

Derivative Free Optimization

Anne Auger (Inria and CMAP, Ecole Polytechnique)
Laurent Dumas (U. Versailles)

AMS Master - Optimization Paris-Saclay Master

RandOpt Team
Inria and CMAP (Ecole Polytechnique)
*anne.auger@inria.fr

2017 — 2018

Course Organization

Part I - Stochastic/Randomized Methods

taught by Anne Auger

Friday 1/12, 08/12, 15/12, 22/12, 12/01 from 2pm to 5:15

Small project graded (grade is part of final grade)

Part II - Deterministic Methods

taught by Laurent Dumas

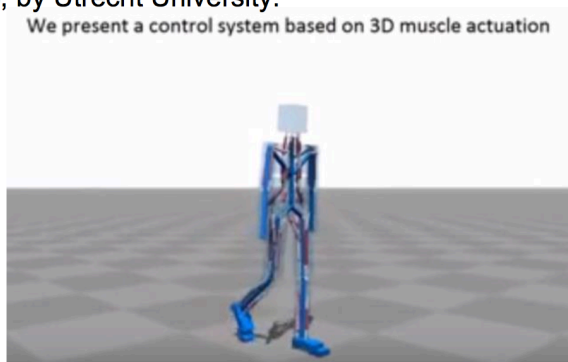
Friday 19/01, 26/01, 02/02, 09/02, 16/02 from 2pm to 5:15

Final Written Exam

Friday 23/02

First Example of a Black-Box Continuous Optimization Problem

Computer simulation teaches itself to walk upright (virtual robots (of different shapes) learning to walk, through stochastic optimization (CMA-ES)), by Utrecht University:



<https://www.youtube.com/watch?v=yCi5Fu10vk>

T. Geitjtenbeek, M. Van de Panne, F. Van der Stappen: "Flexible Muscle-Based Locomotion for Bipedal Creatures", SIGGRAPH Asia, 2013.

Problem Statement

Continuous Domain Search/Optimization

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

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

- ▶ Black Box scenario (direct search scenario)



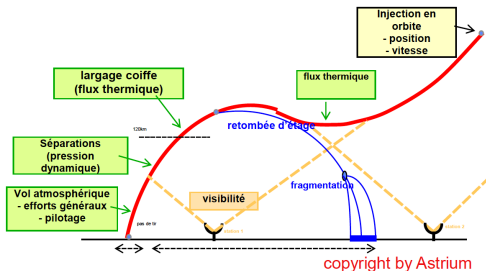
- ▶ gradients are not available or not useful
 - ▶ problem domain specific knowledge is used only within the black box, e.g. within an appropriate encoding
- ▶ Search costs: number of function evaluations (often called runtime of algorithms)
 - ⚠: this is not the "real" runtime (i.e. time you have to wait) but this is typically proportional to the real runtime. This measurement is independent of the programming language/ implementation "tricks" chosen for the implementation.

Optimization of the Design of a Launcher

Example of a black-box problem



Poppy

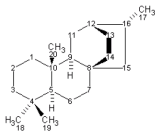
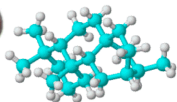
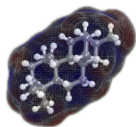


- Scenario: multi-stage launcher brings a satellite into orbit
- Minimize the overall cost of a launch
- Parameters: propellant mass of each stage / diameter of each stage / flux of each engine / parameters of the command law

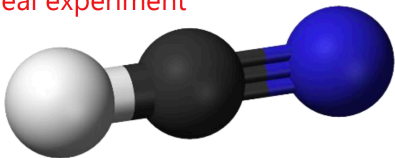
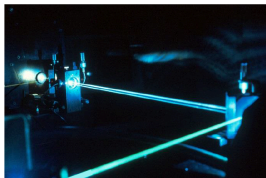
*23 continuous parameters to optimize
+ constraints*

Control of the Alignment of Molecules

Example of a black-box problem (II)



Objective function:
via **numerical simulation**
or a **real experiment**



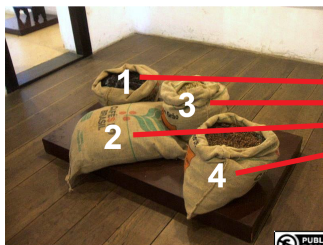
possible application in drug design

Coffee Tasting Problem

Example of a black-box problem (III)

Coffee Tasting Problem

- ▶ Find a mixture of coffee in order to keep the coffee taste from one year to another
- ▶ Objective function = opinion of one expert



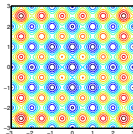
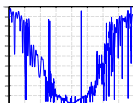
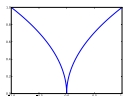
Quasipalm

M. Herdy: "Evolution Strategies with subjective selection", 1996

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



gradient direction Newton direction

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 100 points onto a real interval, say $[0, 1]$.

How many points would you need to get a similar coverage (in terms of distance between adjacent points) in dimension 10?

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 100 points onto a real interval, say $[0, 1]$. To get **similar coverage**, in terms of distance between adjacent points, of the 10-dimensional space $[0, 1]^{10}$ would require $100^{10} = 10^{20}$ points. A 100 points appear now as isolated points in a vast empty space.

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

Curse of Dimensionality

How long would it take to evaluate 10^{20} points ?

```
import timeit
timeit.timeit('import numpy as np ;
np.sum(np.ones(10)*np.ones(10))', number=1000000)
```

> 7.0521080493927

7 seconds for 10^6 evaluations of $f(x) = \sum_{i=1}^{10} x_i^2$

We would need **more than 10^8 days** for evaluating 10^{20} points

[As a reference: origin of human species: roughly $6 \cdot 10^8$ days]

Separable Problems

Definition (Separable Problem)

A function f is separable if

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

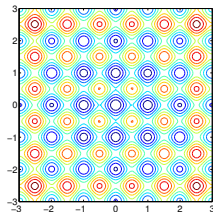
⇒ it follows that f can be optimized in a sequence of n independent 1-D optimization processes

Example: Additively decomposable functions

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

Rastrigin function

$$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$$



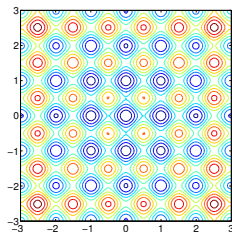
Non-Separable Problems

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

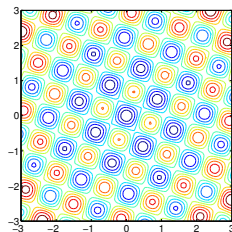
Rotating the coordinate system

- ▶ $f : \mathbf{x} \mapsto f(\mathbf{x})$ separable
- ▶ $f : \mathbf{x} \mapsto f(\mathbf{R}\mathbf{x})$ non-separable

R rotation matrix



R
→



¹Hansen, Ostermeier, Gawelczyk (1995). On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. Sixth ICGA, pp. 57-64, Morgan Kaufmann

²Salomon (1996). "Reevaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions; A survey of some theoretical and practical aspects of genetic algorithms." BioSystems, 39(3):263-278

Ill-Conditioned Problems

- ▶ If f is convex quadratic, $f : \mathbf{x} \mapsto \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} = \frac{1}{2} \sum_i h_{i,i} x_i^2 + \frac{1}{2} \sum_{i \neq j} h_{i,j} x_i x_j$, with \mathbf{H} positive, definite, symmetric matrix
 \mathbf{H} is the Hessian matrix of f
- ▶ ill-conditioned means a high condition number of Hessian Matrix \mathbf{H}

$$\text{cond}(\mathbf{H}) = \frac{\lambda_{\max}(\mathbf{H})}{\lambda_{\min}(\mathbf{H})}$$

Example / exercise

The level-sets of a function are defined as

$$\mathcal{L}_c = \{x \in \mathbb{R}^n | f(x) = c\}, c \in \mathbb{R}.$$

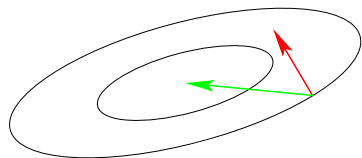
Consider the objective function $f(\mathbf{x}) = \frac{1}{2}(x_1^2 + 9x_2^2)$

1. Plot the level sets of f .
2. Compute the condition number of the Hessian matrix of f , relate it to the axis ratio of the level sets of f .
3. Generalize 1. and 2. to a general convex-quadratic function.

Ill-conditioned Problems

consider the curvature of the level sets of a function

ill-conditioned means “squeezed” lines of equal function value (high curvatures)



gradient direction $-f'(\mathbf{x})^T$

Newton direction
 $-H^{-1}f'(\mathbf{x})^T$

Condition number equals nine here. Condition numbers up to 10^{10} are not unusual in real world problems.

Landscape of Derivative Free Optimization Algorithms

Deterministic Algorithms

Quasi-Newton with estimation of gradient (BFGS) [Broyden et al. 1970]

Simplex downhill [Nelder and Mead 1965]

Pattern search [Hooke and Jeeves 1961]

Trust-region methods (NEWUOA, BOBYQA) [Powell 2006, 2009]

Stochastic (randomized) search methods

Evolutionary Algorithms (continuous domain)

- ▶ Differential Evolution [Storn and Price 1997]
- ▶ Particle Swarm Optimization [Kennedy and Eberhart 1995]
- ▶ **Evolution Strategies, CMA-ES** [Rechenberg 1965, Hansen and Ostermeier 2001]
- ▶ Estimation of Distribution Algorithms (EDAs) [Larrañaga, Lozano, 2002]
- ▶ Cross Entropy Method (same as EDA) [Rubinstein, Kroese, 2004]
- ▶ Genetic Algorithms [Holland 1975, Goldberg 1989]

Simulated annealing [Kirkpatrick et al. 1983]

Simultaneous perturbation stochastic approximation (SPSA) [Spall 2000]

Stochastic Search

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

Initialize distribution parameters θ , set population size $\lambda \in \mathbb{N}$

While not terminate

1. Sample distribution $P(\mathbf{x}|\theta) \rightarrow \mathbf{x}_1, \dots, \mathbf{x}_\lambda \in \mathbb{R}^n$
2. Evaluate $\mathbf{x}_1, \dots, \mathbf{x}_\lambda$ on f
3. Update parameters $\theta \leftarrow F_\theta(\theta, \mathbf{x}_1, \dots, \mathbf{x}_\lambda, f(\mathbf{x}_1), \dots, f(\mathbf{x}_\lambda))$

Stochastic Search

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

Initialize distribution parameters θ , set population size $\lambda \in \mathbb{N}$

While not terminate

1. Sample distribution $P(\mathbf{x}|\theta) \rightarrow \mathbf{x}_1, \dots, \mathbf{x}_\lambda \in \mathbb{R}^n$
2. Evaluate $\mathbf{x}_1, \dots, \mathbf{x}_\lambda$ on f
3. Update parameters $\theta \leftarrow F_\theta(\theta, \mathbf{x}_1, \dots, \mathbf{x}_\lambda, f(\mathbf{x}_1), \dots, f(\mathbf{x}_\lambda))$

