Stochastic Methods for Continuous Optimization

Anne Auger et Dimo Brockhoff

Paris-Saclay Master - Master 2 Informatique - Parcours Apprentissage, Information et Contenu (AIC)

2016

Slides taken from Auger, Hansen, Gecco's tutorial
Problem Statement

Continuous Domain Search/Optimization

- Task: minimize an objective function (fitness function, loss function) in continuous domain

\[ f : \mathcal{X} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}, \quad x \mapsto f(x) \]

- Black Box scenario (direct search scenario)

  - gradients are not available or not useful
  - problem domain specific knowledge is used only within the black box, e.g. within an appropriate encoding

- Search costs: number of function evaluations
What Makes a Function Difficult to Solve?

Why stochastic search?

- non-linear, non-quadratic, non-convex
  on linear and quadratic functions much better
  search policies are available

- ruggedness
  non-smooth, discontinuous, multimodal, and/or
  noisy function

- dimensionality (size of search space)
  (considerably) larger than three

- non-separability
  dependencies between the objective variables

- ill-conditioning
Ruggedness

non-smooth, discontinuous, multimodal, and/or noisy

cut from a 5-D example, (easily) solvable with evolution strategies
Curse of Dimensionality

The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space.

Example: Consider placing 20 points equally spaced onto the interval $[0, 1]$. Now consider the 10-dimensional space $[0, 1]^{10}$. To get similar coverage in terms of distance between adjacent points requires $20^{10} \approx 10^{13}$ points. 20 points appear now as isolated points in a vast empty space.

Remark: distance measures break down in higher dimensionalities (the central limit theorem kicks in)

Consequence: a search policy that is valuable in small dimensions might be useless in moderate or large dimensional search spaces. Example: exhaustive search.
Curse of Dimensionality

The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space.

Example: Consider placing 20 points equally spaced onto the interval $[0, 1]$. Now consider the 10-dimensional space $[0, 1]^{10}$. To get similar coverage in terms of distance between adjacent points requires $20^{10} \approx 10^{13}$ points. 20 points appear now as isolated points in a vast empty space.

Remark: distance measures break down in higher dimensionalities (the central limit theorem kicks in)

Consequence: a search policy that is valuable in small dimensions might be useless in moderate or large dimensional search spaces. Example: exhaustive search.
Curse of Dimensionality

The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space.

Example: Consider placing 20 points equally spaced onto the interval $[0, 1]$. Now consider the 10-dimensional space $[0, 1]^{10}$. To get similar coverage in terms of distance between adjacent points requires $20^{10} \approx 10^{13}$ points. 20 points appear now as isolated points in a vast empty space.

Remark: *distance measures* break down in higher dimensionalities (the central limit theorem kicks in)

Consequence: a search policy that is valuable in small dimensions might be useless in moderate or large dimensional search spaces. Example: exhaustive search.
Curse of Dimensionality

The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space.

Example: Consider placing 20 points equally spaced onto the interval $[0, 1]$. Now consider the 10-dimensional space $[0, 1]^{10}$. To get similar coverage in terms of distance between adjacent points requires $20^{10} \approx 10^{13}$ points. 20 points appear now as isolated points in a vast empty space.

Remark: *distance measures* break down in higher dimensionalities (the central limit theorem kicks in)

Consequence: a *search policy* that is valuable in small dimensions might be useless in moderate or large dimensional search spaces.

Example: exhaustive search.
Separable Problems

**Definition (Separable Problem)**

A function $f$ is separable if

$$\arg\min_{(x_1, \ldots, x_n)} f(x_1, \ldots, x_n) = \left( \arg\min_{x_1} f(x_1, \ldots), \ldots, \arg\min_{x_n} f(\ldots, x_n) \right)$$

⇒ it follows that $f$ can be optimized in a sequence of $n$ independent 1-D optimization processes

**Example: Additively decomposable functions**

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n} f_i(x_i)$$

Rastrigin function
Non-Separable Problems

Building a non-separable problem from a separable one \(^{(1,2)}\)

