# Natural Evolution Strategies for Direct Search

PGMO-COPI 2014
Recent Advances on Continuous Randomized black-box optimization

Thursday October 30, 2014

**Tobias Glasmachers**
Institut für Neuroinformatik
Ruhr-Universität Bochum, Germany

# Introduction

# Introduction

"Function value free" black-box optimization:

# Introduction

"Function value free" black-box optimization:

1. Minimize $f : \mathbb{R}^d \to \mathbb{R}$.

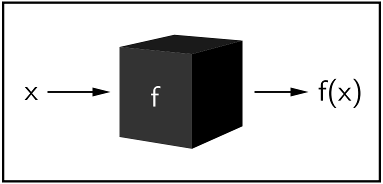
# Introduction

"Function value free" black-box optimization:

1. Minimize $f : \mathbb{R}^d \to \mathbb{R}$.
2. 0-th order (direct search) setting.

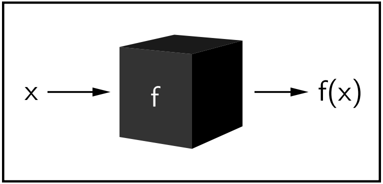
"Function value free" black-box optimization:

1. Minimize $f : \mathbb{R}^d \to \mathbb{R}$.
2. 0-th order (direct search) setting.
3. Black-box model



"Function value free" black-box optimization:

1. Minimize $f : \mathbb{R}^d \to \mathbb{R}$.
2. 0-th order (direct search) setting.
3. Black-box model




4. Don't ever rely on function values, only comparisons $f(x) < f(x')$, e.g., for ranking.

# Introduction

Algorithm operations:

# Introduction

Algorithm operations:

1. generate solution $x \in \mathbb{R}^d$ – most of the logic

# Introduction

Algorithm operations:

1. generate solution $x \in \mathbb{R}^d$ – most of the logic
2. evaluate $f(x)$ – black box, most of the computation time

# Introduction

Algorithm operations:

1. generate solution $x \in \mathbb{R}^d$ – most of the logic
2. evaluate $f(x)$ – black box, most of the computation time
3. compare (rank) against other solutions: $f(x) < f(x')$?

# Introduction: Challenges

# Introduction: Challenges

Facing a black-box objective, anything can happen.
We'd better prepare for the following **challenges**:

Facing a black-box objective, anything can happen.
We'd better prepare for the following **challenges**:

- non-smooth or even discontinuous objective

# Introduction: Challenges

Facing a black-box objective, anything can happen.
We'd better prepare for the following **challenges**:

- non-smooth or even discontinuous objective
- multi-modality

# Introduction: Challenges

Facing a black-box objective, anything can happen.
We'd better prepare for the following **challenges**:

- non-smooth or even discontinuous objective
- multi-modality
- observations of objective values perturbed by noise

# Introduction: Challenges

Facing a black-box objective, anything can happen.
We'd better prepare for the following **challenges**:

- non-smooth or even discontinuous objective
- multi-modality
- observations of objective values perturbed by noise
- high dimensionality, e.g., $d \gg 1000$

# Introduction: Challenges

Facing a black-box objective, anything can happen.
We'd better prepare for the following **challenges**:

- non-smooth or even discontinuous objective
- multi-modality
- observations of objective values perturbed by noise
- high dimensionality, e.g., $d \gg 1000$
- black-box constraints (possibly non-smooth, ...)

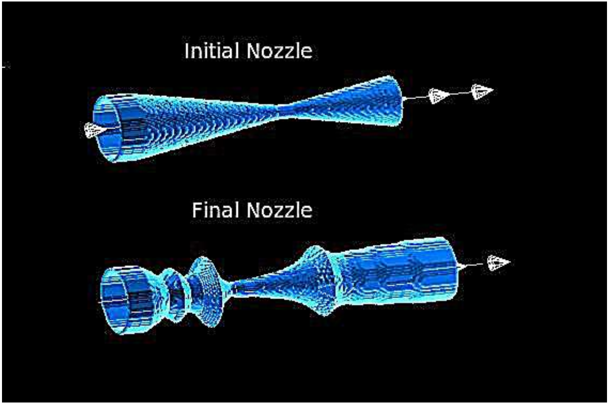
# Introduction: Application Examples

# Introduction: Application Examples

1. Tuning of parameters of expensive simulations, e.g., in engineering.

# Introduction: Application Examples

1. Tuning of parameters of expensive simulations, e.g., in engineering.

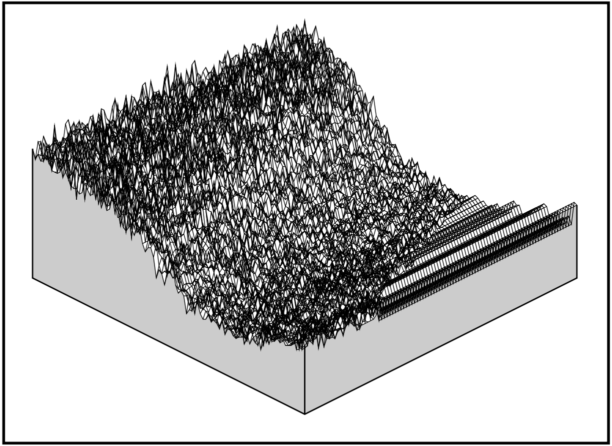


# Introduction: Application Examples

1. Tuning of parameters of expensive simulations, e.g., in engineering.
2. Tuning of parameters of machine learning algorithms.

