Derivative Free Optimization

Anne Auger (Inria)* and Laurent Dumas (U. Versailles)

AMS Master - Optimization Paris-Saclay Master

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

2016 — 2017

Course Organization

Part I - Stochastic Methods

taught by Anne Auger

Friday 25/11, 02/12, 9/12, 16/12, 06/01 from 2pm to 5:15

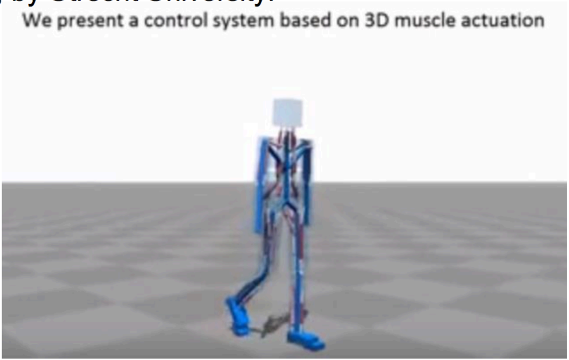
Part II - Deterministic Methods

taught by Laurent Dumas

Friday 13/01, 20/01, 27/01, 03/02, 10/02 from 2pm to 5:15

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=yci5FuI1ovk

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

$$f : \mathcal{X} \subseteq \mathbb{R}^n \to \mathbb{R}, \qquad \boldsymbol{x} \mapsto f(\boldsymbol{x})$$

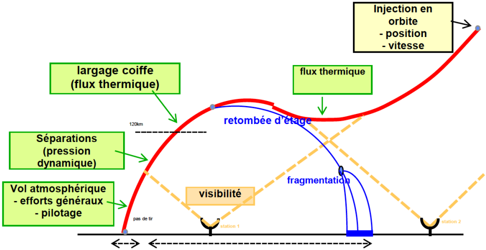
- Black Box scenario (direct search scenario)

  - 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

Optimization of the Design of a Launcher

Example of a black-box problem

Poppy



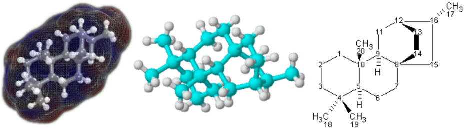
- Scenario: multi-stage launcher brings a satellite into orbit
- Minimize the overall cost of a launch
- Parameters: propellant mass of each stage / diameter of each stage / flux of each engine / parameters of the command law

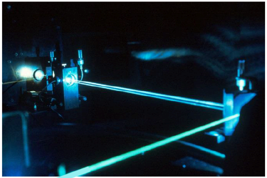
23 continuous parameters to optimize

+ constraints

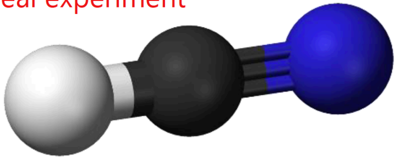
Control of the Alignement of Molecules

Example of a black-box problem (II)





Objective function:
via numerical simulation
or a real experiment



possible application in drug design

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

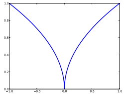
- dimensionality (size of search space)

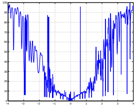
  (considerably) larger than three

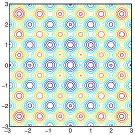
- non-separability

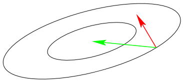
  dependencies between the objective variables

- ill-conditioning









gradient direction Newton directio

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.

Separable Problems

Definition (Separable Problem)

A function f is separable if

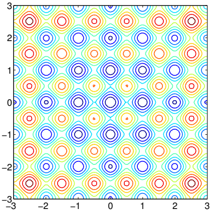
$$\arg\min_{(x_1,\ldots,x_n)} f(x_1,\ldots,x_n) = \left(\arg\min_{x_1} f(x_1,\ldots),\ldots,\arg\min_{x_n} f(\ldots,x_n)\right)$$

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



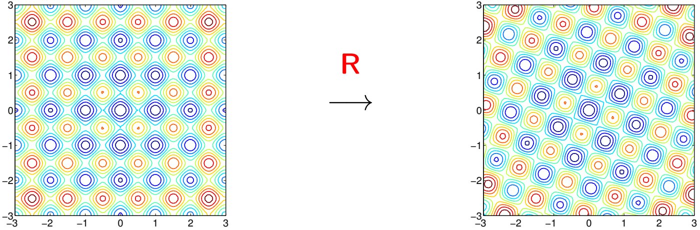
Non-Separable Problems

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



[1] Hansen, 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

[2] Salomon (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, $f : \boldsymbol{x} \mapsto \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x} = \frac{1}{2}\sum_i h_{i,i}\, x_i^2 + \frac{1}{2}\sum_{i\neq j} h_{i,j}\, x_i x_j$, with $\boldsymbol{H}$ positive, definite, symmetric matrix