Rotating the coordinate system

- \( f : x \mapsto f(x) \) separable
- \( f : x \mapsto f(Rx) \) non-separable

\( R \) rotation matrix

---


Ill-Conditioned Problems

Curvature of level sets

Consider the convex-quadratic function
\[ f(x) = \frac{1}{2} (x - x^*)^T H (x - x^*) = \frac{1}{2} \sum_i h_{i,i} (x_i - x_i^*)^2 + \frac{1}{2} \sum_{i \neq j} h_{i,j} (x_i - x_i^*) (x_j - x_j^*) \]

\( H \) is Hessian matrix of \( f \) and symmetric positive definite

Ill-conditioning means squeezed level sets (high curvature). Condition number equals nine here. Condition numbers up to \( 10^{10} \) are not unusual in real world problems.

If \( H \approx I \) (small condition number of \( H \)) first order information (e.g. the gradient) is sufficient. Otherwise second order information (estimation of \( H^{-1} \)) is necessary.
What Makes a Function Difficult to Solve?

...and what can be done

<table>
<thead>
<tr>
<th>The Problem</th>
<th>Possible Approaches</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimensionality</strong></td>
<td>exploiting the problem structure</td>
</tr>
<tr>
<td></td>
<td>separability, locality/neighborhood, encoding</td>
</tr>
<tr>
<td><strong>Ill-conditioning</strong></td>
<td>second order approach</td>
</tr>
<tr>
<td></td>
<td>changes the neighborhood metric</td>
</tr>
<tr>
<td><strong>Ruggedness</strong></td>
<td>non-local policy, large sampling width (step-size)</td>
</tr>
<tr>
<td></td>
<td>as large as possible while preserving a reasonable convergence speed</td>
</tr>
<tr>
<td></td>
<td>population-based method, stochastic, non-elitistic</td>
</tr>
<tr>
<td></td>
<td>recombination operator</td>
</tr>
<tr>
<td></td>
<td>serves as repair mechanism</td>
</tr>
<tr>
<td></td>
<td>restarts</td>
</tr>
</tbody>
</table>

...metaphors
Stochastic Search

A black box search template to minimize $f : \mathbb{R}^n \rightarrow \mathbb{R}$

Initialize distribution parameters $\theta$, set population size $\lambda \in \mathbb{N}$
While not terminate

1. Sample distribution $P (x | \theta) \rightarrow x_1, \ldots, x_\lambda \in \mathbb{R}^n$
2. Evaluate $x_1, \ldots, x_\lambda$ on $f$
3. Update parameters $\theta \leftarrow F_\theta (\theta, x_1, \ldots, x_\lambda, f(x_1), \ldots, f(x_\lambda))$

Everything depends on the definition of $P$ and $F_\theta$
deterministic algorithms are covered as well

In many Evolutionary Algorithms the distribution $P$ is implicitly defined
via operators on a population, in particular, selection, recombination
and mutation

Natural template for (incremental) Estimation of Distribution Algorithms
Evolution Strategies

New search points are sampled normally distributed

\[ x_i \sim m + \sigma \mathcal{N}_i(0, C) \quad \text{for } i = 1, \ldots, \lambda \]

as perturbations of \( m \), where \( x_i, m \in \mathbb{R}^n \), \( \sigma \in \mathbb{R}_+ \), \( C \in \mathbb{R}^{n \times n} \)

where

- the mean vector \( m \in \mathbb{R}^n \) represents the favorite solution
- the so-called step-size \( \sigma \in \mathbb{R}_+ \) controls the step length
- the covariance matrix \( C \in \mathbb{R}^{n \times n} \) determines the shape of the distribution ellipsoid

here, all new points are sampled with the same parameters

The question remains how to update \( m \), \( C \), and \( \sigma \).
Normal Distribution

probability density of the 1-D standard normal distribution

probability density of a 2-D normal distribution
Normal Distribution

1-D case

probability density of the 1-D standard normal distribution \( \mathcal{N}(0, 1) \)
(expected (mean) value, variance) = (0,1)

\[
p(x) = \frac{1}{\sqrt{2\pi}} \exp\left( -\frac{x^2}{2} \right)
\]