# Introduction: Application Examples

1. Tuning of parameters of expensive simulations, e.g., in engineering.
2. Tuning of parameters of machine learning algorithms.



# Evolution Strategies

**input** $m \in \mathbb{R}^d$, $\sigma > 0$, $C (= \mathrm{I}) \in \mathbb{R}^{d \times d}$
**loop**
  sample "offspring" $x_1, \dots, x_\lambda \sim \mathcal{N}(m, \sigma^2 C)$
  evaluate $f(x_1), \dots, f(x_\lambda)$
  select new "population" of size $\mu$ (e.g., best offspring)
  update mean vector $m$
  update global step size $\sigma$
  update covariance matrix $C$
**until** stopping criterion met

# Evolution Strategies

**input** $m \in \mathbb{R}^d$, $\sigma > 0$, $C (= \mathrm{I}) \in \mathbb{R}^{d \times d}$
**loop**
  sample "offspring" $x_1, \dots, x_\lambda \sim \mathcal{N}(m, \sigma^2 C)$
  evaluate $f(x_1), \dots, f(x_\lambda)$
  select new "population" of size $\mu$ (e.g., best offspring)
  update mean vector $m$
  update global step size $\sigma$
  update covariance matrix $C$
**until** stopping criterion met

- randomized
- population-based
- rank-based (function-value free)
- step size control
- metric learning (covariance matrix adaptation, CMA)

# Evolution Strategies: Design Principles

- Why rely on a population?

# Evolution Strategies: Design Principles

- Why rely on a population?
  Not necessary, "hillclimber" (1+1)-ES also works, but populations are more robust for noisy and multi-modal problems.

# Evolution Strategies: Design Principles

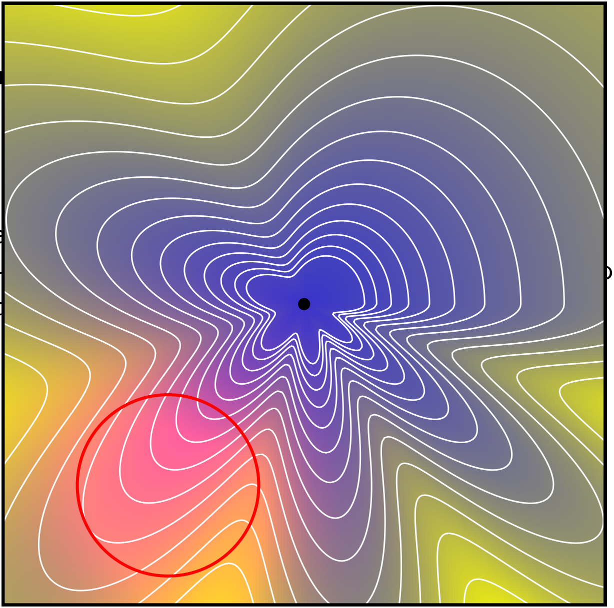
- Why rely on a population?
  Not necessary, "hillclimber" (1+1)-ES also works, but populations are more robust for noisy and multi-modal problems.
- Why online adaptation of the step size $\sigma$?

# Evolution Strategies: Design Principles

- Why rely on a population?
  Not necessary, "hillclimber" (1+1)-ES also works, but populations are more robust for noisy and multi-modal problems.
- Why online adaptation of the step size $\sigma$?
  Scale-invariant algorithm $\Rightarrow$ linear convergence on scale-invariant problems $\Rightarrow$ locally linear convergence on $\mathcal{C}^2$ functions.

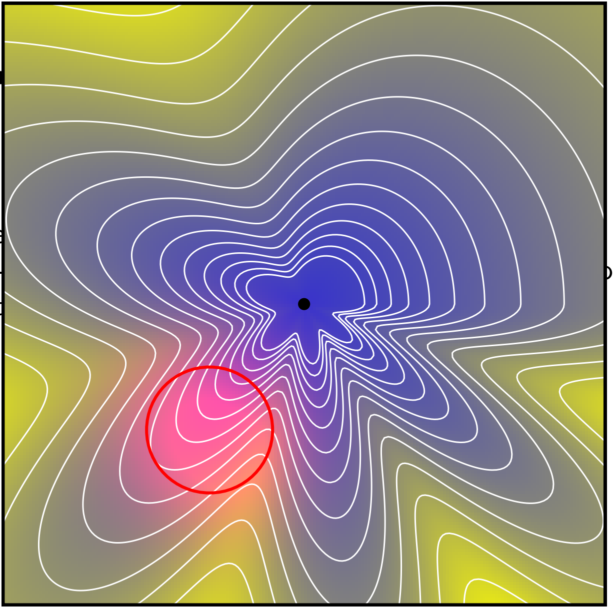
# Evolution Strategies: Design Principles