  $\boldsymbol{H}$ is the Hessian matrix of f

- ill-conditioned means a high condition number of Hessian Matrix $\boldsymbol{H}$

$$\mathrm{cond}(\boldsymbol{H}) = \frac{\lambda_{\mathsf{max}}(\boldsymbol{H})}{\lambda_{\mathsf{min}}(\boldsymbol{H})}$$

Example / exercice

The level-sets of a function are defined as

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

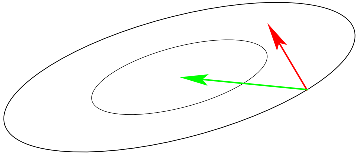
Consider the objective function $f(\boldsymbol{x}) = \frac{1}{2}(x_1^2 + 9x_2^2)$

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-conditionned Problems

consider the curvature of the level sets of a function

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



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

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

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 $\boldsymbol{\theta}$, set population size $\lambda \in \mathbb{N}$

While not terminate

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

Stochastic Search

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

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

While not terminate

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

Stochastic Search

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

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

While not terminate

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

Stochastic Search

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

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

While not terminate

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

Stochastic Search

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

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

While not terminate

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

Stochastic Search

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

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

While not terminate

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

Stochastic Search

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

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

While not terminate

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

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 $\boldsymbol{\theta}$, set population size $\lambda \in \mathbb{N}$

While not terminate

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

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

Exercice

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

Non-adaptive Algorithm

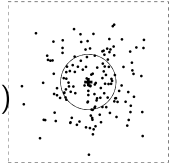
For the pure random search $P(\boldsymbol{x}|\boldsymbol{\theta})$ is independent of $\boldsymbol{\theta}$ (i.e. no $\boldsymbol{\theta}$ to be adapted): the algorithm is "blind"

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

Evolution Strategies

New search points are sampled normally distributed

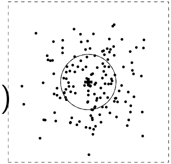
$$\boldsymbol{x}_i = \boldsymbol{m} + \sigma\, \boldsymbol{y}_i \qquad \text{for } i = 1, \ldots, \lambda \text{ with } \boldsymbol{y}_i \text{ i.i.d.} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \;:$$



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

Evolution Strategies

New search points are sampled normally distributed



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

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

where

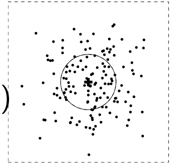
- the mean vector $\boldsymbol{m} \in \mathbb{R}^n$ represents the favorite solution
- the so-called step-size $\sigma \in \mathbb{R}_+$ controls the *step length*
- the covariance matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ determines the shape of the distribution ellipsoid

here, all new points are sampled with the same parameters

Evolution Strategies

New search points are sampled normally distributed

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



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

where

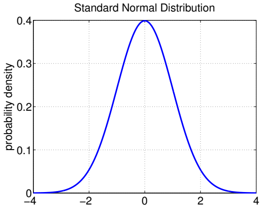
- the mean vector $\boldsymbol{m} \in \mathbb{R}^n$ represents the favorite solution
- the so-called step-size $\sigma \in \mathbb{R}_+$ controls the *step length*
- the covariance matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ determines the shape of the distribution ellipsoid

here, all new points are sampled with the same parameters

The question remains how to update $\boldsymbol{m}$, $\mathbf{C}$, 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 $\mathcal{N}(\boldsymbol{m}, \sigma^2)$

  (expected value, variance) $= (\boldsymbol{m}, \sigma^2)$

  density: $p_{\boldsymbol{m},\sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\boldsymbol{m})^2}{2\sigma^2}\right)$

- 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 $\boldsymbol{m} + \sigma\mathcal{N}(0,1) = \mathcal{N}(\boldsymbol{m}, \sigma^2)$

Normal Distribution

General case

A random variable following a 1-D normal distribution is determined by its mean value $\boldsymbol{m}$ 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 $\boldsymbol{X} = (X_1, \ldots, X_n)^T$ 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)

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

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

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

Σ is symmetric, positive definite

The Multi-Variate (n-Dimensional) Normal Distribution

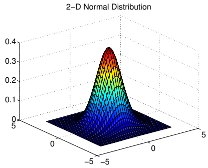
Any multi-variate normal distribution $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$ is uniquely determined by its mean value $\boldsymbol{m} \in \mathbb{R}^n$ and its symmetric positive definite $n \times n$ covariance matrix $\mathbf{C}$.