General case

- Normal distribution \( \mathcal{N}(m, \sigma^2) \)
  (expected value, variance) = \((m, \sigma^2)\)
density: \( p_{m,\sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left( -\frac{(x-m)^2}{2\sigma^2} \right) \)

- A normal distribution is entirely determined by its mean value and variance

- The family of normal distributions is closed under linear transformations: if \( X \) is normally distributed then a linear transformation \( aX + b \) is also normally distributed

- Exercise: Show that \( m + \sigma \mathcal{N}(0, 1) = \mathcal{N}(m, \sigma^2) \)
Normal Distribution

General case

A random variable following a 1-D normal distribution is determined by its mean value $m$ and variance $\sigma^2$.

In the $n$-dimensional case it is determined by its mean vector and covariance matrix.

Covariance Matrix

If the entries in a vector $\mathbf{X} = (X_1, \ldots, X_n)^T$ are random variables, each with finite variance, then the covariance matrix $\Sigma$ is the matrix whose $(i, j)$ entries are the covariance of $(X_i, X_j)$

$$
\Sigma_{ij} = \text{cov}(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)]
$$

where $\mu_i = E(X_i)$. Considering the expectation of a matrix as the expectation of each entry, we have

$$
\Sigma = E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T]
$$

$\Sigma$ is symmetric, positive definite.
The Multi-Variate (\(n\)-Dimensional) Normal Distribution

Any multi-variate normal distribution \(\mathcal{N}(\mathbf{m}, \mathbf{C})\) is uniquely determined by its mean value \(\mathbf{m} \in \mathbb{R}^n\) and its symmetric positive definite \(n \times n\) covariance matrix \(\mathbf{C}\).

\[
\text{density: } p_{\mathcal{N}(\mathbf{m}, \mathbf{C})}(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{m})\right),
\]

The mean value \(\mathbf{m}\)

- determines the displacement (translation)
- value with the largest density (modal value)
- the distribution is symmetric about the distribution mean

\[
\mathcal{N}(\mathbf{m}, \mathbf{C}) = \mathbf{m} + \mathcal{N}(0, \mathbf{C})
\]

The covariance matrix \(\mathbf{C}\)

- determines the shape
- geometrical interpretation: any covariance matrix can be uniquely identified with the iso-density ellipsoid

\[
\{\mathbf{x} \in \mathbb{R}^n \mid (\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x} - \mathbf{m}) = 1\}
\]
any covariance matrix can be uniquely identified with the iso-density ellipsoid
\[ \{ x \in \mathbb{R}^n \mid (x - m)^T C^{-1} (x - m) = 1 \} \]

Lines of Equal Density

\[ \mathcal{N}(m, \sigma^2 I) \sim m + \sigma \mathcal{N}(0, I) \]
\text{one degree of freedom} \quad \sigma
\text{components are independent standard normally distributed}

\[ \mathcal{N}(m, D^2) \sim m + D \mathcal{N}(0, I) \]
\text{\( n \) degrees of freedom}
\text{components are independent, scaled}

\[ \mathcal{N}(m, C) \sim m + C^{1/2} \mathcal{N}(0, I) \]
\text{\( (n^2 + n)/2 \) degrees of freedom}
\text{components are correlated}

where \( I \) is the identity matrix (isotropic case) and \( D \) is a diagonal matrix (reasonable for separable problems) and \( A \times \mathcal{N}(0, I) \sim \mathcal{N}(0, AA^T) \) holds for all \( A \).
The \((\mu/\mu, \lambda)\)-ES

Non-elitist selection and intermediate (weighted) recombination

Given the \(i\)-th solution point \(x_i = m + \sigma \mathcal{N}_i(0, \mathbf{C}) = m + \sigma y_i\)

Let \(x_{i:\lambda}\) the \(i\)-th ranked solution point, such that \(f(x_{1:\lambda}) \leq \cdots \leq f(x_{\lambda:\lambda})\).