- Why
  Not
  popu
  probl
- Why
  Scale
  scale- … on $\mathcal{C}^2$
  funct



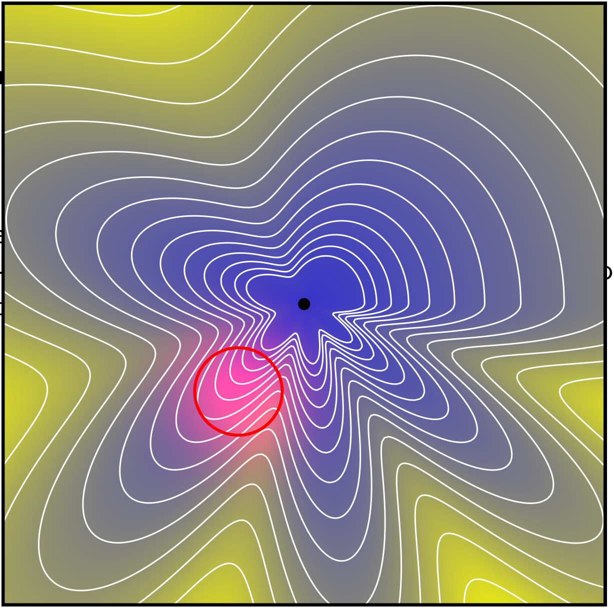
# Evolution Strategies: Design Principles

- Why
  Not
  popu
  probl
- Why
  Scale
  scale- on $\mathcal{C}^2$
  funct



# Evolution Strategies: Design Principles

- Why
  Not
  popu
  probl
- Why
  Scale
  scale- on $\mathcal{C}^2$
  funct



# Evolution Strategies: Design Principles

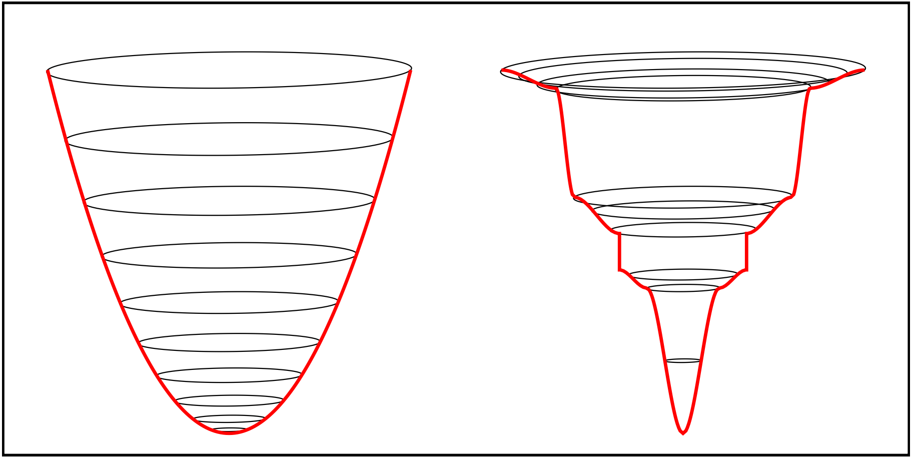
- Why rely on a population?
  Not necessary, "hillclimber" (1+1)-ES also works, but populations are more robust for noisy and multi-modal problems.
- Why online adaptation of the step size $\sigma$?
  Scale-invariant algorithm $\Rightarrow$ linear convergence on scale-invariant problems $\Rightarrow$ locally linear convergence on $\mathcal{C}^2$ functions.
- Why discard function values and keep only ranks?

# Evolution Strategies: Design Principles

- Why rely on a population?
  Not necessary, "hillclimber" (1+1)-ES also works, but populations are more robust for noisy and multi-modal problems.
- Why online adaptation of the step size $\sigma$?
  Scale-invariant algorithm $\Rightarrow$ linear convergence on scale-invariant problems $\Rightarrow$ locally linear convergence on $\mathcal{C}^2$ functions.
- Why discard function values and keep only ranks?
  Invariance to strictly monotonic (rank-preserving) transformations of objective values.

# Evolution Strategies: Design Principles

- Why rely on a population?



# Evolution Strategies: Design Principles

- Why rely on a population?
  Not necessary, "hillclimber" (1+1)-ES also works, but populations are more robust for noisy and multi-modal problems.

- Why online adaptation of the step size $\sigma$?
  Scale-invariant algorithm $\Rightarrow$ linear convergence on scale-invariant problems $\Rightarrow$ locally linear convergence on $\mathcal{C}^2$ functions.