Stochastic Search

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

Initialize distribution parameters θ , set population size $\lambda \in \mathbb{N}$

While not terminate

1. Sample distribution $P(\mathbf{x}|\theta) \rightarrow \mathbf{x}_1, \dots, \mathbf{x}_\lambda \in \mathbb{R}^n$
2. Evaluate $\mathbf{x}_1, \dots, \mathbf{x}_\lambda$ on f
3. Update parameters $\theta \leftarrow F_\theta(\theta, \mathbf{x}_1, \dots, \mathbf{x}_\lambda, f(\mathbf{x}_1), \dots, f(\mathbf{x}_\lambda))$

Stochastic Search

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

Initialize distribution parameters θ , set population size $\lambda \in \mathbb{N}$

While not terminate

1. Sample distribution $P(\mathbf{x}|\theta) \rightarrow \mathbf{x}_1, \dots, \mathbf{x}_\lambda \in \mathbb{R}^n$
2. Evaluate $\mathbf{x}_1, \dots, \mathbf{x}_\lambda$ on f
3. Update parameters $\theta \leftarrow F_\theta(\theta, \mathbf{x}_1, \dots, \mathbf{x}_\lambda, f(\mathbf{x}_1), \dots, f(\mathbf{x}_\lambda))$

Stochastic Search

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

Initialize distribution parameters θ , set population size $\lambda \in \mathbb{N}$

While not terminate

1. Sample distribution $P(\mathbf{x}|\theta) \rightarrow \mathbf{x}_1, \dots, \mathbf{x}_\lambda \in \mathbb{R}^n$
2. Evaluate $\mathbf{x}_1, \dots, \mathbf{x}_\lambda$ on f
3. Update parameters $\theta \leftarrow F_\theta(\theta, \mathbf{x}_1, \dots, \mathbf{x}_\lambda, f(\mathbf{x}_1), \dots, f(\mathbf{x}_\lambda))$

Stochastic Search

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

Initialize distribution parameters θ , set population size $\lambda \in \mathbb{N}$

While not terminate

1. Sample distribution $P(\mathbf{x}|\theta) \rightarrow \mathbf{x}_1, \dots, \mathbf{x}_\lambda \in \mathbb{R}^n$
2. Evaluate $\mathbf{x}_1, \dots, \mathbf{x}_\lambda$ on f
3. Update parameters $\theta \leftarrow F_\theta(\theta, \mathbf{x}_1, \dots, \mathbf{x}_\lambda, f(\mathbf{x}_1), \dots, f(\mathbf{x}_\lambda))$

Stochastic Search

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

Initialize distribution parameters θ , set population size $\lambda \in \mathbb{N}$

While not terminate

1. Sample distribution $P(\mathbf{x}|\theta) \rightarrow \mathbf{x}_1, \dots, \mathbf{x}_\lambda \in \mathbb{R}^n$
2. Evaluate $\mathbf{x}_1, \dots, \mathbf{x}_\lambda$ on f
3. Update parameters $\theta \leftarrow F_\theta(\theta, \mathbf{x}_1, \dots, \mathbf{x}_\lambda, f(\mathbf{x}_1), \dots, f(\mathbf{x}_\lambda))$

Everything depends on the definition of P and F_θ

Stochastic Search

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

Initialize distribution parameters θ , set population size $\lambda \in \mathbb{N}$

While not terminate

1. Sample distribution $P(\mathbf{x}|\theta) \rightarrow \mathbf{x}_1, \dots, \mathbf{x}_\lambda \in \mathbb{R}^n$
2. Evaluate $\mathbf{x}_1, \dots, \mathbf{x}_\lambda$ on f
3. Update parameters $\theta \leftarrow F_\theta(\theta, \mathbf{x}_1, \dots, \mathbf{x}_\lambda, f(\mathbf{x}_1), \dots, f(\mathbf{x}_\lambda))$

Everything depends on the definition of P and F_θ

In Evolutionary Algorithms the distribution P is often implicitly defined via [operators on a population](#), in particular, selection, recombination and mutation

Natural template for *Estimation of Distribution Algorithms*

A Simple Example: The Pure Random Search

Also an Ineffective Example

The Pure Random Search

- ▶ Sample *uniformly* at random a solution
- ▶ Return the best solution ever found

Exercise

See the exercise on the document "Exercices - class 1".

Non-adaptive Algorithm

For the pure random search $P(\mathbf{x}|\theta)$ is independent of θ (i.e. no θ to be adapted): the algorithm is "blind"

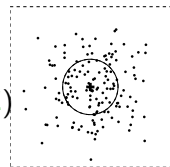
In this class: present algorithms that are "much better" than that

Evolution Strategies

New search points are sampled normally distributed

$$\mathbf{x}_i = \mathbf{m} + \sigma \mathbf{y}_i \quad \text{for } i = 1, \dots, \lambda \text{ with } \mathbf{y}_i \text{ i.i.d. } \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) :$$

as perturbations of \mathbf{m} , where $\mathbf{x}_i, \mathbf{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$,
 $\mathbf{C} \in \mathbb{R}^{n \times n}$

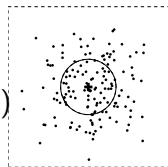


Evolution Strategies

New search points are sampled normally distributed

$$\mathbf{x}_i = \mathbf{m} + \sigma \mathbf{y}_i \quad \text{for } i = 1, \dots, \lambda \text{ with } \mathbf{y}_i \text{ i.i.d. } \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) :$$

as perturbations of \mathbf{m} , where $\mathbf{x}_i, \mathbf{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$,
 $\mathbf{C} \in \mathbb{R}^{n \times n}$



where

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

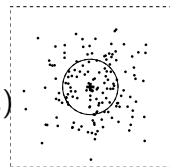
here, all new points are sampled with the same parameters

Evolution Strategies

New search points are sampled normally distributed

$$\mathbf{x}_i = \mathbf{m} + \sigma \mathbf{y}_i \quad \text{for } i = 1, \dots, \lambda \text{ with } \mathbf{y}_i \text{ i.i.d. } \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) :$$

as perturbations of \mathbf{m} , where $\mathbf{x}_i, \mathbf{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$,
 $\mathbf{C} \in \mathbb{R}^{n \times n}$



where

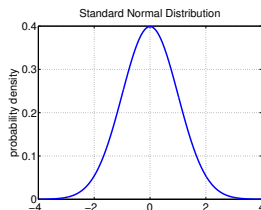
- ▶ the **mean** vector $\mathbf{m} \in \mathbb{R}^n$ represents the favorite solution
- ▶ the so-called **step-size** $\sigma \in \mathbb{R}_+$ controls the *step length*
- ▶ the **covariance matrix** $\mathbf{C} \in \mathbb{R}^{n \times n}$ determines the **shape** of the distribution ellipsoid

here, all new points are sampled with the same parameters

The question remains how to update \mathbf{m} , \mathbf{C} , and σ .

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, σ^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
- ▶ **Exercice:** 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 σ^2 .

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

Covariance Matrix

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

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

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

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

Σ is symmetric, positive definite

The Multi-Variate (n -Dimensional) Normal Distribution

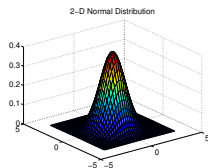
Any multi-variate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$ is uniquely determined by its mean value $\mathbf{m} \in \mathbb{R}^n$ and its symmetric positive definite $n \times n$ covariance matrix \mathbf{C} .

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

The mean value \mathbf{m}

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

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



The Multi-Variate (n -Dimensional) Normal Distribution

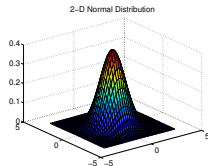
Any multi-variate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$ is uniquely determined by its mean value $\mathbf{m} \in \mathbb{R}^n$ and its symmetric positive definite $n \times n$ covariance matrix \mathbf{C} .

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

The mean value \mathbf{m}

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

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

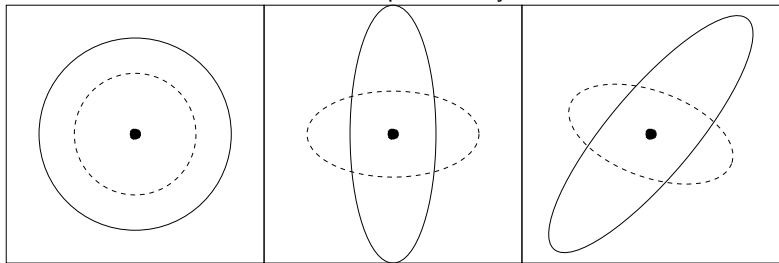


The covariance matrix \mathbf{C}

- ▶ 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



$$\mathcal{N}(m, \sigma^2 \mathbf{I}) \sim m + \sigma \mathcal{N}(0, \mathbf{I})$$

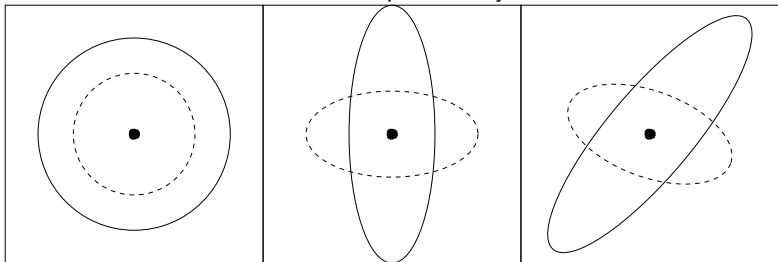
one degree of freedom σ

components are
independent standard
normally distributed

where \mathbf{I} is the identity matrix (isotropic case) and \mathbf{D} is a diagonal matrix (reasonable for separable problems) and $\mathbf{A} \times \mathcal{N}(0, \mathbf{I}) \sim \mathcal{N}(0, \mathbf{A}\mathbf{A}^T)$ holds for all \mathbf{A} .

... 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



$$\mathcal{N}(m, \sigma^2 \mathbf{I}) \sim m + \sigma \mathcal{N}(0, \mathbf{I})$$

one degree of freedom σ

components are
independent standard
normally distributed

$$\mathcal{N}(m, \mathbf{D}^2) \sim m + \mathbf{D} \mathcal{N}(0, \mathbf{I})$$

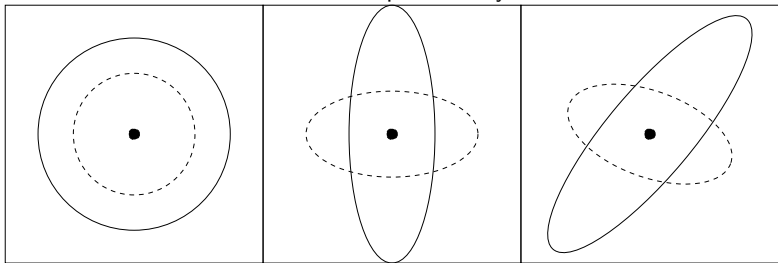
n degrees of freedom

components are
independent, scaled

where \mathbf{I} is the identity matrix (isotropic case) and \mathbf{D} is a diagonal matrix (reasonable for separable problems) and $\mathbf{A} \times \mathcal{N}(0, \mathbf{I}) \sim \mathcal{N}(0, \mathbf{A}\mathbf{A}^T)$ holds for all \mathbf{A} .