The new mean reads

\[
m \leftarrow \sum_{i=1}^{\mu} w_i x_{i:\lambda} = m + \sigma \sum_{i=1}^{\mu} w_i y_{i:\lambda} =: y_w
\]

where

\[
w_1 \geq \cdots \geq w_\mu > 0, \quad \sum_{i=1}^{\mu} w_i = 1, \quad \frac{1}{\sum_{i=1}^{\mu} w_i^2} =: \mu_w \approx \frac{\lambda}{4}
\]

The best \(\mu\) points are selected from the new solutions (non-elitistic) and weighted intermediate recombination is applied.
Invariance Under Monotonically Increasing Functions

Rank-based algorithms

Update of all parameters uses only the ranks

\[ f(x_1: \lambda) \leq f(x_2: \lambda) \leq \ldots \leq f(x_\lambda: \lambda) \]

\[ g(f(x_1: \lambda)) \leq g(f(x_2: \lambda)) \leq \ldots \leq g(f(x_\lambda: \lambda)) \quad \forall g \]

\(^3\) Whitley 1989. The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best, ICGA
Basic Invariance in Search Space

- **translation invariance**

  \[ f(x) \leftrightarrow f(x - a) \]

  is true for most optimization algorithms.

Identical behavior on \( f \) and \( f_a \)

\[
\begin{align*}
  f : & \quad x \mapsto f(x), & x^{(t=0)} &= x_0 \\
  f_a : & \quad x \mapsto f(x - a), & x^{(t=0)} &= x_0 + a
\end{align*}
\]

No difference can be observed w.r.t. the argument of \( f \)
Invariance Under Rigid Search Space Transformations

\[ f = h_{\text{Rast}} \]

- Level sets in dimension 2

\[ f = h \]

for example, invariance under search space rotation

(separable \( \Leftrightarrow \) non-separable)
Invariance Under Rigid Search Space Transformations

\[ f = h_{\text{Rast}} \circ R \]

\( f \)-level sets in dimension 2

\[ f = h \circ R \]

for example, invariance under search space rotation

(separable \( \Leftrightarrow \) non-separable)
Invariance

*The grand aim of all science is to cover the greatest number of empirical facts by logical deduction from the smallest number of hypotheses or axioms.*
— Albert Einstein

- **Empirical performance results**
  - from benchmark functions
  - from solved real world problems

  are only useful if they do generalize to other problems

- **Invariance** is a strong non-empirical statement about generalization

  generalizing (identical) performance from a single function to a whole class of functions

consequently, invariance is important for the evaluation of search algorithms
Comparison to BFGS, NEWUOA, PSO and DE

\( f \) convex quadratic, separable with varying condition number \( \alpha \)

Ellipsoid dimension 20, 21 trials, tolerance 1e^{-09}, eval max 1e+07

\[
f(x) = g(x^T H x) \quad \text{with} \\
H \text{ diagonal} \\
g \text{ identity (for BFGS and NEWUOA)} \\
g \text{ any order-preserving = strictly increasing function (for all other)}
\]

\( \text{SP1} = \text{average number of objective function evaluations}^{\text{14}} \) to reach the target function value of \( g^{-1}(10^{-9}) \)

---

\text{14} \quad \text{Auger et.al. (2009): Experimental comparisons of derivative free optimization algorithms, SEA}
Comparison to BFGS, NEWUOA, PSO and DE

\( f \) convex quadratic, non-separable (rotated) with varying condition number \( \alpha \)

Rotated Ellipsoid dimension 20, 21 trials, tolerance 1e−09, eval max 1e+07

\[ f(x) = g(x^T H x) \]

- **BFGS** (Broyden et al 1970)
- **NEWUOA** (Powell 2004)
- **DE** (Storn & Price 1996)
- **PSO** (Kennedy & Eberhart 1995)
- **CMA-ES** (Hansen & Ostermeier 2001)

\( H \) full

\( g \) identity (for BFGS and NEWUOA)

\( g \) any order-preserving = strictly increasing function (for all other)

\[ SP1 = \text{average number of objective function evaluations}^{15} \text{ to reach the target function value of } g^{-1}(10^{-9}) \]

---

\(^{15}\) Auger et.al. (2009): Experimental comparisons of derivative free optimization algorithms, SEA
Comparing Experiments