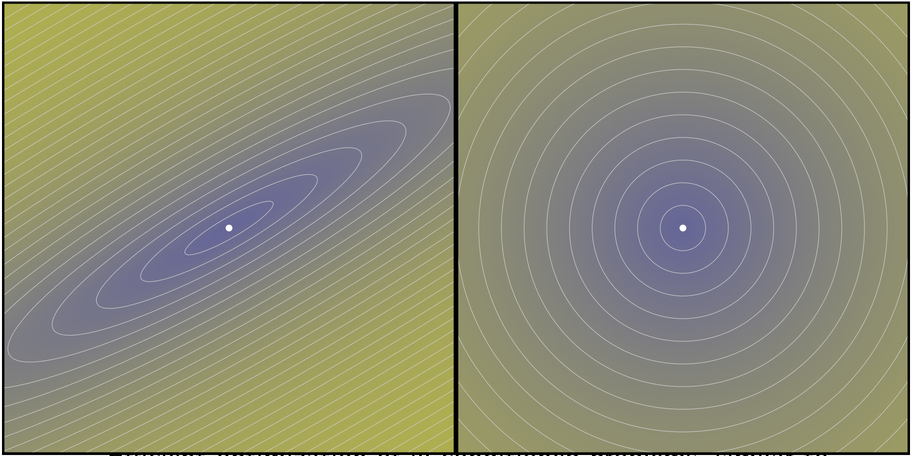
- Why discard function values and keep only ranks?
  Invariance to strictly monotonic (rank-preserving) transformations of objective values.

- Why CMA?

# Evolution Strategies: Design Principles

- Why rely on a population?
  Not necessary, "hillclimber" (1+1)-ES also works, but populations are more robust for noisy and multi-modal problems.
- Why online adaptation of the step size $\sigma$?
  Scale-invariant algorithm $\Rightarrow$ linear convergence on scale-invariant problems $\Rightarrow$ locally linear convergence on $\mathcal{C}^2$ functions.
- Why discard function values and keep only ranks?
  Invariance to strictly monotonic (rank-preserving) transformations of objective values.
- Why CMA?
  Efficient optimization of ill-conditioned problems, similar to (quasi) Newton methods.

- Why rely on a population?



Efficient optimization of ill-conditioned problems, similar to (quasi) Newton methods.

# This Talk

**Given samples and their ranks,**

**how to update the distribution parameters?**

# Information Geometric Perspective

# Information Geometric Perspective

- Change of perspective: algorithm state
  **population** $x_1, \dots, x_\mu \rightarrow$ **distribution** $\mathcal{N}(m, C)$.

# Information Geometric Perspective

- Change of perspective: algorithm state **population** $x_1, \dots, x_\mu \to$ **distribution** $\mathcal{N}(m, C)$.
- Parameters $(m, C) = \theta \in \Theta$, distribution $P_\theta$, pdf $p_\theta$.

# Information Geometric Perspective

- Change of perspective: algorithm state
  **population** $x_1, \dots, x_\mu \to$ **distribution** $\mathcal{N}(m, C)$.
- Parameters $(m, C) = \theta \in \Theta$, distribution $P_\theta$, pdf $p_\theta$.
- For multi-variate Gaussians:

$$\Theta = \mathbb{R}^d \times \mathcal{P}_d$$

$$\mathcal{P}_d = \left\{ M \in \mathbb{R}^{d \times d} \,\middle|\, M = M^T, M \text{ pos. def.} \right\}$$

# Information Geometric Perspective

- Change of perspective: algorithm state **population** $x_1, \dots, x_\mu \to$ **distribution** $\mathcal{N}(m, C)$.
- Parameters $(m, C) = \theta \in \Theta$, distribution $P_\theta$, pdf $p_\theta$.
- For multi-variate Gaussians:

$$\Theta = \mathbb{R}^d \times \mathcal{P}_d$$

$$\mathcal{P}_d = \left\{ M \in \mathbb{R}^{d \times d} \,\middle|\, M = M^T, M \text{ pos. def.} \right\}$$

- **Statistical manifold** of distributions

$$\left\{ P_\theta \,\middle|\, \theta \in \Theta \right\} \cong \Theta \ .$$

- Change of perspective: algorithm state **population** $x_1, \ldots, x_\mu \to$ **distribution** $\mathcal{N}(m, C)$.
- Parameters $(m, C) = \theta \in \Theta$, distribution $P_\theta$, pdf $p_\theta$.
- For multi-variate Gaussians:

$$\Theta = \mathbb{R}^d \times \mathcal{P}_d$$
$$\mathcal{P}_d = \left\{ M \in \mathbb{R}^{d \times d} \,\middle|\, M = M^T, M \text{ pos. def.} \right\}$$

- **Statistical manifold** of distributions

$$\left\{ P_\theta \,\middle|\, \theta \in \Theta \right\} \cong \Theta \ .$$

- Equipped with intrinsic (Riemannian) geometry; metric tensor given by Fisher information matrix $\mathcal{I}(\theta)$.

# Information Geometric Perspective

- Goal: optimization of $\theta \in \Theta$ instead of $x \in \mathbb{R}^d$.

# Information Geometric Perspective

- Goal: optimization of $\theta \in \Theta$ instead of $x \in \mathbb{R}^d$.
- Lift objective function $f : \mathbb{R}^d \to \mathbb{R}$ to objective $W_f : \Theta \to \mathbb{R}$.