... 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



$\mathcal{N}(m, \sigma^2 \mathbf{I}) \sim m + \sigma \mathcal{N}(0, \mathbf{I})$
 one degree of freedom σ
 components are independent standard normally distributed

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

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

where \mathbf{I} is the identity matrix (isotropic case) and \mathbf{D} is a diagonal matrix (reasonable for separable problems) and $\mathbf{A} \times \mathcal{N}(0, \mathbf{I}) \sim \mathcal{N}(0, \mathbf{A}\mathbf{A}^T)$ holds for all \mathbf{A} .

Where are we?

Problem Statement

- Black Box Optimization and Its Difficulties
- Non-Separable Problems
- Ill-Conditioned Problems

Stochastic search algorithms - basics

- A Search Template
- A Natural Search Distribution: the Normal Distribution
- Adaptation of Distribution Parameters: What to Achieve?

Adaptive Evolution Strategies

- Mean Vector Adaptation
- Step-size control
 - Theory
 - Algorithms
- Covariance Matrix Adaptation
 - Rank-One Update
 - Cumulation—the Evolution Path
 - Rank- μ Update

Adaptation: What do we want to achieve?

New search points are sampled normally distributed

$$\mathbf{x}_i = \mathbf{m} + \sigma \mathbf{y}_i \quad \text{for } i = 1, \dots, \lambda \text{ with } \mathbf{y}_i \text{ i.i.d. } \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$$

where $\mathbf{x}_i, \mathbf{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\mathbf{C} \in \mathbb{R}^{n \times n}$

- ▶ the **mean** vector should represent the favorite solution
- ▶ the **step-size** controls the step-length and thus convergence rate
 - should allow to reach fastest convergence rate possible
- ▶ the **covariance matrix** $\mathbf{C} \in \mathbb{R}^{n \times n}$ determines the **shape** of the distribution ellipsoid
 - adaptation should allow to learn the “topography” of the problem
 - particularly important for **ill-conditioned** problems
 - $\mathbf{C} \propto \mathbf{H}^{-1}$ on convex quadratic functions

Problem Statement

Black Box Optimization and Its Difficulties
Non-Separable Problems
Ill-Conditioned Problems

Stochastic search algorithms - basics

A Search Template
A Natural Search Distribution: the Normal Distribution
Adaptation of Distribution Parameters: What to Achieve?

Adaptive Evolution Strategies

Mean Vector Adaptation

Step-size control

Theory

Algorithms

Covariance Matrix Adaptation

Rank-One Update

Cumulation—the Evolution Path

Rank- μ Update

Evolution Strategies (ES)

Simple Update for Mean Vector

Let μ : # parents, λ : # offspring

Plus (elitist) and comma (non-elitist) selection

$(\mu + \lambda)$ -ES: selection in $\{\text{parents}\} \cup \{\text{offspring}\}$

(μ, λ) -ES: selection in $\{\text{offspring}\}$

ES algorithms emerged in the community of bio-inspired methods where a parallel between optimization and evolution of species as described by Darwin served in the origin as inspiration for the methods. Nowadays this parallel is mainly visible through the terminology used: candidate solutions are parents or offspring, the objective function is a fitness function, ...

$(1 + 1)$ -ES

Sample one offspring from parent \mathbf{m}

$$\mathbf{x} = \mathbf{m} + \sigma \mathcal{N}(\mathbf{0}, \mathbf{C})$$

If \mathbf{x} better than \mathbf{m} select

$$\mathbf{m} \leftarrow \mathbf{x}$$

The $(\mu/\mu, \lambda)$ -ES - Update of the mean vector

Non-elitist selection and intermediate (weighted) recombination

Given the i -th solution point $\mathbf{x}_i = \mathbf{m} + \sigma \underbrace{\mathbf{y}_i}_{\sim \mathcal{N}(\mathbf{0}, \mathbf{C})}$

Let $\mathbf{x}_{i:\lambda}$ the i -th ranked solution point, such that $f(\mathbf{x}_{1:\lambda}) \leq \dots \leq f(\mathbf{x}_{\lambda:\lambda})$.

Notation: we denote $\mathbf{y}_{i:\lambda}$ the vector such that $\mathbf{x}_{i:\lambda} = \mathbf{m} + \sigma \mathbf{y}_{i:\lambda}$

Exercise: realize that $\mathbf{y}_{i:\lambda}$ is generally not distributed as $\mathcal{N}(\mathbf{0}, \mathbf{C})$

The best μ points are selected from the new solutions (non-elitistic) and weighted intermediate recombination is applied.

The $(\mu/\mu, \lambda)$ -ES - Update of the mean vector

Non-elitist selection and intermediate (weighted) recombination

Given the i -th solution point $\mathbf{x}_i = \mathbf{m} + \sigma \underbrace{\mathbf{y}_i}_{\sim \mathcal{N}(\mathbf{0}, \mathbf{C})}$

Let $\mathbf{x}_{i:\lambda}$ the i -th ranked solution point, such that $f(\mathbf{x}_{1:\lambda}) \leq \dots \leq f(\mathbf{x}_{\lambda:\lambda})$.

Notation: we denote $\mathbf{y}_{i:\lambda}$ the vector such that $\mathbf{x}_{i:\lambda} = \mathbf{m} + \sigma \mathbf{y}_{i:\lambda}$

Exercise: realize that $\mathbf{y}_{i:\lambda}$ is generally not distributed as $\mathcal{N}(\mathbf{0}, \mathbf{C})$

The new mean reads

$$\mathbf{m} \leftarrow \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}$$

where

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

The best μ points are selected from the new solutions (non-elitistic) and weighted intermediate recombination is applied.

The $(\mu/\mu, \lambda)$ -ES - Update of the mean vector

Non-elitist selection and intermediate (weighted) recombination

Given the i -th solution point $\mathbf{x}_i = \mathbf{m} + \sigma \underbrace{\mathbf{y}_i}_{\sim \mathcal{N}(\mathbf{0}, \mathbf{C})}$

Let $\mathbf{x}_{i:\lambda}$ the i -th ranked solution point, such that $f(\mathbf{x}_{1:\lambda}) \leq \dots \leq f(\mathbf{x}_{\lambda:\lambda})$.

Notation: we denote $\mathbf{y}_{i:\lambda}$ the vector such that $\mathbf{x}_{i:\lambda} = \mathbf{m} + \sigma \mathbf{y}_{i:\lambda}$

Exercise: realize that $\mathbf{y}_{i:\lambda}$ is generally not distributed as $\mathcal{N}(\mathbf{0}, \mathbf{C})$

The new mean reads

$$\mathbf{m} \leftarrow \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda} = \mathbf{m} + \sigma \underbrace{\sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}}_{=: \mathbf{y}_w}$$

where

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

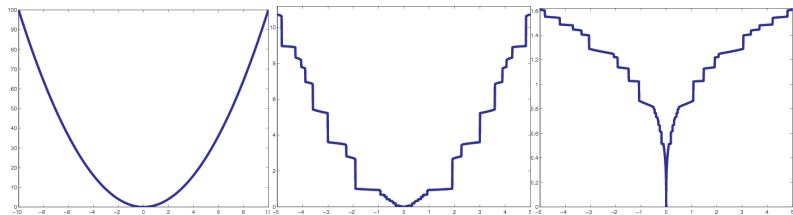
The best μ points are selected from the new solutions (non-elitistic) and weighted intermediate recombination is applied.

Invariance Under Monotonically Increasing Functions

Rank-based algorithms

Update of all parameters uses only the ranks

$$f(x_{1:\lambda}) \leq f(x_{2:\lambda}) \leq \dots \leq f(x_{\lambda:\lambda})$$



$$g(f(x_{1:\lambda})) \leq g(f(x_{2:\lambda})) \leq \dots \leq g(f(x_{\lambda:\lambda})) \quad \forall g$$

g is strictly monotonically increasing
 g preserves ranks

Problem Statement

Black Box Optimization and Its Difficulties
Non-Separable Problems
Ill-Conditioned Problems

Stochastic search algorithms - basics

A Search Template
A Natural Search Distribution: the Normal Distribution
Adaptation of Distribution Parameters: What to Achieve?

Adaptive Evolution Strategies

Mean Vector Adaptation

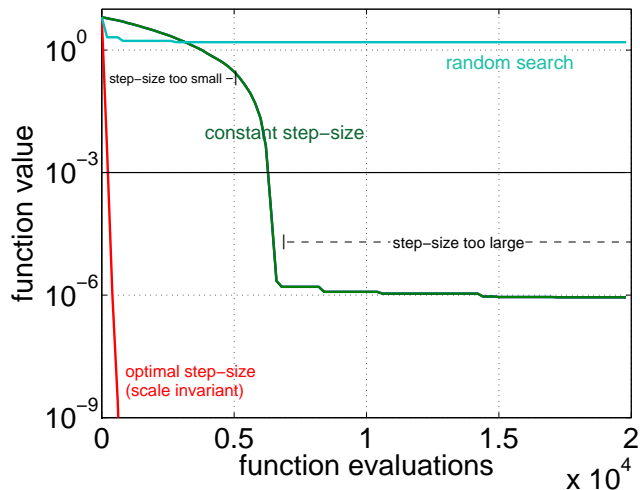
Step-size control

Theory
Algorithms

Covariance Matrix Adaptation

Rank-One Update
Cumulation—the Evolution Path
Rank- μ Update

Why Step-Size Control?



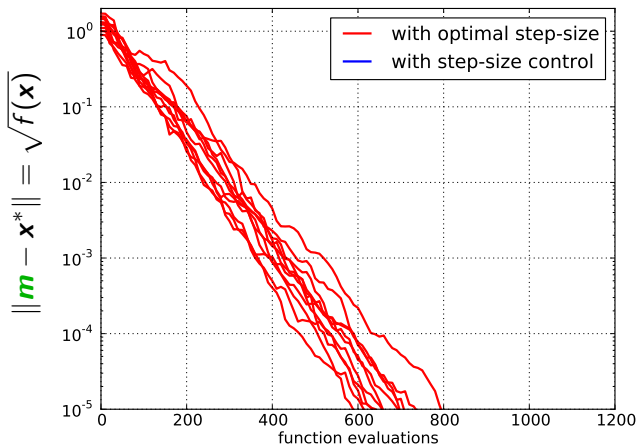
(1+1)-ES
(red & green)

$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$$

in $[-2.2, 0.8]^n$
for $n = 10$

Why Step-Size Control?