Comparison to BFGS, NEWUOA, PSO and DE

\( f \) non-convex, non-separable (rotated) with varying condition number \( \alpha \)

Sqrt of sqrt of rotated ellipsoid dimension 20, 21 trials, tolerance 1e\(^{-09} \), eval max 1e+07

\[ f(x) = g(x^T H x) \]

BFGS (Broyden et al 1970)
NEWUOA (Powell 2004)
DE (Storn & Price 1996)
PSO (Kennedy & Eberhart 1995)
CMA-ES (Hansen & Ostermeier 2001)

\( H \) full
\( g : x \mapsto x^{1/4} \) (for BFGS and NEWUOA)
\( g \) any order-preserving = strictly increasing function (for all other)

SP1 = average number of objective function evaluations\(^{16} \) to reach the target function value of \( g^{-1}(10^{-9}) \)

\(^{16}\) Auger et.al. (2009): Experimental comparisons of derivative free optimization algorithms, SEA
Zoom on ESs: Objectives

Illustrate why and how sampling distribution is controlled

- step-size control (overall standard deviation)
  allows to achieve linear convergence

- covariance matrix control
  allows to solve ill-conditioned problems
**Why Step-Size Control?**

![Graph showing function evaluations and step sizes]

- **Constant step-size**
- **Step-size too small**
- **Step-size too large**
- **Optimal step-size (scale invariant)**
- **Random search**

The graph demonstrates the performance of step-size control in optimization problems. The function $f(x) = \sum_{i=1}^{n} x_i^2$, defined for $n = 10$ in the interval $[-2.2, 0.8]^n$. The optimal step-size is scale invariant, ensuring efficient function value reduction over function evaluations.
Methods for Step-Size Control

- **1/5-th success rule**\(^ab\), often applied with “+”-selection
  
  increase step-size if more than 20\% of the new solutions are successful,
  decrease otherwise

- **σ-self-adaptation**\(^c\), applied with “,”-selection
  
  mutation is applied to the step-size and the better, according to the
  objective function value, is selected

  simplified “global” self-adaptation

- **path length control**\(^d\) (Cumulative Step-size Adaptation, CSA)\(^e\)
  
  self-adaptation derandomized and non-localized

---


\(^b\) Schumer and Steiglitz 1968. Adaptive step size random search. *IEEE TAC*


\(^e\) Ostermeier et al 1994, Step-size adaptation based on non-local use of selection information, *PPSN IV*
Path Length Control (CSA)
The Concept of Cumulative Step-Size Adaptation

\[ x_i = m + \sigma y_i \]

\[ m \leftarrow m + \sigma y_w \]

Measure the length of the *evolution path*

the pathway of the mean vector \( m \) in the generation sequence

loosely speaking steps are

- perpendicular under random selection (in expectation)
- perpendicular in the desired situation (to be most efficient)
Path Length Control (CSA)

The Equations

Initialize $\mathbf{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, evolution path $\mathbf{p}_\sigma = \mathbf{0}$, set $c_\sigma \approx 4/n$, $d_\sigma \approx 1$.

\[
\begin{align*}
\mathbf{m} &\leftarrow \mathbf{m} + \sigma \mathbf{y}_w & \text{where } \mathbf{y}_w = \sum_{i=1}^{\mu} w_i \mathbf{y}_i : \lambda \\
\mathbf{p}_\sigma &\leftarrow (1 - c_\sigma) \mathbf{p}_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w} \mathbf{y}_w \\
\sigma &\leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma\|}{\mathbb{E}\|\mathcal{N}(0,1)\|} - 1\right)\right)
\end{align*}
\]

$>1 \iff \|\mathbf{p}_\sigma\|$ is greater than its expectation
Path Length Control (CSA)

The Equations

Initialize $m \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, evolution path $p_\sigma = 0$,
set $c_\sigma \approx 4/n$, $d_\sigma \approx 1$.