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

The mean value $\boldsymbol{m}$

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

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



The Multi-Variate (n-Dimensional) Normal Distribution

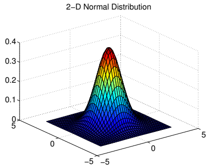
Any multi-variate normal distribution $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$ is uniquely determined by its mean value $\boldsymbol{m} \in \mathbb{R}^n$ and its symmetric positive definite $n \times n$ covariance matrix $\mathbf{C}$.

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

The mean value $\boldsymbol{m}$

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

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

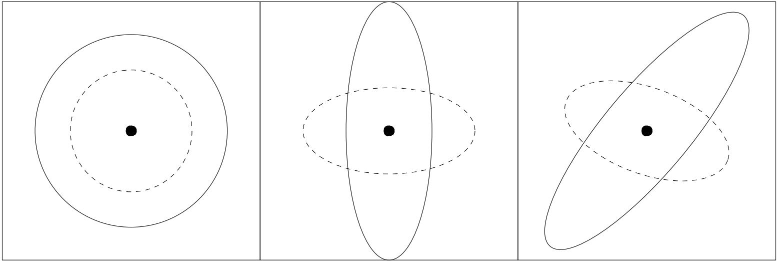


The covariance matrix $\mathbf{C}$

- determines the shape
- geometrical interpretation: any covariance matrix can be uniquely identified with the iso-density ellipsoid $\{\boldsymbol{x} \in \mathbb{R}^n \,|\, (\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}\mathbf{C}^{-1}(\boldsymbol{x}-\boldsymbol{m}) = 1\}$

... any covariance matrix can be uniquely identified with the iso-density ellipsoid $\{\boldsymbol{x} \in \mathbb{R}^n \,|\, (\boldsymbol{x} - \boldsymbol{m})^{\mathrm{T}} \mathbf{C}^{-1} (\boldsymbol{x} - \boldsymbol{m}) = 1\}$

Lines of Equal Density



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

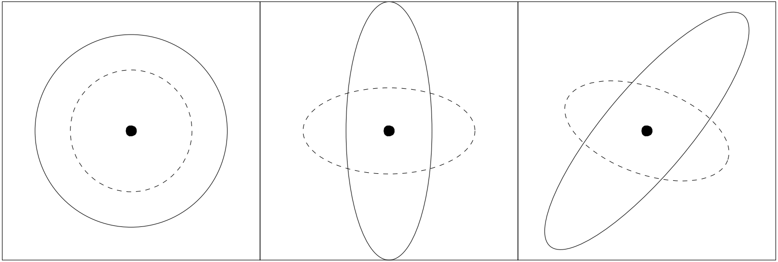
one degree of freedom σ

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 $\mathbf{A} \times \mathcal{N}(\mathbf{0}, \mathbf{I}) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{A}\mathbf{A}^{\mathrm{T}}\right)$ holds for all $\mathbf{A}$.

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

Lines of Equal Density



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

one degree of freedom σ

components are independent standard normally distributed

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

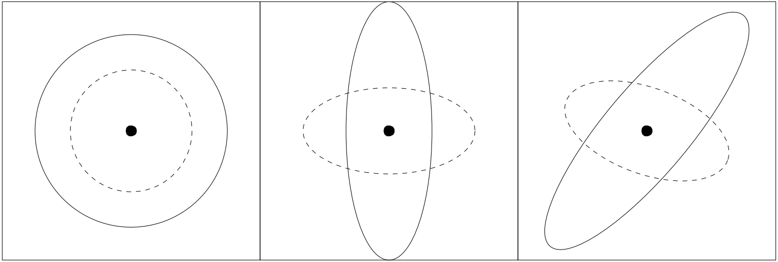
n degrees of freedom

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 $\mathbf{A} \times \mathcal{N}(\mathbf{0}, \mathbf{I}) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}\mathbf{A}^{\mathrm{T}})$ holds for all $\mathbf{A}$.

