}^3}\|\nabla f(\mathbf{x}_k)\|\right).$$

then $\mathbf{x}_{k+1} \neq \mathbf{x}_k$ and $\alpha_{k+1} \geq \alpha_k$.

A step size sequence goes to zero

For all realizations of the method,

$$\sum_{k \in \mathcal{K}} \alpha_k^2 < \mathcal{O}\left(\frac{d_{\max}^2}{\sigma^2}\right) < \infty,$$

where $\mathcal{K}$ is the set of succesful iterations for which $\mathbf{P}_k$ is $(\eta, \sigma, P_{\max})$-well aligned and $\mathcal{D}_k$ is $(\kappa, d_{\max})$-descent.

Martingale argument

Proposition

Define

$$z_k = 1\left(\mathcal{D}_0\ (\kappa, d_{\max})\text{-descent and } \boldsymbol{P}_0\ (q, \eta, \sigma, P_{\max})\text{-well aligned}\right).$$

For all realizations of the algorithm, one has

$$\sum_{l=0}^{k} z_\ell \ \leq\ \mathcal{O}\left(\frac{1}{\left(\min_{0 \leq l \leq k} \|\nabla f(\boldsymbol{x}_\ell)\|\right)^2}\right) + p_0\, k$$

with $p_0 = \max\left\{\frac{1}{1+\mu}, \frac{\mu}{1+\mu}\right\}$ and $\mu = \log_\gamma(1/\theta)$.

Martingale argument

Proposition

Define

$$z_k = \mathbb{1}\left(\mathcal{D}_0 \ (\kappa, d_{\max})\text{-descent and } \boldsymbol{P}_0 \ (q, \eta, \sigma, P_{\max})\text{-well aligned}\right).$$

For all realizations of the algorithm, one has

$$\sum_{l=0}^{k} z_\ell \ \leq \ \mathcal{O}\left(\frac{1}{\left(\min_{0 \leq l \leq k} \|\nabla f(\boldsymbol{x}_\ell)\|\right)^2}\right) + p_0\, k$$

with $p_0 = \max\left\{\frac{1}{1+\mu}, \frac{\mu}{1+\mu}\right\}$ and $\mu = \log_\gamma(1/\theta)$.

- $\sum_\ell z_\ell$ satisfies a concentration bound;
- Best case: $\theta = \gamma^{-1} = 1/2$.

Complexity analysis

Theorem (Roberts and Royer, 2022)

Assume:

- $\{\mathcal{D}_k\}$ $(p, \kappa, d_{\max})$-descent, $|\mathcal{D}_k| = m$;
- $\{\boldsymbol{P}_k\}$ $(q, \eta, \sigma, P_{\max})$-well aligned.

Let N_ϵ the number of function evaluations needed to have $\|\nabla f(\boldsymbol{x}_k)\| \le \epsilon$.

$$\mathbb{P}\left(N_\epsilon \le \mathcal{O}\left(\frac{m\phi\epsilon^{-2}}{pq - p_0}\right)\right) \ge 1 - \exp\left(-\mathcal{O}\left(\frac{pq - p_0}{pq}\phi\epsilon^{-2}\right)\right).$$

where $\phi = \eta^{-2}\sigma^{-2}P_{\max}^4 d_{\max}^8 \kappa^{-2}$.

Complexity analysis

Theorem (Roberts and Royer, 2022)

Assume:

- $\{\mathcal{D}_k\}$ $(p, \kappa, d_{\max})$-descent, $|\mathcal{D}_k| = m$;
- $\{\boldsymbol{P}_k\}$ $(q, \eta, \sigma, P_{\max})$-well aligned.

Let N_ϵ the number of function evaluations needed to have $\|\nabla f(\boldsymbol{x}_k)\| \le \epsilon$.

$$\mathbb{P}\left(N_\epsilon \le \mathcal{O}\left(\frac{m\phi\epsilon^{-2}}{pq - p_0}\right)\right) \ge 1 - \exp\left(-\mathcal{O}\left(\frac{pq - p_0}{pq}\phi\epsilon^{-2}\right)\right).$$

where $\phi = \eta^{-2}\sigma^{-2}P_{\max}^4 d_{\max}^8 \kappa^{-2}$.