(5/5_w, 10)-ES, 11 runs



with optimal step-size σ

$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$$

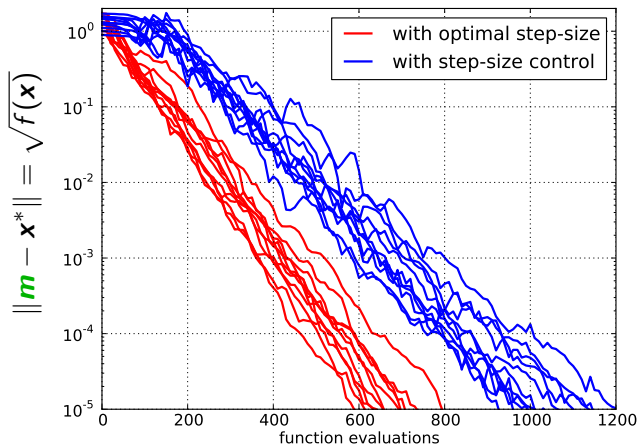
for $n = 10$

and

$$\mathbf{x}^0 \in [-0.2, 0.8]^n$$

Why Step-Size Control?

(5/5_w, 10)-ES, 2×11 runs



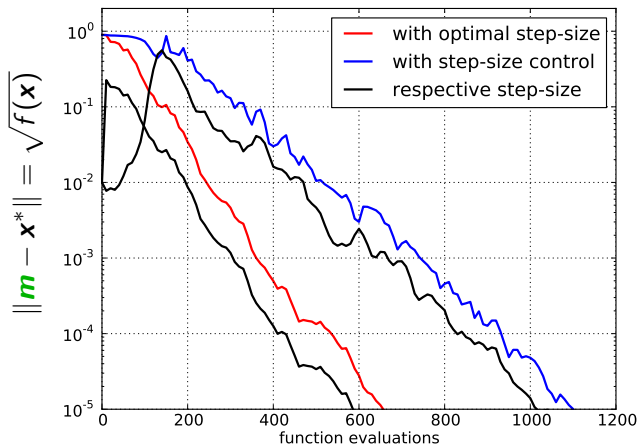
$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$$

for $n = 10$
and
 $\mathbf{x}^0 \in [-0.2, 0.8]^n$

with **optimal** versus **adaptive** step-size σ with too small initial σ

Why Step-Size Control?

(5/5_w, 10)-ES



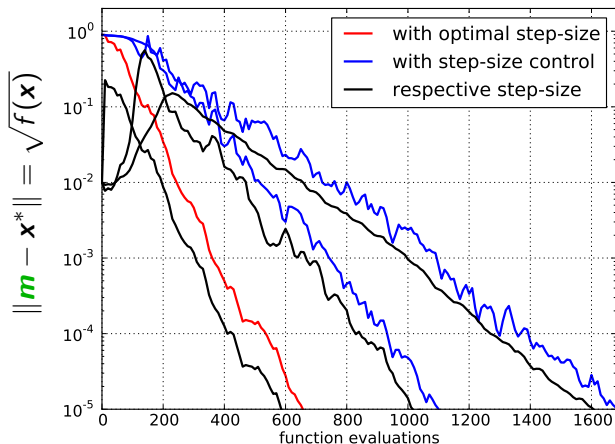
$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$$

for $n = 10$
and
 $\mathbf{x}^0 \in [-0.2, 0.8]^n$

comparing number of f -evals to reach $\|m\| = 10^{-5}$: $\frac{1100-100}{650} \approx 1.5$

Why Step-Size Control?

(5/5_w, 10)-ES



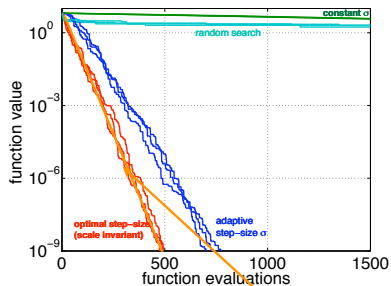
$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$$

for $n = 10$
and
 $\mathbf{x}^0 \in [-0.2, 0.8]^n$

comparing optimal versus default damping parameter d_σ :

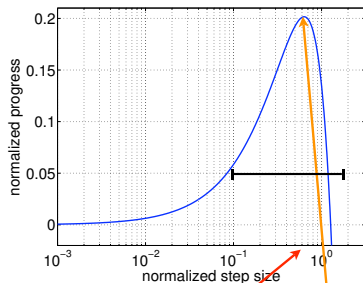
$$\frac{1700}{1100} \approx 1.5$$

Why Step-Size Control?



$$\sigma \leftarrow \sigma_{\text{opt}} \|\text{parent}\|$$

$$\frac{\varphi^*}{n}$$



σ_{opt}^*

φ^*

evolution window refers to the step-size interval (---) where reasonable performance is observed

Step-size control

Theory

- ▶ On well conditioned problem (sphere function $f(\mathbf{x}) = \|\mathbf{x}\|^2$) step-size adaptation should allow to reach (close to) optimal convergence rates
need to be able to solve optimally simple scenario (linear function, sphere function) that quite often (always?) need to be solved when addressing a real-world problem
- ▶ Is it possible to quantify optimal convergence rate for step-size adaptive ESs?

Lower bound for convergence

Exemplified on (1+1)-ES

Consider a (1+1)-ES with any **step-size adaptation** mechanism

(1+1)-ES with adaptive step-size

Iteration k :

$$\underbrace{\tilde{\mathbf{X}}_{k+1}}_{\text{offspring}} = \underbrace{\mathbf{X}_k}_{\text{parent}} + \underbrace{\sigma_k}_{\text{step-size}} \mathcal{N}_{k+1} \text{ with } (\mathcal{N}_k)_k \text{ i.i.d. } \sim \mathcal{N}(0, \mathbf{I})$$

$$\mathbf{X}_{k+1} = \begin{cases} \tilde{\mathbf{X}}_{k+1} & \text{if } f(\tilde{\mathbf{X}}_{k+1}) \leq f(\mathbf{X}_k) \\ \mathbf{X}_k & \text{otherwise} \end{cases}$$

Lower bound for convergence (II)

Exemplify on (1+1)-ES

Theorem

For any objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, for any $y^* \in \mathbb{R}^n$

$$E[\ln \|\mathbf{X}_{k+1} - y^*\|] \geq E[\ln \|\mathbf{X}_k - y^*\|] \underbrace{- \tau}_{\text{lower bound}}$$

where

$$\tau = \max_{\sigma \in \mathbb{R}^+} E[\ln \|\underbrace{\mathbf{e}_1}_{(1,0,\dots,0)} + \sigma \mathcal{N}\|]$$

$\underbrace{\hspace{10em}}_{=:\varphi(\sigma)}$

"Tight" lower bound

Theorem

Lower bound reached on the sphere function $f(\mathbf{x}) = g(\|\mathbf{x} - y^\|)$, (with $g : \mathbb{R} \rightarrow \mathbb{R}$, increasing mapping) for step-size proportional to the distance to the optimum where $\sigma_k = \sigma \|\mathbf{x} - y^*\|$ with $\sigma := \sigma_{\text{opt}}$ such that $\varphi(\sigma_{\text{opt}}) = \tau$.*

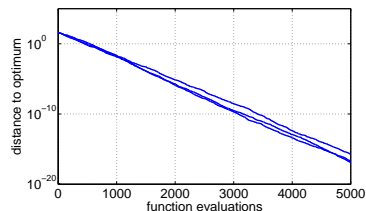
(Log)-Linear convergence of scale-invariant step-size ES

Theorem

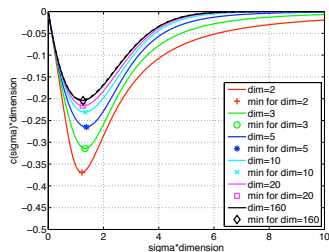
The (1+1)-ES with step-size proportional to the distance to the optimum $\sigma_k = \sigma \|\mathbf{x}\|$ converges (log)-linearly on the sphere function $f(\mathbf{x}) = g(\|\mathbf{x}\|)$, (with $g : \mathbb{R} \rightarrow \mathbb{R}$, increasing mapping) in the sense

$$\frac{1}{k} \ln \frac{\|\mathbf{X}_k\|}{\|\mathbf{X}_0\|} \xrightarrow{k \rightarrow \infty} -\varphi(\sigma) =: \text{CR}_{(1+1)}(\sigma)$$

almost surely.



$n = 20$ and $\sigma = 0.6/n$



Asymptotic results

When $n \rightarrow \infty$

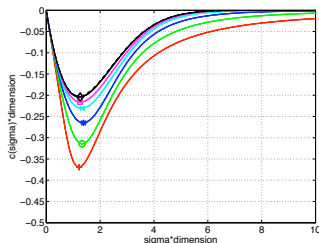
Theorem

Let $\sigma > 0$, the convergence rate of the (1+1)-ES with scale-invariant step-size on spherical functions satisfies at the limit

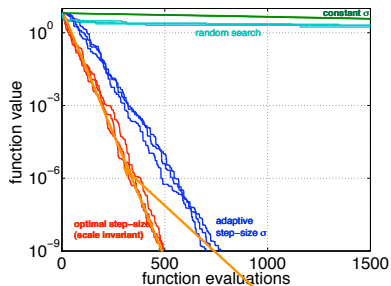
$$\lim_{n \rightarrow \infty} n \times \text{CR}_{(1+1)} \left(\frac{\sigma}{n} \right) = \frac{-\sigma}{\sqrt{2\pi}} \exp \left(-\frac{\sigma^2}{8} \right) + \frac{\sigma^2}{2} \Phi \left(-\frac{\sigma}{2} \right)$$

where Φ is the cumulative distribution of a normal distribution.

optimal convergence rate decreases to zero like $\frac{1}{n}$

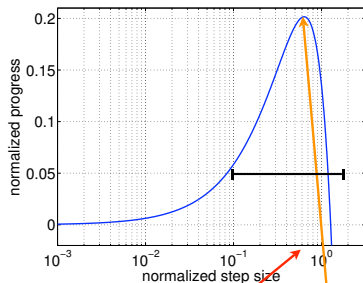


Summary of theory results



$$\sigma \leftarrow \sigma_{\text{opt}} \|\text{parent}\|$$

$$\frac{\varphi^*}{n}$$



σ_{opt}^*

φ^*

evolution window refers to the step-size interval (---) where reasonable performance is observed

Problem Statement

Black Box Optimization and Its Difficulties

Non-Separable Problems

Ill-Conditioned Problems

Stochastic search algorithms - basics

A Search Template

A Natural Search Distribution: the Normal Distribution

Adaptation of Distribution Parameters: What to Achieve?