\[
\begin{align*}
  m &\leftarrow m + \sigma y_w \quad \text{where } y_w = \sum_{i=1}^{\mu} w_i y_i : \lambda \\
  p_\sigma &\leftarrow (1 - c_\sigma) p_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w} y_w \\
  \sigma &\leftarrow \sigma \times \exp \left( \frac{c_\sigma}{d_\sigma} \left( \frac{||p_\sigma||}{\mathbb{E}||\mathcal{N}(0, 1)||} - 1 \right) \right)
\end{align*}
\]

update mean

accounts for $1 - c_\sigma$

accounts for $w_i$

update step-size

$>1 \iff ||p_\sigma||$ is greater than its expectation
(5/5, 10)-CSA-ES, default parameters

\[ f(x) = \sum_{i=1}^{n} x_i^2 \]

in \([-0.2, 0.8]^n\)

for \(n = 30\)
Evolution Strategies

Recalling

New search points are sampled normally distributed

\[ x_i \sim m + \sigma N_i(0, C) \quad \text{for } i = 1, \ldots, \lambda \]

as perturbations of \( m \), where \( x_i, m \in \mathbb{R}^n, \sigma \in \mathbb{R}_+, C \in \mathbb{R}^{n \times n} \)

where

- the mean vector \( m \in \mathbb{R}^n \) represents the favorite solution
- the so-called step-size \( \sigma \in \mathbb{R}_+ \) controls the step length
- the covariance matrix \( C \in \mathbb{R}^{n \times n} \) determines the shape of the distribution ellipsoid

The remaining question is how to update \( C \).
Covariance Matrix Adaptation

Rank-One Update

\[ m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i y_i, \quad y_i \sim \mathcal{N}_i(0, C) \]

initial distribution, \( C = I \)
Covariance Matrix Adaptation

Rank-One Update

\[
\mathbf{m} \leftarrow \mathbf{m} + \sigma \mathbf{y}_w, \quad \mathbf{y}_w = \sum_{i=1}^{\mu} w_i \mathbf{y}_i, \quad \mathbf{y}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C})
\]

initial distribution, \( \mathbf{C} = \mathbf{I} \)
Covariance Matrix Adaptation

Rank-One Update

\[ m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i y_i : \lambda, \quad y_i \sim \mathcal{N}_i(0, C) \]

\( y_w \), movement of the population mean \( m \) (disregarding \( \sigma \))
Covariance Matrix Adaptation

Rank-One Update

\[ m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i y_i, \quad y_i \sim \mathcal{N}_i(0, C) \]

mixture of distribution \( C \) and step \( y_w \),

\[ C \leftarrow 0.8 \times C + 0.2 \times y_w y_w^T \]

... equations
Covariance Matrix Adaptation (CMA)

Covariance Matrix Rank-One Update

**Rank-One Update**

\[
m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i y_i, \quad y_i \sim \mathcal{N}_i(0, C)
\]

*new distribution (disregarding $\sigma$)*
Covariance Matrix Adaptation

Rank-One Update

\[ m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i y_{i:}, \quad y_i \sim \mathcal{N}_i(0, C) \]

new distribution (disregarding \( \sigma \))
Covariance Matrix Adaptation

Rank-One Update

\[ m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i y_i^{\lambda}, \quad y_i \sim \mathcal{N}_i(0, C) \]

movement of the population mean \( m \)
Covariance Matrix Adaptation

Rank-One Update

\[ m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i y_i: \lambda, \quad y_i \sim \mathcal{N}_i(0, \mathbf{C}) \]

mixture of distribution \( \mathbf{C} \) and step \( y_w \),
\[ \mathbf{C} \leftarrow 0.8 \times \mathbf{C} + 0.2 \times y_w y_w^T \]
Covariance Matrix Adaptation
Rank-One Update

\[ m \leftarrow m + \sigma y_w, \quad y_w = \sum_{i=1}^{\mu} w_i y_i, \quad y_i \sim \mathcal{N}_i(0, C) \]

new distribution,
\[ C \leftarrow 0.8 \times C + 0.2 \times y_w y_w^T \]
the ruling principle: the adaptation increases the likelihood of successful steps, \( y_w \), to appear again
another viewpoint: the adaptation follows a natural gradient approximation of the expected fitness
Covariance Matrix Adaptation