How does this bound depend on n?

Can we really improve the dimension dependence?

$$m\eta^{-2}\sigma^{-2}P_{\max}^4 d_{\max}^8 \kappa^{-2}\epsilon^{-2}.$$

Can we really improve the dimension dependence?

$$m\eta^{-2}\sigma^{-2}P_{\max}^4 d_{\max}^8 \kappa^{-2}\epsilon^{-2}.$$

A first simplification

- $\mathcal{D}_k = \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_r, -\boldsymbol{e}_1, \dots, -\boldsymbol{e}_r\}$ in $\mathbb{R}^r$;
- $\kappa = \frac{1}{\sqrt{r}}$, $m = 2r$.

$\Rightarrow$ Bound becomes $2r^2\eta^{-2}\sigma^{-2}P_{\max}^4\epsilon^{-2}$.

Can we really improve the dimension dependence?

$$m\eta^{-2}\sigma^{-2}P_{\max}^4 d_{\max}^8 \kappa^{-2}\epsilon^{-2}.$$

A first simplification

- $\mathcal{D}_k = \{\mathbf{e}_1, \dots, \mathbf{e}_r, -\mathbf{e}_1, \dots, -\mathbf{e}_r\}$ in $\mathbb{R}^r$;
- $\kappa = \frac{1}{\sqrt{r}}$, $m = 2r$.

$\Rightarrow$ Bound becomes $2r^2\eta^{-2}\sigma^{-2}P_{\max}^4\epsilon^{-2}$.

Using sketching techniques

$\mathbf{P}_k$	σ	$P_{\max}$
Identity	1	1
Gaussian	$\Theta(\sqrt{n/r})$	$\Theta(\sqrt{n/r})$
Hashing	$\Theta(\sqrt{n/r})$	$\sqrt{n}$
Orthogonal	$\sqrt{n/r}$	$\sqrt{n/r}$.

$\Rightarrow$ Get a bound in $\mathcal{O}(n\epsilon^{-2})$ even when $r = \mathcal{O}(1)$ and $\eta = \mathcal{O}(1)$!

Outline

1. DFO and direct search
2. Direct search and reduced dimensions
   - Algorithm and complexity
   - Numerical illustration
3. Decentralizing direct search

Experiments in larger dimensions

Benchmark:

- Medium-scale test set (90 CUTEst problems of dimension ≈ 100);
- Large-scale test set (28 CUTEst problems of dimension ≈ 1000).

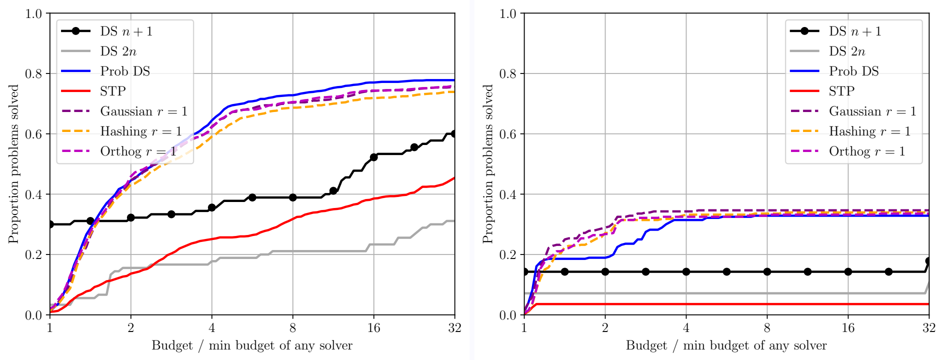
Budget: $200(n+1)$ evaluations.

Comparison:

- Deterministic methods with $\mathcal{D}_k = \mathcal{D}_{\oplus}$ or $\mathcal{D}_k = \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_n, -\sum_{i=1}^n \boldsymbol{e}_i\}$;
- Probabilistic direct search with 2 directions;
- Stochastic Three Point;
- Reduced dimension methods with Gaussian/Hashing/Orthogonal $\boldsymbol{P}_k$ matrices + 2 directions in the subspace.