Adaptive Evolution Strategies

Mean Vector Adaptation

Step-size control

Theory

Algorithms

Covariance Matrix Adaptation

Rank-One Update

Cumulation—the Evolution Path

Rank- μ Update

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 one, according to the objective function value, is selected
 - simplified “global” self-adaptation
- ▶ **path length control^d** (Cumulative Step-size Adaptation, CSA)^e, applied with “,”-selection

^aRechenberg 1973, *Evolutionsstrategie, Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog

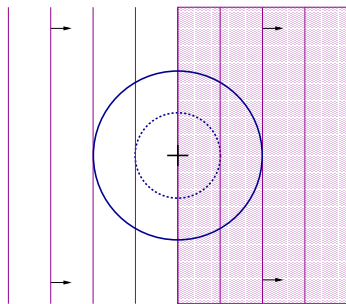
^bSchumer and Steiglitz 1968. Adaptive step size random search. *IEEE TAC*

^cSchwefel 1981, *Numerical Optimization of Computer Models*, Wiley

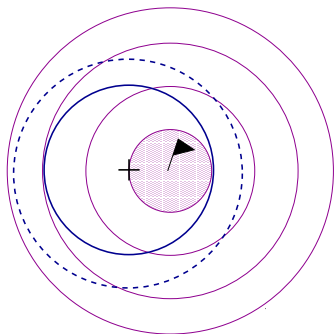
^dHansen & Ostermeier 2001, Completely Derandomized Self-Adaptation in Evolution Strategies, *Evol. Comput.* 9(2)

^eOstermeier et al 1994, Step-size adaptation based on non-local use of selection information, *PPSN*

One-fifth success rule

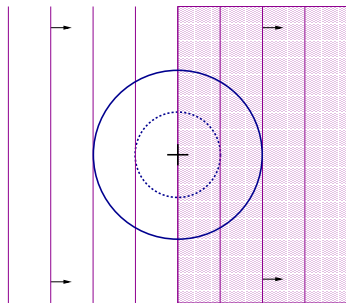


increase σ



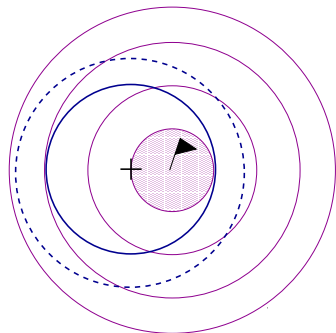
decrease σ

One-fifth success rule



Probability of success (p_s)

$1/2$



Probability of success (p_s)

"too small"

$1/5$

One-fifth success rule

p_s : # of successful offspring / # offspring (per iteration)

$$\sigma \leftarrow \sigma \times \exp\left(\frac{1}{3} \times \frac{p_s - p_{\text{target}}}{1 - p_{\text{target}}}\right)$$

Increase σ if $p_s > p_{\text{target}}$
Decrease σ if $p_s < p_{\text{target}}$

(1 + 1)-ES

$$p_{\text{target}} = 1/5$$

IF *offspring better parent*

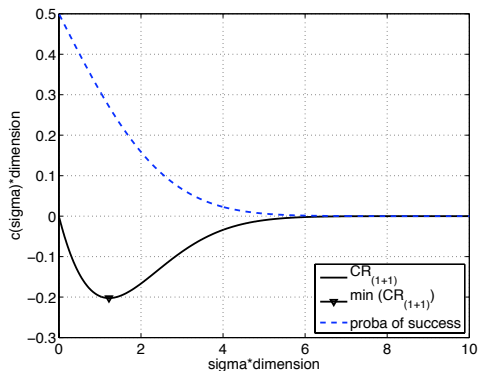
$$p_s = 1, \sigma \leftarrow \sigma \times \exp(1/3)$$

ELSE

$$p_s = 0, \sigma \leftarrow \sigma / \exp(1/3)^{1/4}$$

Why 1/5?

Asymptotic convergence rate and probability of success of scale-invariant step-size (1+1)-ES



sphere - asymptotic results, i.e. $n = \infty$ (see slides before)

1/5 trade-off of optimal probability of success on the sphere and
corridor

Path Length Control (CSA)

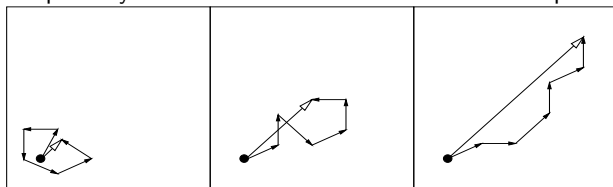
The Concept of Cumulative Step-Size Adaptation

$$x_i = m + \sigma y_i$$

$$m \leftarrow m + \sigma y_w$$

Measure the length of the *evolution path*

the pathway of the mean vector m in the iteration sequence



↓
decrease σ

↓
increase σ

Path Length Control (CSA)

The Equations

Sampling of solutions, notations as on slide “The $(\mu/\mu, \lambda)$ -ES - Update of the mean vector” with \mathbf{C} equal to the identity.

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

Path Length Control (CSA)

The Equations

Sampling of solutions, notations as on slide “The $(\mu/\mu, \lambda)$ -ES - Update of the mean vector” with \mathbf{C} equal to the identity.

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

$$\mathbf{m} \leftarrow \mathbf{m} + \sigma \mathbf{y}_w \quad \text{where } \mathbf{y}_w = \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda} \quad \text{update mean}$$

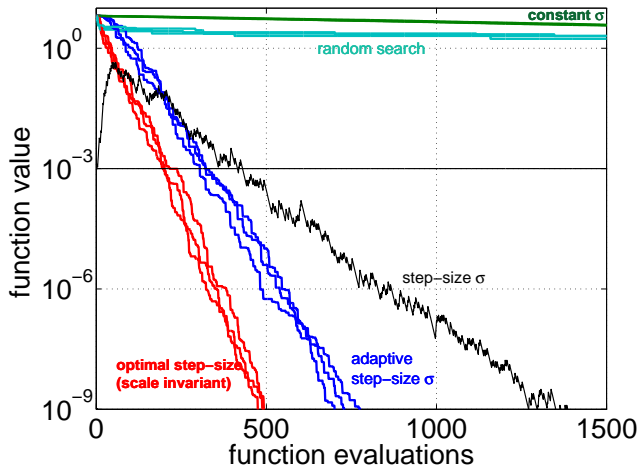
$$\mathbf{p}_\sigma \leftarrow (1 - c_\sigma) \mathbf{p}_\sigma + \underbrace{\sqrt{1 - (1 - c_\sigma)^2}}_{\text{accounts for } 1 - c_\sigma} \underbrace{\sqrt{\mu_w}}_{\text{accounts for } w_i} \mathbf{y}_w$$

$$\sigma \leftarrow \sigma \times \underbrace{\exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1\right)\right)}_{>1 \iff \|\mathbf{p}_\sigma\| \text{ is greater than its expectation}} \quad \text{update step-size}$$

Step-size adaptation

What is achieved

(1 + 1)-ES with one-fifth success rule (blue)



$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$$

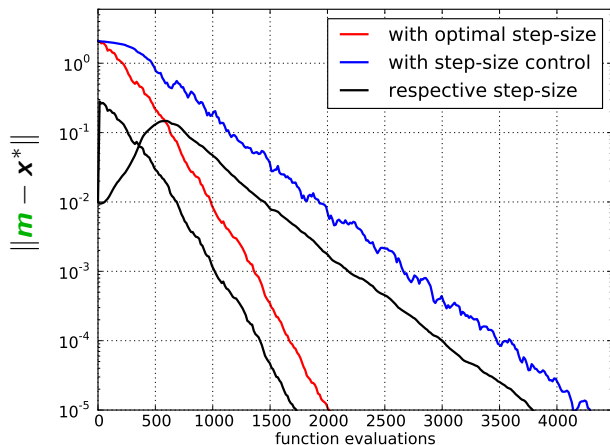
in $[-0.2, 0.8]^n$
for $n = 10$

Linear convergence

Step-size adaptation

What is achieved

(5/5, 10)-CSA-ES, default parameters



$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$$

in $[-0.2, 0.8]^n$
for $n = 30$

Problem Statement

Black Box Optimization and Its Difficulties

Non-Separable Problems

Ill-Conditioned Problems

Stochastic search algorithms - basics

A Search Template

A Natural Search Distribution: the Normal Distribution

Adaptation of Distribution Parameters: What to Achieve?

Adaptive Evolution Strategies

Mean Vector Adaptation

Step-size control

Theory

Algorithms

Covariance Matrix Adaptation

Rank-One Update

Cumulation—the Evolution Path

Rank- μ Update

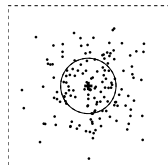
Evolution Strategies

Recalling

New search points are sampled normally distributed

$$\mathbf{x}_i \sim \mathbf{m} + \sigma \mathcal{N}_i(\mathbf{0}, \mathbf{C}) \quad \text{for } i = 1, \dots, \lambda$$

as perturbations of \mathbf{m} , where $\mathbf{x}_i, \mathbf{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$,
 $\mathbf{C} \in \mathbb{R}^{n \times n}$



where

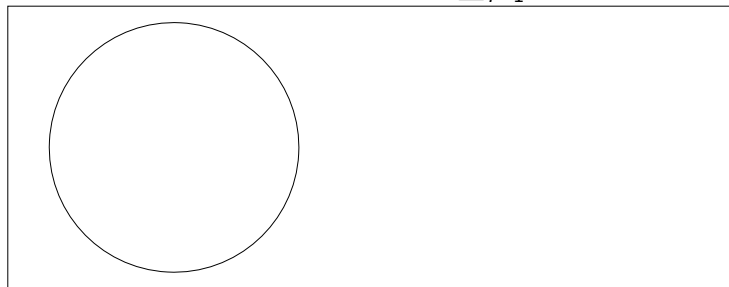
- ▶ the **mean** vector $\mathbf{m} \in \mathbb{R}^n$ represents the favorite solution
- ▶ the so-called **step-size** $\sigma \in \mathbb{R}_+$ controls the *step length*
- ▶ the **covariance matrix** $\mathbf{C} \in \mathbb{R}^{n \times n}$ determines the **shape** of the distribution ellipsoid

The remaining question is how to update \mathbf{C} .