... any covariance matrix can be uniquely identified with the iso-density ellipsoid $\{\boldsymbol{x} \in \mathbb{R}^n \,|\, (\boldsymbol{x} - \boldsymbol{m})^{\mathrm{T}} \mathbf{C}^{-1} (\boldsymbol{x} - \boldsymbol{m}) = 1\}$

Lines of Equal Density



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

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

$\mathcal{N}(\boldsymbol{m}, \mathbf{C}) \sim \boldsymbol{m} + \mathbf{C}^{\frac{1}{2}} \mathcal{N}(\mathbf{0}, \mathbf{I})$
$(n^2 + n)/2$ 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 $\mathbf{A} \times \mathcal{N}(\mathbf{0}, \mathbf{I}) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}\mathbf{A}^{\mathrm{T}})$ holds for all $\mathbf{A}$.

Where are we?

Adaptation: What do we want to achieve?

New search points are sampled normally distributed

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

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

- 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

  particulary important for ill-conditionned problems

  $\mathbf{C} \propto \boldsymbol{H}^{-1}$ on convex quadratic functions

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 $\boldsymbol{m}$

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

If $\boldsymbol{x}$ better than $\boldsymbol{m}$ select

$$\boldsymbol{m} \leftarrow \boldsymbol{x}$$

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

Non-elitist selection and intermediate (weighted) recombination

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

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

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

Exercice: realize that $\boldsymbol{y}_{i:\lambda}$ 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 $\boldsymbol{x}_i = \boldsymbol{m} + \sigma \underbrace{\boldsymbol{y}_i}_{\sim\mathcal{N}(\mathbf{0},\mathbf{C})}$

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

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

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

The new mean reads

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

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

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

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

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

The new mean reads

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

where

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

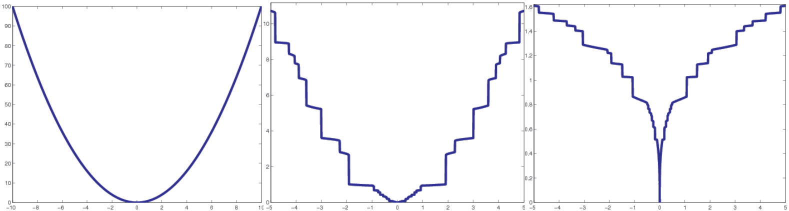
The best μ points are selected from the new solutions (non-elitistic) and weighted intermediate recombination is applied.

Invariance Under Monotonically Increasing Functions

Rank-based algorithms

Update of all parameters uses only the ranks

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

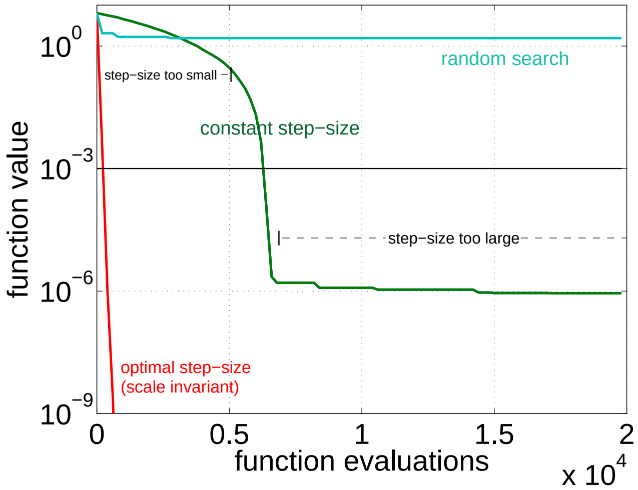


$$g(f(x_{1:\lambda})) \leq g(f(x_{2:\lambda})) \leq \ldots \leq g(f(x_{\lambda:\lambda})) \quad \forall g$$

g is strictly monotonically increasing

g preserves ranks

Why Step-Size Control?