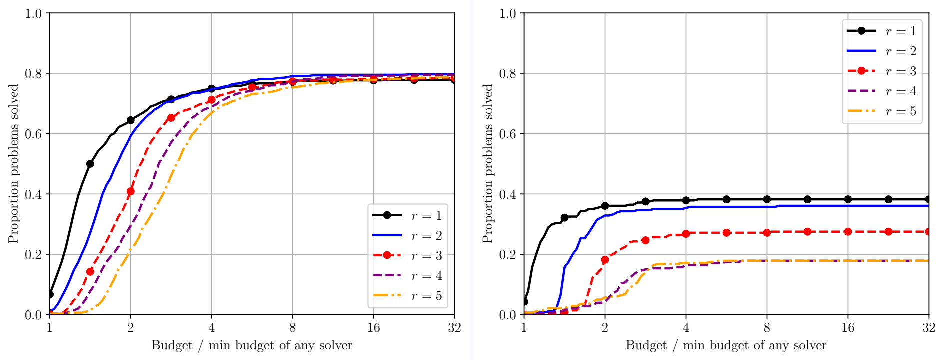
Goal: Satisfy $f(\boldsymbol{x}_k) - f_{best} \leq 0.1(f(\boldsymbol{x}_0) - f_{best})$.

Comparison of all methods



Left: Medium scale; Right: Large scale.

Gaussian matrices and the value of r



Left: Medium scale; Right: Large scale.

Summary of our findings

If you want to scale up...

- Can use less directions through sketching;
- But always a (hidden) dependency on n!

Numerically

- Sketches of dimension > 1 may improve things...
- ...but in general opposite Gaussian directions are quite good!

Outline

Setup

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\, f(\boldsymbol{x}) := \sum_{i=1}^{N} f_i(\boldsymbol{x}).$$

- All f_is are $\mathcal{C}^{1,1}$, $\sum_{i=1}^N f_i$ bounded below.
- Data for computing f_i is stored locally by an agent.
- N agents communicate through a network/graph.

Network structure

- Doubly stochastic matrix $\boldsymbol{W} = [w_{ij}] \in \mathbb{R}^{N\times N}$;
- $\mathcal{N}_i$: set of neighbors of agent i;
- $w_{ij} \neq 0$ iff $i = j$ or $j \in \mathcal{N}_i$.

Popular approach: Consensus optimization

- Each agent i has a local vector $\boldsymbol{x}^{(i)}$;
- Agent i updates $\boldsymbol{x}^{(i)}$ and communicates with its neighbors;
- Goal:

$$\underbrace{\sum_{i=1}^{N} \nabla f(\boldsymbol{x}^{(i)}) = 0,}_{\text{optimality}} \qquad \underbrace{\boldsymbol{x}^{(i)} = \sum_{j=1}^{N} w_{ij} \boldsymbol{x}^{(j)}}_{\text{consensus}}.$$

Popular approach: Consensus optimization

- Each agent i has a local vector $\boldsymbol{x}^{(i)}$;
- Agent i updates $\boldsymbol{x}^{(i)}$ and communicates with its neighbors;
- Goal:

$$\underbrace{\sum_{i=1}^{N} \nabla f(\boldsymbol{x}^{(i)}) = 0,}_{\text{optimality}} \qquad \underbrace{\boldsymbol{x}^{(i)} = \sum_{j=1}^{N} w_{ij} \boldsymbol{x}^{(j)}}_{\text{consensus}}.$$

Penalized formulation ($\sigma > 0$)

$$\operatorname*{minimize}_{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(N)} \in \mathbb{R}^n} \sum_{i=1}^{N} f_i(\boldsymbol{x}^{(i)}) + \sigma \left(\sum_{i=1}^{N} \|\boldsymbol{x}^{(i)}\|^2 - \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} [\boldsymbol{x}^{(i)}]^{\mathrm{T}} \boldsymbol{x}^{(j)} \right).$$