# Information Geometric Perspective

- Goal: optimization of $\theta \in \Theta$ instead of $x \in \mathbb{R}^d$.
- Lift objective function $f : \mathbb{R}^d \to \mathbb{R}$ to objective $W_f : \Theta \to \mathbb{R}$.
- Simplest choice:

$$W_f(\theta) = \mathbb{E}_{x \sim P_\theta}\Big[f(x)\Big]$$

# Information Geometric Perspective

- Goal: optimization of $\theta \in \Theta$ instead of $x \in \mathbb{R}^d$.
- Lift objective function $f : \mathbb{R}^d \to \mathbb{R}$ to objective $W_f : \Theta \to \mathbb{R}$.
- Simplest choice:

$$W_f(\theta) = \mathbb{E}_{x \sim P_\theta}\Big[f(x)\Big]$$

- More flexible choice:

$$W_f(\theta) = \mathbb{E}_{x \sim P_\theta}\Big[w\big(f(x)\big)\Big]$$

  with monotonic weight function $w : \mathbb{R} \to \mathbb{R}$.

# Information Geometric Perspective

- Expectation operator “adds one degree of smoothness”.

# Information Geometric Perspective

- Expectation operator "adds one degree of smoothness".
- Hence under weak assumptions $W_f(\theta)$ can be optimized with **gradient descent (GD)**:

$$\theta \leftarrow \theta - \eta \cdot \nabla_\theta W_f(\theta)$$

# Information Geometric Perspective

- Expectation operator "adds one degree of smoothness".
- Hence under weak assumptions $W_f(\theta)$ can be optimized with **gradient descent (GD)**:

$$\theta \leftarrow \theta - \eta \cdot \nabla_\theta W_f(\theta)$$

- "log-likelohood trick":

$$\begin{aligned} \nabla_\theta W_f(\theta) &= \nabla_\theta \mathbb{E}_{x \sim P_\theta}\Big[w\big(f(x)\big)\Big] \\ &= \mathbb{E}_{x \sim P_\theta}\Big[\nabla_\theta \log\big(p_\theta(x)\big) \cdot w\big(f(x)\big)\Big] \end{aligned}$$

- Problem: this does not work well. Why?

# Information Geometric Perspective

- Problem: this does not work well. Why?
- We have to replace the "plain" gradient on (Euclidean) parameter space $\Theta$ with the **natural gradient** respecting the intrinsic geometry of the statistical manifold:

$$\tilde{\nabla}_\theta W_f(\theta) = \Big(\mathcal{I}(\theta)\Big)^{-1} \cdot \nabla_\theta W_f(\theta)$$

# Information Geometric Perspective

- Problem: this does not work well. Why?
- We have to replace the "plain" gradient on (Euclidean) parameter space $\Theta$ with the **natural gradient** respecting the intrinsic geometry of the statistical manifold:

$$\tilde{\nabla}_\theta W_f(\theta) = \Big(\mathcal{I}(\theta)\Big)^{-1} \cdot \nabla_\theta W_f(\theta)$$

- The natural gradient is invariant under changes of the parameterization $\theta \mapsto P_\theta$.

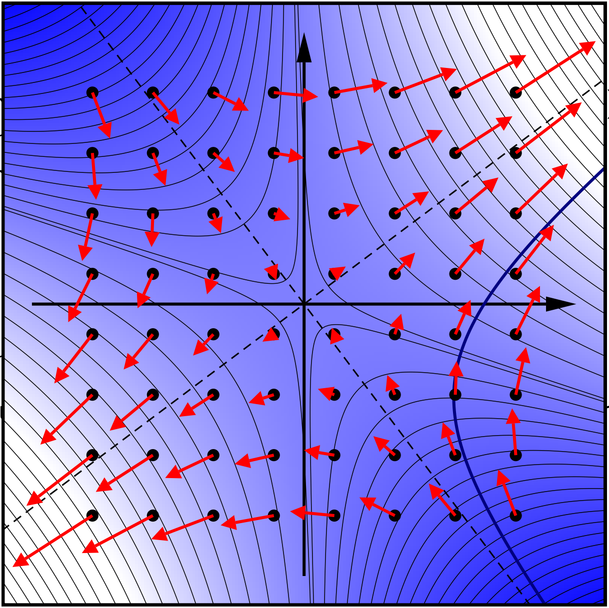
# Information Geometric Perspective

- Problem: this does not work well. Why?
- We have to replace the "plain" gradient on (Euclidean) parameter space $\Theta$ with the **natural gradient** respecting the intrinsic geometry of the statistical manifold:

$$\tilde{\nabla}_\theta W_f(\theta) = \Big(\mathcal{I}(\theta)\Big)^{-1} \cdot \nabla_\theta W_f(\theta)$$

- The natural gradient is invariant under changes of the parameterization $\theta \mapsto P_\theta$.
- (Natural) gradient vector field $\Theta \to T\Theta$ defines (natural) gradient flow $\phi^t(\theta)$ tangential to $\tilde{\nabla}_\theta W_f(\theta)$.