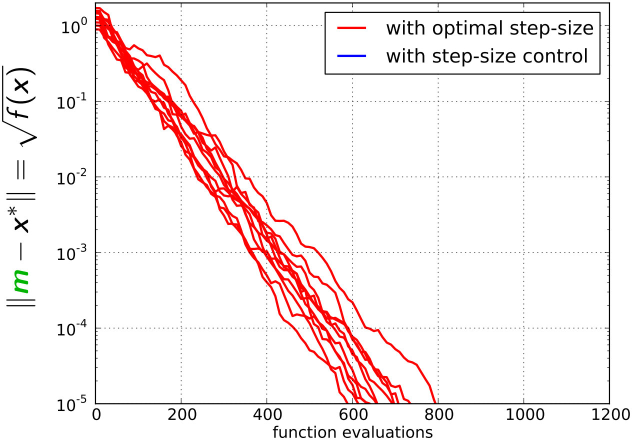
(1+1)-ES
(red & green)

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

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

Why Step-Size Control?

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



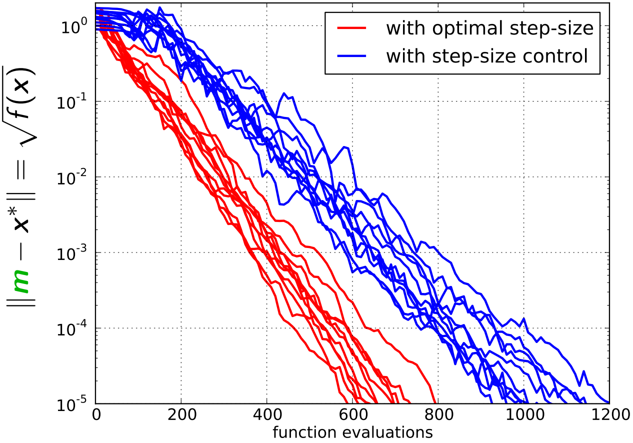
$$f(x) = \sum_{i=1}^{n} x_i^2$$

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

with optimal step-size σ

Why Step-Size Control?

$(5/5_w, 10)$-ES, 2×11 runs



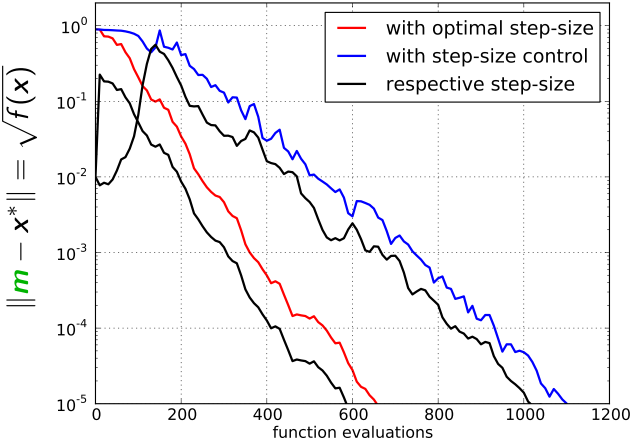
$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$$

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

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

Why Step-Size Control?

$(5/5_\mathrm{w}, 10)$-ES



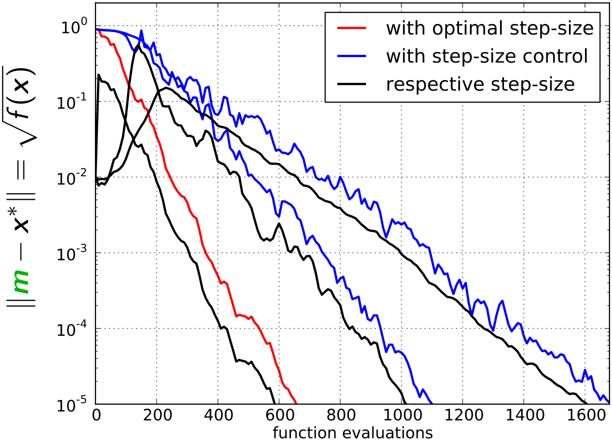
$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$$

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

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

Why Step-Size Control?

$(5/5_w, 10)$-ES

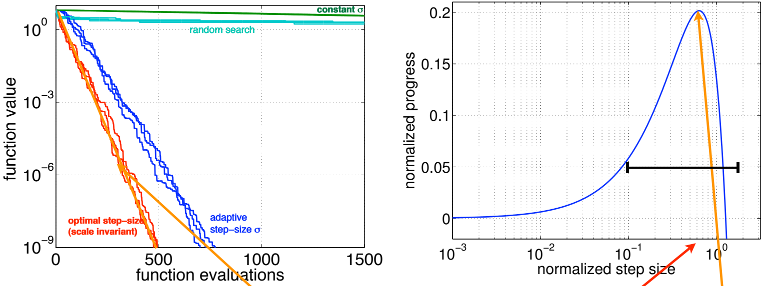


$$f(x) = \sum_{i=1}^{n} x_i^2$$

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

comparing optimal versus default damping parameter d_σ:
$\frac{1700}{1100} \approx 1.5$

Why Step-Size Control?