Covariance Matrix Adaptation

Rank-One Update

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

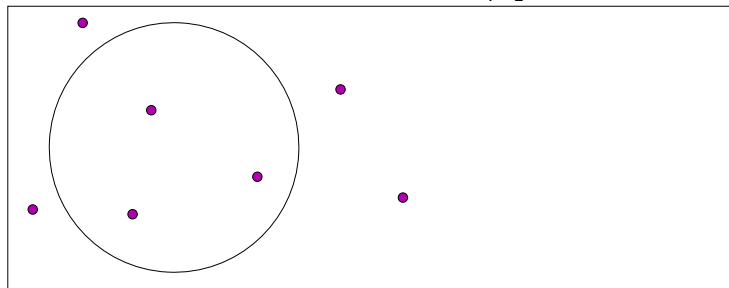


initial distribution, $\mathbf{C} = \mathbf{I}$

Covariance Matrix Adaptation

Rank-One Update

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

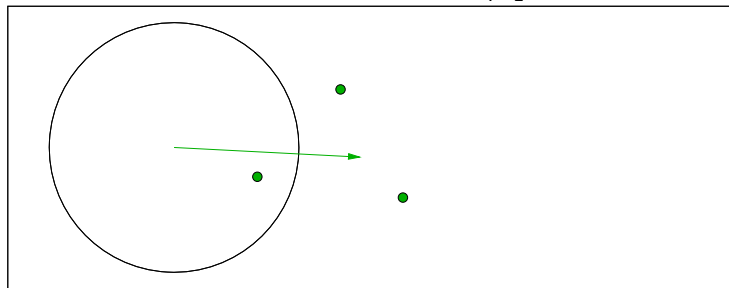


initial distribution, $\mathbf{C} = \mathbf{I}$

Covariance Matrix Adaptation

Rank-One Update

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

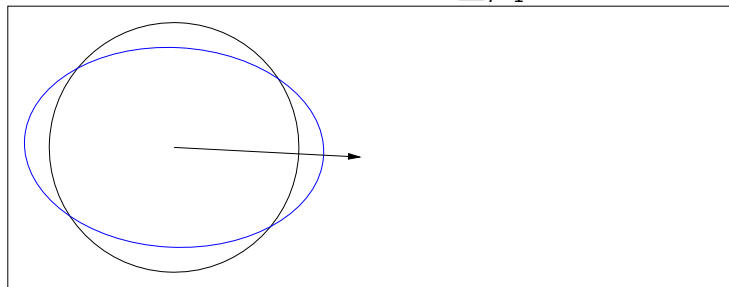


\mathbf{y}_w , movement of the population mean \mathbf{m} (disregarding σ)

Covariance Matrix Adaptation

Rank-One Update

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



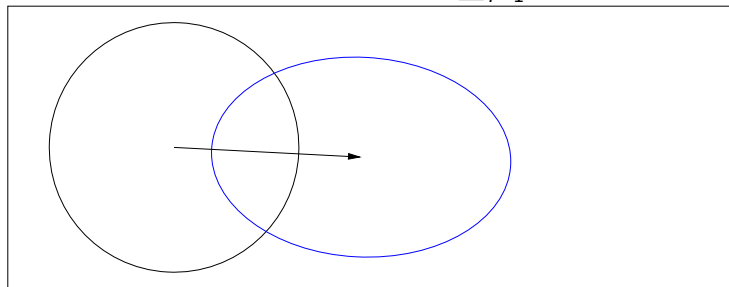
mixture of distribution \mathbf{C} and step \mathbf{y}_w ,

$$\mathbf{C} \leftarrow 0.8 \times \mathbf{C} + 0.2 \times \mathbf{y}_w \mathbf{y}_w^T$$

Covariance Matrix Adaptation

Rank-One Update

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

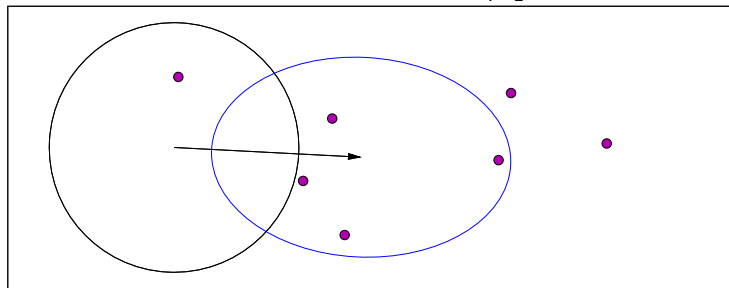


new distribution (disregarding σ)

Covariance Matrix Adaptation

Rank-One Update

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

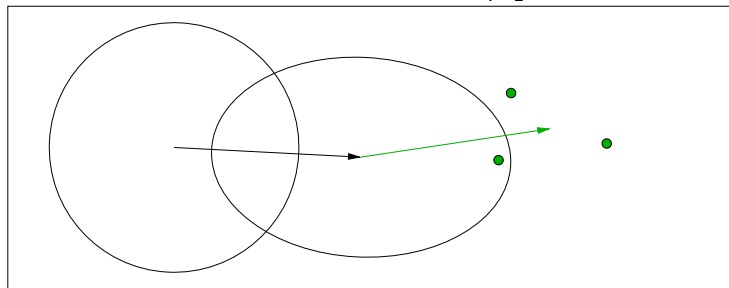


new distribution (disregarding σ)

Covariance Matrix Adaptation

Rank-One Update

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

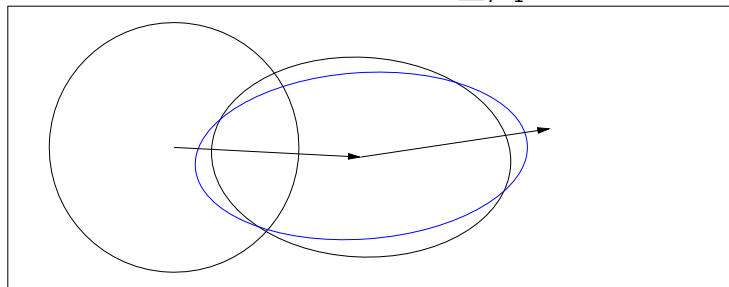


movement of the population mean \mathbf{m}

Covariance Matrix Adaptation

Rank-One Update

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



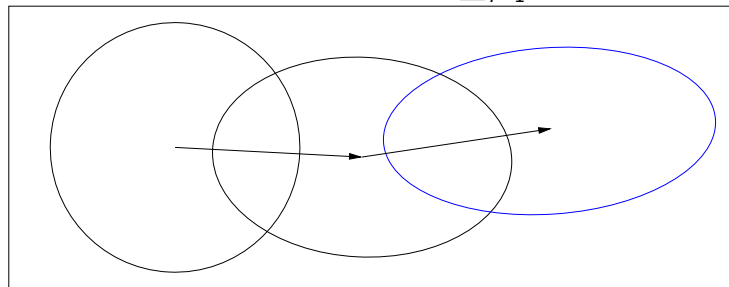
mixture of distribution \mathbf{C} and step \mathbf{y}_w ,

$$\mathbf{C} \leftarrow 0.8 \times \mathbf{C} + 0.2 \times \mathbf{y}_w \mathbf{y}_w^T$$

Covariance Matrix Adaptation

Rank-One Update

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



new distribution,

$$\mathbf{C} \leftarrow 0.8 \times \mathbf{C} + 0.2 \times \mathbf{y}_w \mathbf{y}_w^T$$

the ruling principle: the adaptation increases the likelihood of successful steps, \mathbf{y}_w , to appear again

Covariance Matrix Adaptation

Rank-One Update

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

While not terminate

$$\mathbf{x}_i = \mathbf{m} + \sigma \mathbf{y}_i, \quad \mathbf{y}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}),$$

$$\mathbf{m} \leftarrow \mathbf{m} + \sigma \mathbf{y}_w \quad \text{where } \mathbf{y}_w = \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}$$

$$\mathbf{C} \leftarrow (1 - c_{\text{cov}})\mathbf{C} + c_{\text{cov}}\mu_w \underbrace{\mathbf{y}_w \mathbf{y}_w^T}_{\text{rank-one}} \quad \text{where } \mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \geq 1$$

Problem Statement

Stochastic search algorithms - basics

Adaptive Evolution Strategies

- Mean Vector Adaptation

- Step-size control

- Covariance Matrix Adaptation

 - Rank-One Update

 - Cumulation—the Evolution Path

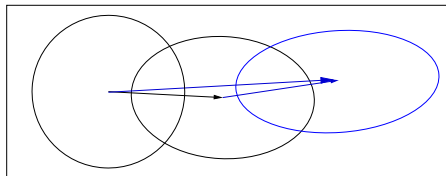
 - Rank- μ Update

Cumulation

The Evolution Path

Evolution Path

Conceptually, the evolution path is the **search path** the strategy takes **over a number of iteration steps**. It can be expressed as a sum of consecutive *steps* of the mean m .



An exponentially weighted sum of steps y_w is used

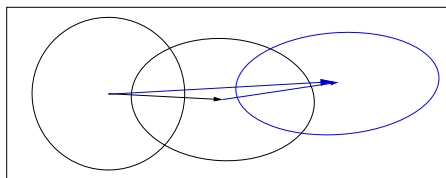
$$p_c \propto \sum_{i=0}^g \underbrace{(1 - c_c)^{g-i}}_{\text{exponentially fading weights}} y_w^{(i)}$$

Cumulation

The Evolution Path

Evolution Path

Conceptually, the evolution path is the **search path** the strategy takes **over a number of iteration steps**. It can be expressed as a sum of consecutive *steps* of the mean m .



An exponentially weighted sum of steps y_w is used

$$p_c \propto \sum_{i=0}^g \underbrace{(1 - c_c)^{g-i}}_{\text{exponentially fading weights}} y_w^{(i)}$$

The recursive construction of the evolution path (cumulation):

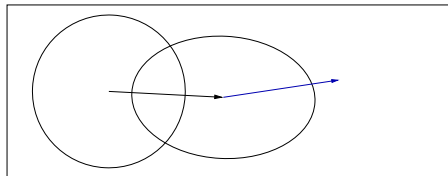
$$p_c \leftarrow \underbrace{(1 - c_c)}_{\text{decay factor}} p_c + \underbrace{\sqrt{1 - (1 - c_c)^2}}_{\text{normalization factor}} \sqrt{\mu_w} \underbrace{y_w}_{\text{input} = \frac{m - m_{\text{old}}}{\sigma}}$$

where $\mu_w = \frac{1}{\sum w_i^2}$, $c_c \ll 1$. **History information** is accumulated in the evolution path.

Cumulation

Utilizing the Evolution Path

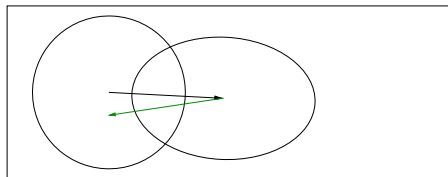
We used $\mathbf{y}_w \mathbf{y}_w^T$ for updating **C**. Because $\mathbf{y}_w \mathbf{y}_w^T = -\mathbf{y}_w (-\mathbf{y}_w)^T$ the sign of \mathbf{y}_w is lost.