# Information Geometric Perspective



- Prob
- We h )
  paran ng the
  intrin
- The
  paran
- (Natu al)
  gradi

# Information Geometric Perspective

- Problem: this does not work well. Why?
- We have to replace the "plain" gradient on (Euclidean) parameter space $\Theta$ with the **natural gradient** respecting the intrinsic geometry of the statistical manifold:

$$\tilde{\nabla}_\theta W_f(\theta) = \Big(\mathcal{I}(\theta)\Big)^{-1} \cdot \nabla_\theta W_f(\theta)$$

- The natural gradient is invariant under changes of the parameterization $\theta \mapsto P_\theta$.
- (Natural) gradient vector field $\Theta \to T\Theta$ defines (natural) gradient flow $\phi^t(\theta)$ tangential to $\tilde{\nabla}_\theta W_f(\theta)$.
- Optimization: follow inverse flow curves $t \mapsto \phi^{-t}(\theta)$ from $\theta$ into (local) minimum of $W_f$.

# Information Geometric Perspective

- Black-box setting: expectation

$$\mathbb{E}_{x\sim P_\theta}\left[\nabla_\theta \log\big(p_\theta(x)\big)\cdot w\big(f(x)\big)\right]$$

is intractable.

# Information Geometric Perspective

- Black-box setting: expectation

$$\mathbb{E}_{x \sim P_\theta} \Big[ \nabla_\theta \log \big( p_\theta(x) \big) \cdot w\big( f(x) \big) \Big]$$

  is intractable.

- **Monte Carlo (MC)** approximation

$$\nabla_\theta W_f(\theta) \approx G(\theta) = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \Big[ \nabla_\theta \log \big( p_\theta(x_i) \big) \cdot w\big( f(x_i) \big) \Big]$$

  for $x_1, \dots, x_\lambda \sim P_\theta$.

# Information Geometric Perspective

- Black-box setting: expectation

$$\mathbb{E}_{x \sim P_\theta}\left[\nabla_\theta \log\big(p_\theta(x)\big) \cdot w\big(f(x)\big)\right]$$

  is intractable.

- **Monte Carlo (MC)** approximation

$$\nabla_\theta W_f(\theta) \approx G(\theta) = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \left[\nabla_\theta \log\big(p_\theta(x_i)\big) \cdot w\big(f(x_i)\big)\right]$$

  for $x_1, \dots, x_\lambda \sim P_\theta$.

- The estimate $\mathbb{E}[G(\theta)] = \nabla_\theta W_f(\theta)$ is consistent, hence following $G(\theta)$ amounts to **stochastic gradient descent**.

# Information Geometric Perspective

- Black-box setting: expectation

$$\mathbb{E}_{x \sim P_\theta} \left[ \nabla_\theta \log \big( p_\theta(x) \big) \cdot w\big( f(x) \big) \right]$$

is intractable.

- **Monte Carlo (MC)** approximation

$$\nabla_\theta W_f(\theta) \approx G(\theta) = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \left[ \nabla_\theta \log \big( p_\theta(x_i) \big) \cdot w\big( f(x_i) \big) \right]$$

for $x_1, \dots, x_\lambda \sim P_\theta$.

- The estimate $\mathbb{E}[G(\theta)] = \nabla_\theta W_f(\theta)$ is consistent, hence following $G(\theta)$ amounts to **stochastic gradient descent**.
- Yields natural gradient MC approximation

$$\tilde{G}(\theta) = \big( \mathcal{I}(\theta) \big)^{-1} \cdot G(\theta) \ .$$

# Information Geometric Perspective

- Stochastic Natural Gradient Descent (SNGD) update rule:

$$\theta \leftarrow \theta - \eta \cdot \tilde{G}(\theta)$$

# Information Geometric Perspective

- Stochastic Natural Gradient Descent (SNGD) update rule:

$$\theta \leftarrow \theta - \eta \cdot \tilde{G}(\theta)$$

- SNGD is a two-fold approximation of the gradient flow:
  - discretized time: Euler steps,
  - randomized gradient: MC sampling.

# Information Geometric Perspective

- Stochastic Natural Gradient Descent (SNGD) update rule:

$$\theta \leftarrow \theta - \eta \cdot \tilde{G}(\theta)$$

- SNGD is a two-fold approximation of the gradient flow:
  - discretized time: Euler steps,
  - randomized gradient: MC sampling.
- Natural gradient flow is invariant under choice of distribution parameters $\theta \mapsto P_\theta$.

# Information Geometric Perspective

- Stochastic Natural Gradient Descent (SNGD) update rule:

$$\theta \leftarrow \theta - \eta \cdot \tilde{G}(\theta)$$

- SNGD is a two-fold approximation of the gradient flow:
  - discretized time: Euler steps,
  - randomized gradient: MC sampling.
- Natural gradient flow is invariant under choice of distribution parameters $\theta \mapsto P_\theta$.
- SNGD algorithm invariant in first order approximation due to Euler steps.

# Information Geometric Perspective

- Stochastic Natural Gradient Descent (SNGD) update rule:

$$\theta \leftarrow \theta - \eta \cdot \tilde{G}(\theta)$$

- SNGD is a two-fold approximation of the gradient flow:
  - discretized time: Euler steps,
  - randomized gradient: MC sampling.
- Natural gradient flow is invariant under choice of distribution parameters $\theta \mapsto P_\theta$.
- SNGD algorithm invariant in first order approximation due to Euler steps.