$\sigma \leftarrow \sigma^*_{\mathrm{opt}}\|\mathrm{parent}\|$ $\qquad -\dfrac{\varphi^*}{n}$ $\qquad \sigma^*_{\mathrm{opt}}$ $\qquad \varphi^*$

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

Step-size control

Theory

- On well conditioned problem (sphere function $f(\boldsymbol{x}) = \|x\|^2$) 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 $f : \mathbb{R}^n \to \mathbb{R}$, for any $y^* \in \mathbb{R}^n$

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

where $\tau = \max_{\sigma \in \mathbb{R}^+} \underbrace{E[\ln^- \| \underbrace{e_1}_{(1,0,\ldots,0)} + \sigma \mathcal{N}\|]}_{=:\varphi(\sigma)}$

"Tight" lower bound

Theorem

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

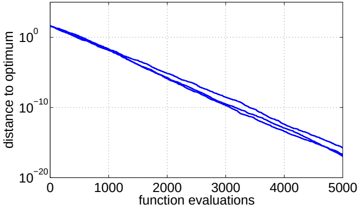
(Log)-Linear convergence of scale-invariant step-size ES

Theorem

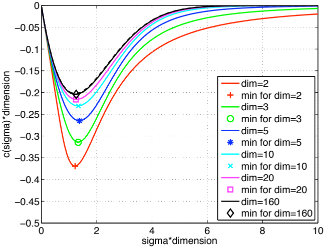
The (1+1)-ES with step-size proportional to the distance to the optimum $\sigma_k = \sigma\|\mathbf{x}\|$ *converges (log)-linearly on the sphere function* $f(\mathbf{x}) = g(\|\mathbf{x}\|)$, *(with* $g : \mathbb{R} \to \mathbb{R}$, *increasing mapping) in the sense*

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

almost surely.



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



Asymptotic results

When $n \to \infty$

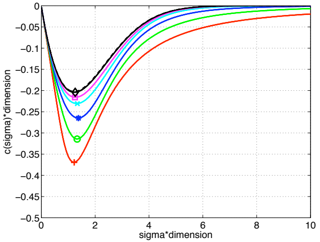
Theorem

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

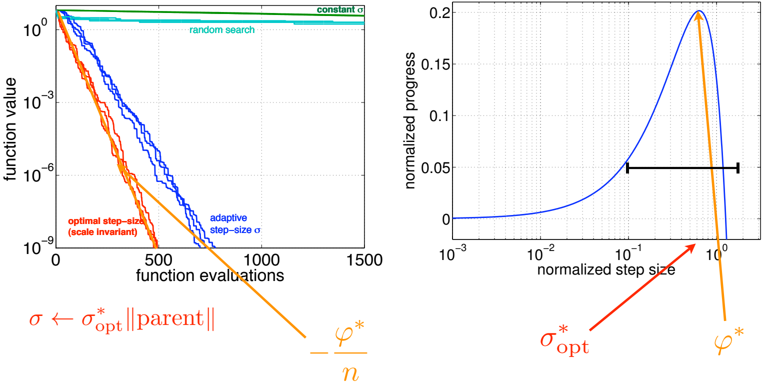
$$\lim_{n\to\infty} n \times \mathrm{CR}_{(1+1)}\left(\frac{\sigma}{n}\right) = \frac{-\sigma}{\sqrt{2\pi}} \exp\left(-\frac{\sigma^2}{8}\right) + \frac{\sigma^2}{2}\Phi\left(-\frac{\sigma}{2}\right)$$

where Φ is the cumulative distribution of a normal distribution.