Cumulation

Utilizing the Evolution Path

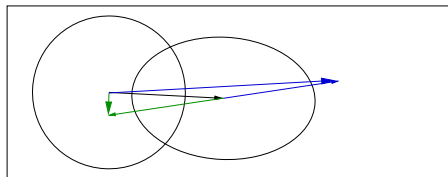
We used $\mathbf{y}_w \mathbf{y}_w^T$ for updating \mathbf{C} . Because $\mathbf{y}_w \mathbf{y}_w^T = -\mathbf{y}_w (-\mathbf{y}_w)^T$ the sign of \mathbf{y}_w is lost.



Cumulation

Utilizing the Evolution Path

We used $\mathbf{y}_w \mathbf{y}_w^T$ for updating \mathbf{C} . Because $\mathbf{y}_w \mathbf{y}_w^T = -\mathbf{y}_w (-\mathbf{y}_w)^T$ the sign of \mathbf{y}_w is lost.



The sign information is (re-)introduced by using the *evolution path*.

$$\begin{aligned} \mathbf{p}_c &\leftarrow \underbrace{(1 - c_c)}_{\text{decay factor}} \mathbf{p}_c + \underbrace{\sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w}}_{\text{normalization factor}} \mathbf{y}_w \\ \mathbf{C} &\leftarrow (1 - c_{\text{cov}}) \mathbf{C} + c_{\text{cov}} \underbrace{\mathbf{p}_c \mathbf{p}_c^T}_{\text{rank-one}} \end{aligned}$$

where $\mu_w = \frac{1}{\sum w_i^2}$, $c_c \ll 1$.

Using an **evolution path** for the **rank-one update** of the covariance matrix reduces the number of function evaluations to adapt to a straight ridge **from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$** .⁽³⁾

The overall model complexity is n^2 but important parts of the model can be learned in time of order n

³Hansen, Müller and Koumoutsakos 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1), pp. 1-18

Rank- μ Update

$$\begin{aligned} \mathbf{x}_i &= \mathbf{m} + \sigma \mathbf{y}_i, & \mathbf{y}_i &\sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), \\ \mathbf{m} &\leftarrow \mathbf{m} + \sigma \mathbf{y}_w, & \mathbf{y}_w &= \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda} \end{aligned}$$

The rank- μ update extends the update rule for large population sizes λ using $\mu > 1$ vectors to update \mathbf{C} at each iteration step.

Rank- μ Update

$$\begin{aligned} \mathbf{x}_i &= \mathbf{m} + \sigma \mathbf{y}_i, & \mathbf{y}_i &\sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), \\ \mathbf{m} &\leftarrow \mathbf{m} + \sigma \mathbf{y}_w, & \mathbf{y}_w &= \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda} \end{aligned}$$

The rank- μ update extends the update rule for large population sizes λ using $\mu > 1$ vectors to update \mathbf{C} at each iteration step.

The matrix

$$\mathbf{C}_{\mu} = \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^T$$

computes a weighted mean of the outer products of the best μ steps and has rank $\min(\mu, n)$ with probability one.

Rank- μ Update

$$\begin{aligned} \mathbf{x}_i &= \mathbf{m} + \sigma \mathbf{y}_i, & \mathbf{y}_i &\sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), \\ \mathbf{m} &\leftarrow \mathbf{m} + \sigma \mathbf{y}_w, & \mathbf{y}_w &= \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda} \end{aligned}$$

The rank- μ update extends the update rule for large population sizes λ using $\mu > 1$ vectors to update \mathbf{C} at each iteration step.

The matrix

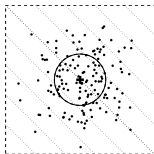
$$\mathbf{C}_{\mu} = \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^T$$

computes a weighted mean of the outer products of the best μ steps and has rank $\min(\mu, n)$ with probability one.

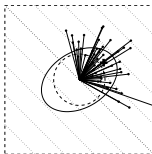
The rank- μ update then reads

$$\mathbf{C} \leftarrow (1 - c_{\text{cov}}) \mathbf{C} + c_{\text{cov}} \mathbf{C}_{\mu}$$

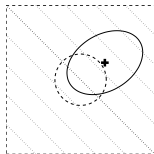
where $c_{\text{cov}} \approx \mu_w / n^2$ and $c_{\text{cov}} \leq 1$.



$$x_i = m + \sigma y_i, \quad y_i \sim \mathcal{N}(0, \mathbf{C})$$



$$\begin{aligned} \mathbf{C}_\mu &= \frac{1}{\mu} \sum y_{i:\lambda} y_{i:\lambda}^T \\ \mathbf{C} &\leftarrow \frac{1}{(1-\alpha)} \times \mathbf{C} + \alpha \times \mathbf{C}_\mu \end{aligned}$$



$$m_{\text{new}} \leftarrow m + \frac{1}{\mu} \sum y_{i:\lambda}$$

sampling of
 $\lambda = 150$ solutions
 where $\mathbf{C} = \mathbf{I}$ and
 $\sigma = 1$

calculating \mathbf{C} where
 $\mu = 50$, $w_1 = \dots =$
 $w_\mu = \frac{1}{\mu}$, and
 $\mathbf{C}_{\text{cov}} = 1$

new distribution

The rank- μ update

- ▶ increases the possible learning rate in large populations
roughly from $2/n^2$ to μ_w/n^2
- ▶ can reduce the number of necessary iterations roughly from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ ⁽⁴⁾
given $\mu_w \propto \lambda \propto n$

Therefore the rank- μ update is the primary mechanism whenever a large population size is used

say $\lambda \geq 3n + 10$

⁴Hansen, Müller, and Koumoutsakos 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1), pp. 1-18

The rank- μ update

- ▶ increases the possible learning rate in large populations
roughly from $2/n^2$ to μ_w/n^2
- ▶ can reduce the number of necessary iterations roughly from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ ⁽⁴⁾
given $\mu_w \propto \lambda \propto n$

Therefore the rank- μ update is the primary mechanism whenever a large population size is used

say $\lambda \geq 3n + 10$

The rank-one update

- ▶ uses the evolution path and reduces the number of necessary function evaluations to learn straight ridges from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.

⁴Hansen, Müller, and Koumoutsakos 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1), pp. 1-18

The rank- μ update

- ▶ increases the possible learning rate in large populations
roughly from $2/n^2$ to μ_w/n^2
- ▶ can reduce the number of necessary iterations roughly from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ ⁽⁴⁾
given $\mu_w \propto \lambda \propto n$

Therefore the rank- μ update is the primary mechanism whenever a large population size is used

say $\lambda \geq 3n + 10$

The rank-one update

- ▶ uses the evolution path and reduces the number of necessary function evaluations to learn straight ridges from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.

Rank-one update and rank- μ update can be combined

⁴Hansen, Müller, and Koumoutsakos 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1), pp. 1-18

Summary of Equations

The Covariance Matrix Adaptation Evolution Strategy

Input: $\mathbf{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ

Initialize: $\mathbf{C} = \mathbf{I}$, and $\mathbf{p}_c = \mathbf{0}$, $\mathbf{p}_\sigma = \mathbf{0}$,

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 $w_{i=1\dots\lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^\mu w_i^2} \approx 0.3 \lambda$

While not terminate

$\mathbf{x}_i = \mathbf{m} + \sigma \mathbf{y}_i$, $\mathbf{y}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C})$, for $i = 1, \dots, \lambda$ sampling

$\mathbf{m} \leftarrow \sum_{i=1}^\mu w_i \mathbf{x}_{i:\lambda} = \mathbf{m} + \sigma \mathbf{y}_w$ where $\mathbf{y}_w = \sum_{i=1}^\mu w_i \mathbf{y}_{i:\lambda}$ update mean

$\mathbf{p}_c \leftarrow (1 - c_c) \mathbf{p}_c + \mathbb{1}_{\{\|\mathbf{p}_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w} \mathbf{y}_w$ cumulation for \mathbf{C}

$\mathbf{p}_\sigma \leftarrow (1 - c_\sigma) \mathbf{p}_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w} \mathbf{C}^{-\frac{1}{2}} \mathbf{y}_w$ cumulation for σ

$\mathbf{C} \leftarrow (1 - c_1 - c_\mu) \mathbf{C} + c_1 \mathbf{p}_c \mathbf{p}_c^T + c_\mu \sum_{i=1}^\mu w_i \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^T$ update \mathbf{C}

$\sigma \leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1\right)\right)$ update of σ

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

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{H} \mathbf{x}$$

to the sphere model

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{x}$$

e.g. $f(\mathbf{x}) = \sum_{i=1}^n 10^{6 \frac{i-1}{n-1}} x_i^2$

without use of derivatives

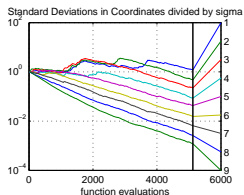
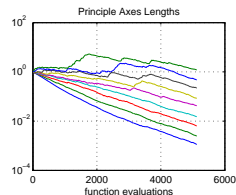
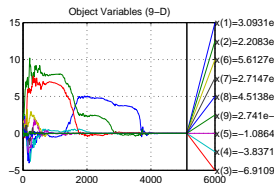
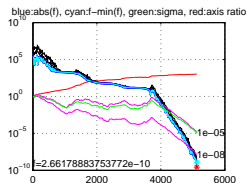
- ▶ lines of equal density align with lines of equal fitness

$$\mathbf{C} \propto \mathbf{H}^{-1}$$

in a stochastic sense

Experimentum Crucis (1)

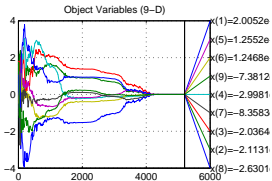
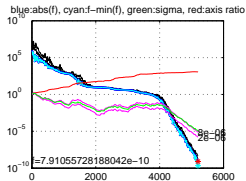
f convex quadratic, separable



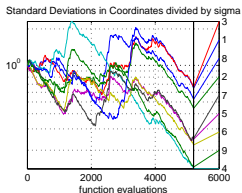
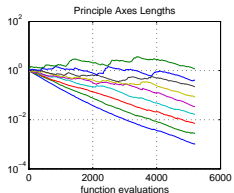
$$f(\mathbf{x}) = \sum_{i=1}^n 10^{\alpha \frac{i-1}{n-1}} x_i^2, \alpha = 6$$

Experimentum Crucis (2)