Ollivier et al. Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles. *arXiv:1106.3708*, 2011.

- Natural (gradient) Evolution Strategies (NES) closely follow this scheme. They were the first ES derived this way.

# Natural (Gradient) Evolution Strategies

- Natural (gradient) Evolution Strategies (NES) closely follow this scheme. They were the first ES derived this way.
- NES approach: optimization of expected objective value with Gaussian distributions.

# Natural (Gradient) Evolution Strategies

- Natural (gradient) Evolution Strategies (NES) closely follow this scheme. They were the first ES derived this way.
- NES approach: optimization of expected objective value with Gaussian distributions.
- Offspring $x_1, \dots, x_\lambda$ act as MC sample for the estimation of $G(\theta)$.

# Natural (Gradient) Evolution Strategies

- Natural (gradient) Evolution Strategies (NES) closely follow this scheme. They were the first ES derived this way.
- NES approach: optimization of expected objective value with Gaussian distributions.
- Offspring $x_1, \dots, x_\lambda$ act as MC sample for the estimation of $G(\theta)$.
- Closed form Fisher tensor for $\mathcal{N}(m, C)$:

$$\mathcal{I}_{i,j}(\theta) = \frac{\partial m^T}{\partial \theta_i} C^{-1} \frac{\partial m}{\partial \theta_j} + \frac{1}{2} \operatorname{tr}\left( C^{-1} \frac{\partial C}{\partial \theta_i} C^{-1} \frac{\partial C}{\partial \theta_j} \right)$$

## NES algorithm

**input** $\theta \in \Theta$, $\lambda \in \mathbb{N}$, $\eta > 0$
**loop**
  sample $x_1, \dots, x_\lambda \sim P_\theta$
  evaluate $f(x_1), \dots, f(x_\lambda)$
  $G(\theta) \leftarrow \frac{1}{\lambda} \sum_{i=1}^{\lambda} \nabla_\theta \log \big( p_\theta(x_i) \big) \cdot f(x_i)$
  $\tilde{G}(\theta) \leftarrow \big( \mathcal{I}(\theta) \big)^{-1} \cdot G(\theta)$
  $\theta \leftarrow \theta - \eta \cdot \tilde{G}(\theta)$
**until** stopping criterion met

Wierstra et al. Natural Evolution Strategies. *CEC*, 2008 and *JMLR*, 2014.

# Natural (Gradient) Evolution Strategies

- NES works better when replacing $f(x_i)$ in

$$G(\theta) = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \left[ \nabla_\theta \log \left( p_\theta(x_i) \right) \cdot f(x_i) \right]$$

with rank-based utility values $w_1, \dots, w_\lambda$.

# Natural (Gradient) Evolution Strategies

- NES works better when replacing $f(x_i)$ in

$$G(\theta) = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \left[ \nabla_\theta \log \left( p_\theta(x_i) \right) \cdot f(x_i) \right]$$

  with rank-based utility values $w_1, \dots, w_\lambda$.

- Sample ranks are $f$-quantile estimators, hence utility values can be represented as $w(f(x_i))$ with special weight function $w = w_\theta$ based on $f$-quantiles under **current** distribution $P_\theta$.

# Natural (Gradient) Evolution Strategies

- NES works better when replacing $f(x_i)$ in

$$G(\theta) = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \left[ \nabla_\theta \log \left( p_\theta(x_i) \right) \cdot f(x_i) \right]$$

  with rank-based utility values $w_1, \dots, w_\lambda$.

- Sample ranks are $f$-quantile estimators, hence utility values can be represented as $w(f(x_i))$ with special weight function $w = w_\theta$ based on $f$-quantiles under **current** distribution $P_\theta$.
- This turns NES into a function value free algorithm.

# Natural (Gradient) Evolution Strategies

- NES works better when replacing $f(x_i)$ in

$$G(\theta) = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \left[ \nabla_\theta \log \left( p_\theta(x_i) \right) \cdot f(x_i) \right]$$

  with rank-based utility values $w_1, \dots, w_\lambda$.
- Sample ranks are $f$-quantile estimators, hence utility values can be represented as $w(f(x_i))$ with special weight function $w = w_\theta$ based on $f$-quantiles under **current** distribution $P_\theta$.
- This turns NES into a function value free algorithm.
- Benefits:
  - invariance under monotonic transformations of objective values
  - linear convergence on scale invariant problems.

## xNES (exponential NES) algorithm

**input** $(m, A) \in \Theta$, $\lambda \in \mathbb{N}$, $\eta_m, \eta_A > 0$
**loop**
  sample $z_1, \dots, z_\lambda \sim \mathcal{N}(0, \mathrm{I})$
  transform $x_k, \dots, x_\lambda \leftarrow A z_k + m$
  evaluate $f(x_1), \dots, f(x_\lambda)$
  $\tilde{G}_m(\theta) \leftarrow \frac{1}{\lambda} \sum_{i=1}^{\lambda} w(f(x_i)) \cdot z_i$
  $\tilde{G}_C(\theta) \leftarrow \frac{1}{\lambda} \sum_{i=1}^{\lambda} w(f(x_i)) \cdot \frac{1}{2}(z_i z_i^T - \mathrm{I})$
  $m \leftarrow m - \eta_m \cdot A \cdot \tilde{G}_m(\theta)$
  $A \leftarrow A \cdot \exp\left(-\eta_A \cdot \frac{1}{2} \tilde{G}_C(\theta)\right)$
**until** stopping criterion met

Glasmachers et al. Exponential Natural Evolution Strategies. *GECCO*, 2010.