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



Summary of theory results



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

Methods for Step-Size Control

- 1/5-th success rule[a,b], 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

---

[a] Rechenberg 1973, *Evolutionsstrategie, Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog

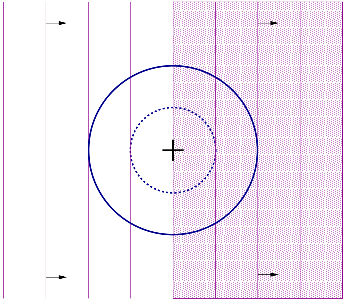
[b] Schumer and Steiglitz 1968. Adaptive step size random search. *IEEE TAC*

[c] Schwefel 1981, *Numerical Optimization of Computer Models*, Wiley

[d] Hansen & Ostermeier 2001, Completely Derandomized Self-Adaptation in Evolution Strategies, *Evol. Comput. 9(2)*

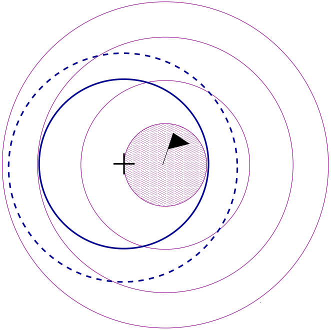
[e] Ostermeier *et al* 1994, Step-size adaptation based on non-local use of selection information, *PPSN IV*

One-fifth success rule



$\downarrow$

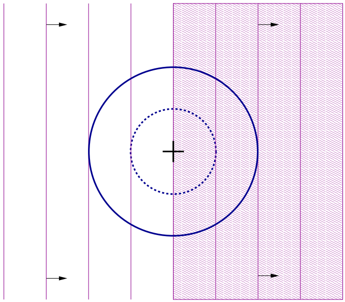
increase σ

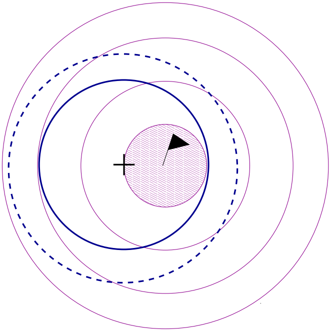


$\downarrow$

decrease σ

One-fifth success rule





Probability of success (p_s)

Probability of success (p_s)

1/2

1/5

"too small"

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

$p_{target} = 1/5$

IF *offspring better parent*

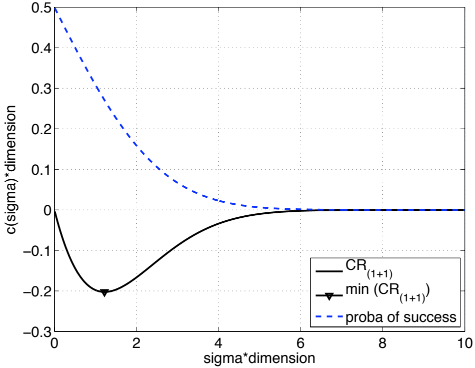
$p_s = 1$, $\sigma \leftarrow \sigma \times \exp(1/3)$

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

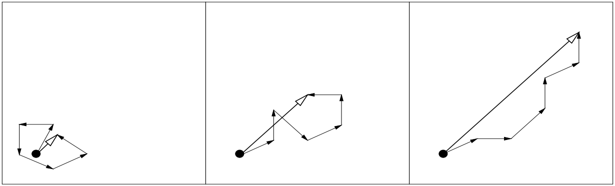
Path Length Control (CSA)

The Concept of Cumulative Step-Size Adaptation

$$\begin{aligned} \mathbf{x}_i &= \mathbf{m} + \sigma\, \mathbf{y}_i \\ \mathbf{m} &\leftarrow \mathbf{m} + \sigma\, \mathbf{y}_w \end{aligned}$$

Measure the length of the *evolution path*

the pathway of the mean vector $\mathbf{m}$ in the iteration sequence



$\Downarrow$ decrease σ

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

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

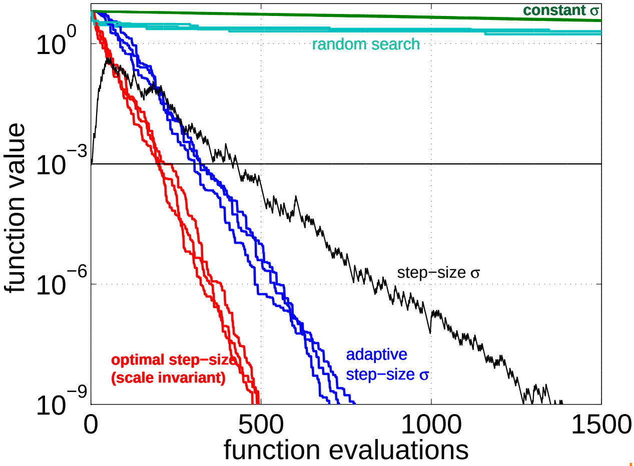
$$\boldsymbol{p}_\sigma \leftarrow (1 - c_\sigma)\, \boldsymbol{p}_\sigma + \underbrace{\sqrt{1 - (1 - c_\sigma)^2}}_{\text{accounts for } 1-c_\sigma} \; \underbrace{\sqrt{\mu_w}}_{\text{accounts for } w_i} \; \boldsymbol{y}_w$$