Rank-One Update

Initialize \( m \in \mathbb{R}^n \), and \( C = I \), set \( \sigma = 1 \), learning rate \( c_{\text{cov}} \approx \frac{2}{n^2} \)

While not terminate

\[
\begin{align*}
\mathbf{x}_i &= m + \sigma \mathbf{y}_i, \\
\mathbf{y}_i &\sim \mathcal{N}_i(0, C), \\
\mathbf{m} &\leftarrow \mathbf{m} + \sigma \mathbf{y}_w \\
\mathbf{C} &\leftarrow (1 - c_{\text{cov}}) \mathbf{C} + c_{\text{cov}} \mu_w \mathbf{y}_w \mathbf{y}_w^T \text{ where } \mu_w = \frac{1}{\sum_{i=1}^\mu w_i^2} \geq 1
\end{align*}
\]

The rank-one update has been found independently in several domains\(^6\)\(^7\)\(^8\)\(^9\)

---

\(^7\) Hansen&Ostermeier 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, ICEC
\(^8\) Ljung 1999. System Identification: Theory for the User
\(^9\) Haario et al 2001. An adaptive Metropolis algorithm, JSTOR
The CMA-ES

Input: \( m \in \mathbb{R}^n, \sigma \in \mathbb{R}_+, \lambda \)

Initialize: \( C = I, \) and \( p_c = 0, p_\sigma = 0, \)

Set: \( c_c \approx 4/n, c_\sigma \approx 4/n, c_1 \approx 2/n^2, c_\mu \approx \mu_w/n^2, c_1 + c_\mu \leq 1, d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}, \)
and \( w_{i=1\ldots\lambda} \) such that \( \mu_w = \frac{1}{\sum_{i=1}^{\lambda} w_i^2} \approx 0.3 \lambda \)

While not terminate

\[
  x_i = m + \sigma y_i, \quad y_i \sim \mathcal{N}_i(0, C), \quad \text{for } i = 1, \ldots, \lambda
\]

sampling

\[
  m \leftarrow \sum_{i=1}^\mu w_i x_{i:\lambda} = m + \sigma y_w \quad \text{where } y_w = \sum_{i=1}^\mu w_i y_{i:\lambda}
\]

update mean

\[
  p_c \leftarrow (1 - c_c) p_c + 1_{\{\|p_\sigma\| < 1.5 \sqrt{n}\}} \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w} y_w
\]

cumulation for \( C \)

\[
  p_\sigma \leftarrow (1 - c_\sigma) p_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w} C^{-\frac{1}{2}} y_w
\]

cumulation for \( \sigma \)

\[
  C \leftarrow (1 - c_1 - c_\mu) C + c_1 p_c p_c^T + c_\mu \sum_{i=1}^\mu w_i y_{i:\lambda} y_{i:\lambda}^T
\]

update \( C \)

\[
  \sigma \leftarrow \sigma \times \exp \left( \frac{c_\sigma}{d_\sigma} \left( \frac{\|p_\sigma\|}{\mathbb{E}\|\mathcal{N}(0, I)\|} - 1 \right) \right)
\]

update of \( \sigma \)

Not covered on this slide: termination, restarts, useful output, boundaries and encoding
Experimentum Crucis (0)

What did we want to achieve?

- reduce any convex-quadratic function
  \[ f(x) = x^T H x \]
  
  e.g. \[ f(x) = \sum_{i=1}^{n} 10^{i-1} x_i^2 \]

  to the sphere model
  \[ f(x) = x^T x \]

  without use of derivatives

- lines of equal density align with lines of equal fitness

  \[ C \propto H^{-1} \]

  in a stochastic sense
Experimentum Crucis (1)

$f$ convex quadratic, separable

\[ f(x) = \sum_{i=1}^{n} 10^{\alpha \frac{i-1}{n-1}} x_i^2, \alpha = 6 \]
Experimentum Crucis (2)

$f$ convex quadratic, as before but non-separable (rotated)

\[ f(x) = g(x^T H x), \quad g : \mathbb{R} \to \mathbb{R} \text{ strictly increasing} \]

\[ C \propto H^{-1} \text{ for all } g, H \]