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



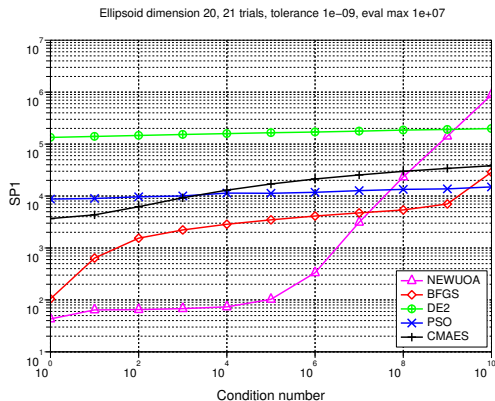
$C \propto H^{-1}$ for all
 g, H



$f(\mathbf{x}) = g(\mathbf{x}^T \mathbf{H} \mathbf{x})$, $g: \mathbb{R} \rightarrow \mathbb{R}$ strictly increasing

Comparison to BFGS, NEWUOA, PSO and DE

f convex quadratic, separable with varying condition number α



SP1 = average number of objective function evaluations⁵ to reach the target function value of $g^{-1}(10^{-9})$

BFGS (Broyden et al 1970)

NEWUOA (Powell 2004)

DE (Storn & Price 1996)

PSO (Kennedy & Eberhart 1995)

CMA-ES (Hansen & Ostermeier 2001)

$f(x) = g(x^T H x)$ with

H diagonal

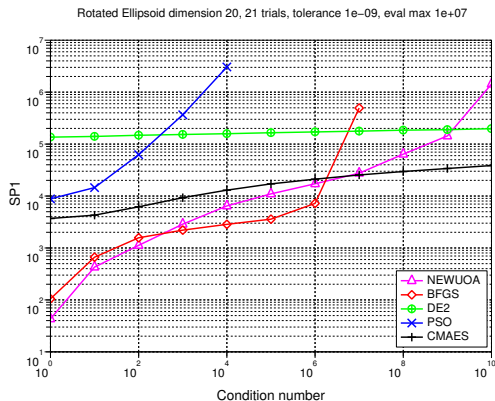
g identity (for **BFGS** and **NEWUOA**)

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

⁵ 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 α



SP1 = average number of objective function evaluations⁶ to reach the target function value of $g^{-1}(10^{-9})$

BFGS (Broyden et al 1970)

NEWUOA (Powell 2004)

DE (Storn & Price 1996)

PSO (Kennedy & Eberhart 1995)

CMA-ES (Hansen & Ostermeier 2001)

$f(x) = g(x^T H x)$ with

H full

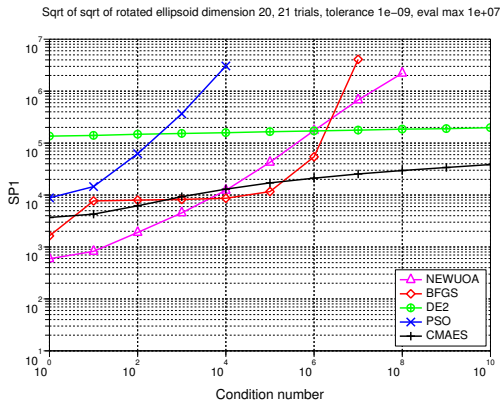
g identity (for **BFGS** and **NEWUOA**)

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

⁶ Auger et.al. (2009): Experimental comparisons of derivative free optimization algorithms, SEA

Comparison to BFGS, NEWUOA, PSO and DE

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



BFGS (Broyden et al 1970)

NEWUOA (Powell 2004)

DE (Storn & Price 1996)

PSO (Kennedy & Eberhart 1995)

CMA-ES (Hansen & Ostermeier 2001)

$f(x) = g(x^T H x)$ with

H full

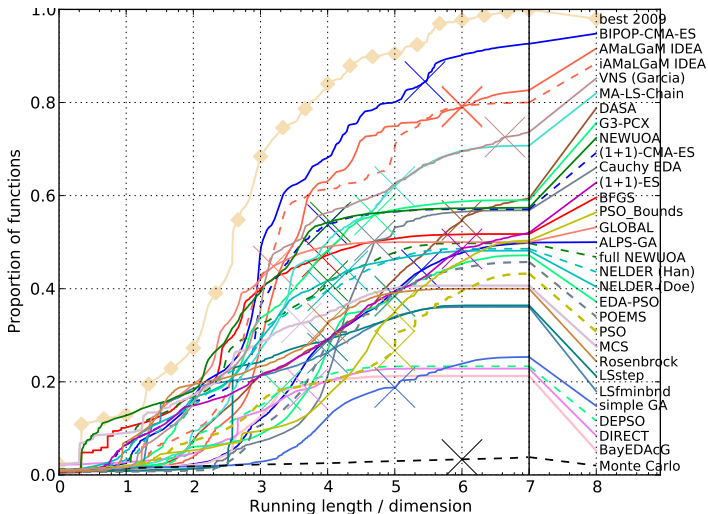
$g : x \mapsto x^{1/4}$ (for **BFGS** and **NEWUOA**)

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

SP1 = average number of objective function evaluations⁷ to reach the target function value of $g^{-1}(10^{-9})$

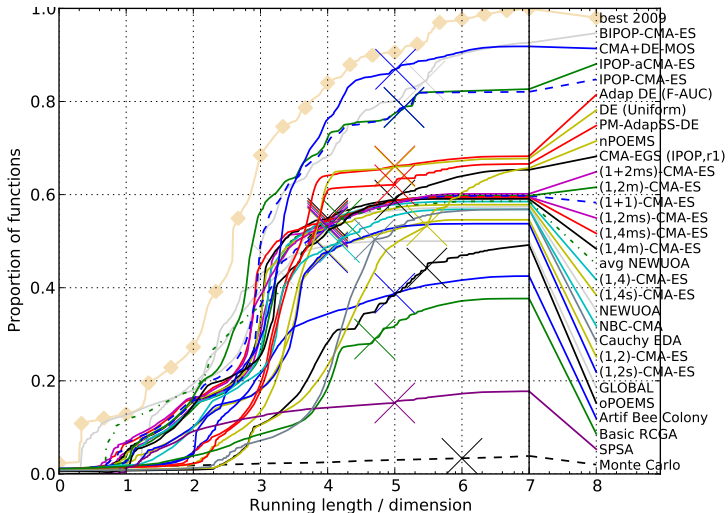
Comparison during BBOB at GECCO 2009

24 functions and 31 algorithms in 20-D



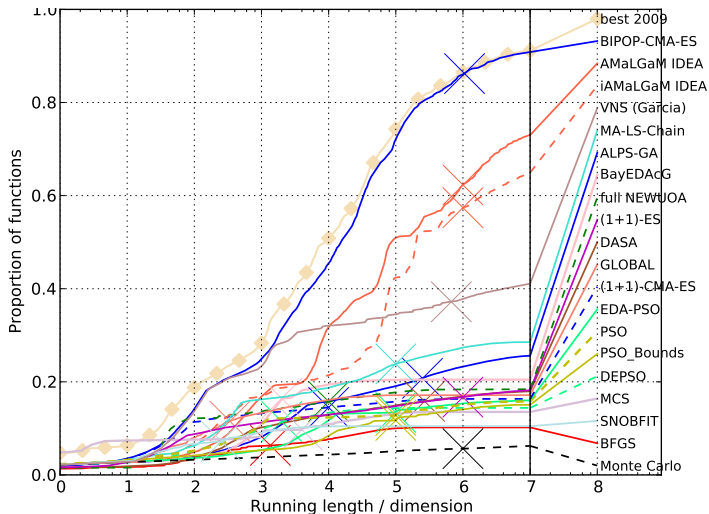
Comparison during BBOB at GECCO 2010

24 functions and 20+ algorithms in 20-D



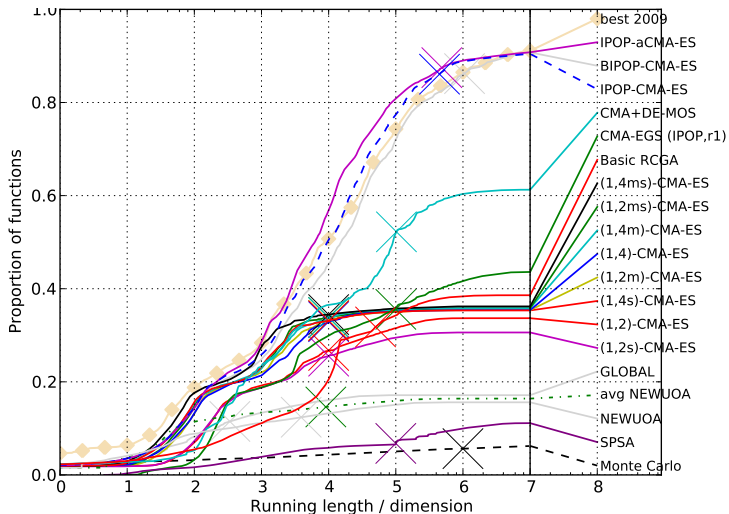
Comparison during BBOB at GECCO 2009

30 noisy functions and 20 algorithms in 20-D



Comparison during BBOB at GECCO 2010

30 noisy functions and 10+ algorithms in 20-D



Problem Statement

Stochastic search algorithms - basics

Adaptive Evolution Strategies

- Mean Vector Adaptation

- Step-size control

- Covariance Matrix Adaptation

 - Rank-One Update

 - Cumulation—the Evolution Path

 - Rank- μ Update

The Continuous Search Problem

Difficulties of a non-linear optimization problem are

- ▶ dimensionality and non-separability
demands to exploit problem structure, e.g. neighborhood
- ▶ ill-conditioning
demands to acquire a second order model
- ▶ ruggedness
demands a non-local (stochastic?) approach

Approach: population based stochastic search, coordinate system independent and with second order estimations (covariances)

Main Features of (CMA) Evolution Strategies

1. Multivariate normal distribution to generate new search points
follows the maximum entropy principle
2. Rank-based selection
implies invariance, same performance on $g(f(x))$ for any increasing g
more invariance properties are featured
3. Step-size control facilitates fast (log-linear) convergence
based on an evolution path (a non-local trajectory)
4. *Covariance matrix adaptation (CMA)* increases the likelihood of previously successful steps and can improve performance by orders of magnitude

$\mathbf{C} \propto \mathbf{H}^{-1} \iff$ adapts a variable metric
 \iff new (rotated) problem representation
 $\implies f(x) = g(x^T \mathbf{H} x)$ reduces to $g(x^T x)$

Limitations

of CMA Evolution Strategies

- ▶ **internal CPU-time:** $10^{-8}n^2$ seconds per function evaluation on a 2GHz PC, tweaks are available
100 000 *f*-evaluations in 1000-D take 1/4 hours
internal CPU-time
- ▶ better methods are presumably available in case of
 - ▶ partly separable problems
 - ▶ specific problems, for example with cheap gradients
specific methods
 - ▶ small dimension ($n \ll 10$)
for example Nelder-Mead
 - ▶ small running times (number of *f*-evaluations $\ll 100n$)
model-based methods