Solving consensus problems

$$\underset{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(N)}\in\mathbb{R}^n}{\text{minimize}} \sum_{i=1}^{N} f_i(\mathbf{x}^{(i)}) + \sigma\left(\sum_{i=1}^{N}\|\mathbf{x}^{(i)}\|^2 - \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}[\mathbf{x}^{(i)}]^{\mathrm{T}}\mathbf{x}^{(j)}\right).$$

With derivatives

- Dual methods (ADMM, etc);
- Decentralized gradient descent:

$$\forall i = 1...N, \quad \mathbf{x}_{k+1}^{(i)} = \sum_{j=1}^{N} w_{ij}\mathbf{x}_k^{(j)} - \alpha_k \nabla f_i(\mathbf{x}_k^{(i)}).$$

Existing derivative-free techniques

- Approximate each gradient via finite differences.
- Randomized using STP-like approaches.

Bringing in direct search (with E. Bergou, Y. Diouane, V. Kungurtsev)

$$\operatorname*{minimize}_{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(N)}\in\mathbb{R}^n} \sum_{i=1}^{N} f_i(\mathbf{x}^{(i)}) + \sigma\left(\sum_{i=1}^{N}\|\mathbf{x}^{(i)}\|^2 - \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}[\mathbf{x}^{(i)}]^{\mathrm{T}}\mathbf{x}^{(j)}\right).$$

Approach

- Define

$$\mathcal{L}_i(\mathbf{x}^{(i)}) = f_i(\mathbf{x}^{(i)}) - \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}[\mathbf{x}^{(i)}]^{\mathrm{T}}\mathbf{x}^{(j)}$$

Bringing in direct search (with E. Bergou, Y. Diouane, V. Kungurtsev)

$$\underset{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(N)}\in\mathbb{R}^n}{\text{minimize}} \sum_{i=1}^{N} f_i(\mathbf{x}^{(i)}) + \sigma\left(\sum_{i=1}^{N}\|\mathbf{x}^{(i)}\|^2 - \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}[\mathbf{x}^{(i)}]^{\mathrm{T}}\mathbf{x}^{(j)}\right).$$

Approach

- Define

$$\mathcal{L}_i(\mathbf{x}^{(i)}) = f_i(\mathbf{x}^{(i)}) - \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}[\mathbf{x}^{(i)}]^{\mathrm{T}}\mathbf{x}^{(j)}$$

- Agent i runs a direct search method.

Bringing in direct search (with E. Bergou, Y. Diouane, V. Kungurtsev)

$$\underset{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(N)}\in\mathbb{R}^n}{\text{minimize}} \sum_{i=1}^{N} f_i(\mathbf{x}^{(i)}) + \sigma\left(\sum_{i=1}^{N}\|\mathbf{x}^{(i)}\|^2 - \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}[\mathbf{x}^{(i)}]^{\mathrm{T}}\mathbf{x}^{(j)}\right).$$

Approach

- Define

$$\mathcal{L}_i(\mathbf{x}^{(i)}) = f_i(\mathbf{x}^{(i)}) - \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}[\mathbf{x}^{(i)}]^{\mathrm{T}}\mathbf{x}^{(j)}$$

- Agent i runs a direct search method.
- $\mathcal{L}_i$ changes over iterations $\Rightarrow$ Forces decrease in the stepsize.

Algorithm

Inputs: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\boldsymbol{x}_0^{(i)} = \boldsymbol{x}_0 \ \forall i$, $\{\alpha_k\}_k \searrow 0$, $t > 0$.

Iteration k, **Agent** i: Given $\left(\boldsymbol{x}_k^{(i)}, \alpha_k\right)$,

- Choose a set $\mathcal{D}_k^{(i)} \subset \mathbb{R}^n$ of m unit vectors.
- If $\exists \ \boldsymbol{d}_k^{(i)} \in \mathcal{D}_k^{(i)}$ such that

$$f(\boldsymbol{x}_k^{(i)} + \alpha_k \, \boldsymbol{d}_k) < f(\boldsymbol{x}_k^{(i)}) - \alpha_k^{1+t},$$

  set $\boldsymbol{x}_{k+1}^{(i)} := \boldsymbol{x}_k^{(i)} + \alpha_k \boldsymbol{d}_k^{(i)}$.
- Otherwise, set $\boldsymbol{x}_{k+1}^{(i)} := \boldsymbol{x}_k^{(i)}$.