# Natural (Gradient) Evolution Strategies

- The resulting algorithm is indeed an ES ($\mathbb{R}^d$ perspective), and at the same time an SNGD algorithm ($\Theta$ perspective).

# Natural (Gradient) Evolution Strategies

- The resulting algorithm is indeed an ES ($\mathbb{R}^d$ perspective), and at the same time an SNGD algorithm ($\Theta$ perspective).
- ES traditionally have three distinct and often very different mechanisms for
  - optimization: adaptation of $m$,
  - step size control: adaptation of $\sigma$,
  - metric learning: adaptation of $C$.

# Natural (Gradient) Evolution Strategies

- The resulting algorithm is indeed an ES ($\mathbb{R}^d$ perspective), and at the same time an SNGD algorithm ($\Theta$ perspective).
- ES traditionally have three distinct and often very different mechanisms for
  - optimization: adaptation of $m$,
  - step size control: adaptation of $\sigma$,
  - metric learning: adaptation of $C$.
- SNGD has only a single mechanism.

# Natural (Gradient) Evolution Strategies

- The resulting algorithm is indeed an ES ($\mathbb{R}^d$ perspective), and at the same time an SNGD algorithm ($\Theta$ perspective).
- ES traditionally have three distinct and often very different mechanisms for
    - optimization: adaptation of $m$,
    - step size control: adaptation of $\sigma$,
    - metric learning: adaptation of $C$.
- SNGD has only a single mechanism.
- Astonishing insight: all (most) parameters can be updated with a single mechanism.

# Exponential Natural Evolution Strategies

- Although derived completely differently, this algorithm turns out to be closely related to CMA-ES.

Akimoto et al. Bidirectional relation between CMA evolution strategies and natural evolution strategies. PPSN 2010.

# Exponential Natural Evolution Strategies

- Although derived completely differently, this algorithm turns out to be closely related to CMA-ES.

Akimoto et al. Bidirectional relation between CMA evolution strategies and natural evolution strategies. PPSN 2010.

- In particular, the update of $m$ is identical and the rank-$\mu$ update of CMA-ES coincides with a NES update.

# Exponential Natural Evolution Strategies

- Although derived completely differently, this algorithm turns out to be closely related to CMA-ES.

Akimoto et al. Bidirectional relation between CMA evolution strategies and natural evolution strategies. PPSN 2010.

- In particular, the update of $m$ is identical and the rank-$\mu$ update of CMA-ES coincides with a NES update.
- This is astonishing, it is surely not by coincidence.

# Exponential Natural Evolution Strategies

- Although derived completely differently, this algorithm turns out to be closely related to CMA-ES.

Akimoto et al. Bidirectional relation between CMA evolution strategies and natural evolution strategies. PPSN 2010.

- In particular, the update of $m$ is identical and the rank-$\mu$ update of CMA-ES coincides with a NES update.
- This is astonishing, it is surely not by coincidence.
- This is insightful: it means that CMA-ES is (essentially) a SNGD algorithm.

# Summary

# Summary

- Evolution Strategies (ES) are randomized direct search methods suitable for continuous black box optimization.

# Summary

- Evolution Strategies (ES) are randomized direct search methods suitable for continuous black box optimization.
- Here we have focused on a core question: how to update the distribution parameters in a principled way?

# Summary

- Evolution Strategies (ES) are randomized direct search methods suitable for continuous black box optimization.
- Here we have focused on a core question: how to update the distribution parameters in a principled way?
- The parameter space is equipped with the non-Euclidean information geometry of the corresponding statistical manifold of search distributions.

# Summary

- Evolution Strategies (ES) are randomized direct search methods suitable for continuous black box optimization.
- Here we have focused on a core question: how to update the distribution parameters in a principled way?
- The parameter space is equipped with the non-Euclidean information geometry of the corresponding statistical manifold of search distributions.
- SNGD on a stochastically relaxed problem results in a direct search algorithm: the Natural-gradient Evolution Strategy (NES) algorithm.

# Summary

- Evolution Strategies (ES) are randomized direct search methods suitable for continuous black box optimization.
- Here we have focused on a core question: how to update the distribution parameters in a principled way?
- The parameter space is equipped with the non-Euclidean information geometry of the corresponding statistical manifold of search distributions.
- SNGD on a stochastically relaxed problem results in a direct search algorithm: the Natural-gradient Evolution Strategy (NES) algorithm.
- The SNGD parameter update is by no means restricted to Gaussian distributions. It is a general construction template for update equations of continuous distribution parameters.

# Thank you!
# Questions?