First results: Convergence

$$\text{minimize}_{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(N)}\in\mathbb{R}^n} \sum_{i=1}^N f_i(\mathbf{x}^{(i)}) + \sigma \left(\sum_{i=1}^N \|\mathbf{x}^{(i)}\|^2 - \sum_{i=1}^N \sum_{j=1}^N w_{ij} [\mathbf{x}^{(i)}]^{\mathrm{T}} \mathbf{x}^{(j)} \right)$$

$$\forall i, \quad \mathcal{L}_i(\mathbf{x}^{(i)}) = f_i(\mathbf{x}^{(i)}) - \sum_{i=1}^N \sum_{j=1}^N w_{ij} [\mathbf{x}^{(i)}]^{\mathrm{T}} \mathbf{x}^{(j)}$$

Theorem (Bergou, Diouane, Kungurstev, R.)

Suppose that every agent runs direct-search iterations based on

- $\mathcal{D}_k^{(i)}$ is κ-descent with unit vectors;
- $\alpha_k = \frac{\alpha_0}{(1+k)^u}$, $u \in (1/2, 1)$;
- decrease in α_k^{1+t} with $u(1+t) < 1$.

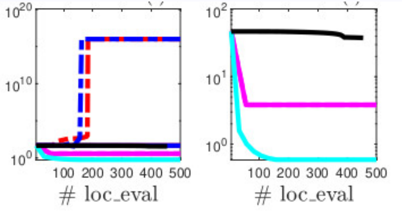
Then,

$$\liminf_{k\to\infty} \sum_{i=1}^N \|\nabla \mathcal{L}_i(\mathbf{x}^{(i)})\| = 0.$$

A toy example

$$\text{minimize}_{x^{(1)},\dots,x^{(N)}\in\mathbb{R}} \sum_{i=1}^N f_i(x^{(i)}) + \sigma\left(\sum_{i=1}^N \|x^{(i)}\|^2 - \sum_{i=1}^N \sum_{j=1}^N w_{ij}[x^{(i)}]^\mathrm{T} \mathbf{x}^{(j)}\right)$$

$$\forall i, \quad f_i(x^{(i)}) = \frac{a_i}{1+\exp(-x_i)} + b_i \log(1+x_i^2).$$



Objective vs Number of calls to $f_i(\cdot)$.

- Blue/Red: Finite-difference techniques;
- Black: Standard direct-search for all nodes;
- Cyan: Separable minimization;
- Magenta: New method.

Summary

DFO and dimension dependence

- A revised probabilistic analysis that allows for dimensionality reduction;
- Complexity results suggest a fundamental limit $\mathcal{O}(n)$;
- One dimensional variants pretty interesting!

Direct search based on probabilistic descent in reduced spaces. L. Roberts and C. W. Royer **(paper/Python toolbox coming soon!)**.

Summary

DFO and dimension dependence

- A revised probabilistic analysis that allows for dimensionality reduction;
- Complexity results suggest a fundamental limit $\mathcal{O}(n)$;
- One dimensional variants pretty interesting!

Direct search based on probabilistic descent in reduced spaces. L. Roberts and C. W. Royer **(paper/Python toolbox coming soon!)**.

Beyond centralized problems

- Typical zeroth-order approach: finite differences;
- Direct-search schemes may also be applicable.
- **Challenge:** Reason about a changing objective.

Decentralized direct search E. Bergou, Y. Diouane, V. Kungurtsev and C. W. Royer, in preparation.

What's next

Stochastic function values

- If sufficiently accurate in probability, things work out!
- Analysis of course (a lot) more technical.
- **Challenge:** Improve accuracy requirements.

What's next

Stochastic function values

- If sufficiently accurate in probability, things work out!
- Analysis of course (a lot) more technical.
- **Challenge:** Improve accuracy requirements.

What's next

Stochastic function values

- If sufficiently accurate in probability, things work out!
- Analysis of course (a lot) more technical.
- **Challenge:** Improve accuracy requirements.

Thank you for your attention!

`clement.royer@dauphine.psl.eu`