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

Step-size adaptation

What is achieved

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



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

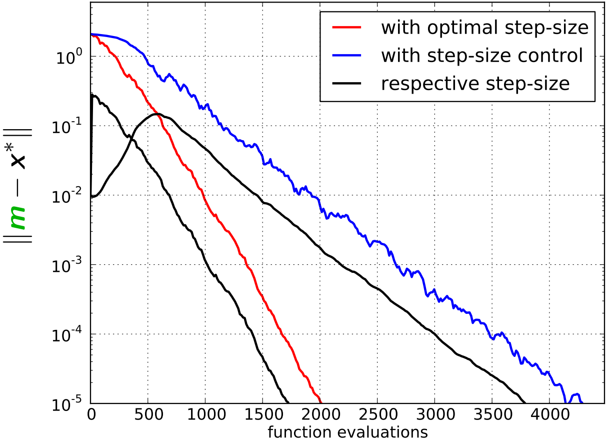
in $[-0.2, 0.8]^n$
for $n = 10$

Linear convergence

Step-size adaptation

What is achieved

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



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

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

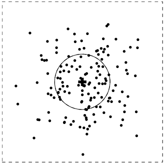
Evolution Strategies

Recalling

New search points are sampled normally distributed

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

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



where

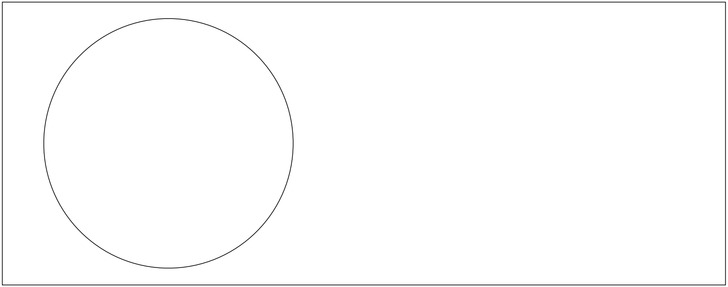
- the mean vector $\boldsymbol{m} \in \mathbb{R}^n$ represents the favorite solution
- the so-called step-size $\sigma \in \mathbb{R}_+$ controls the *step length*
- the covariance matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ determines the shape of the distribution ellipsoid

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

Covariance Matrix Adaptation

Rank-One Update

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

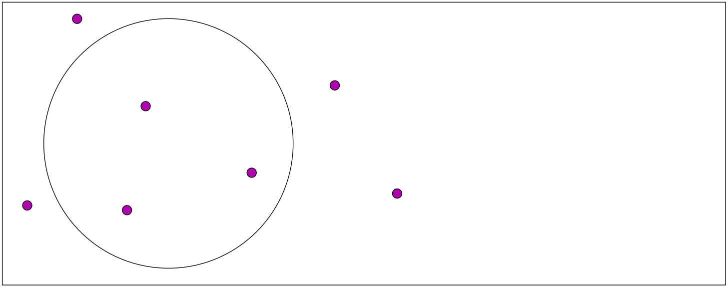


initial distribution, $\mathbf{C} = \mathbf{I}$

Covariance Matrix Adaptation

Rank-One Update

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

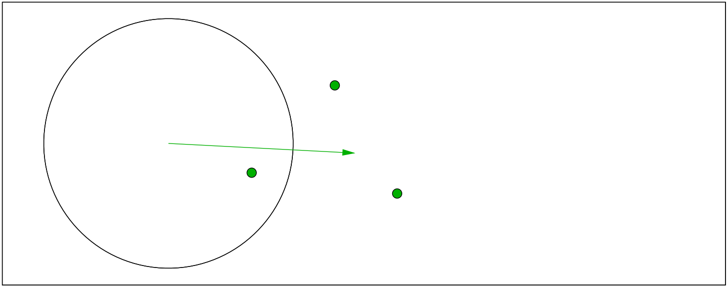


initial distribution, $\mathbf{C} = \mathbf{I}$

Covariance Matrix Adaptation

Rank-One Update

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

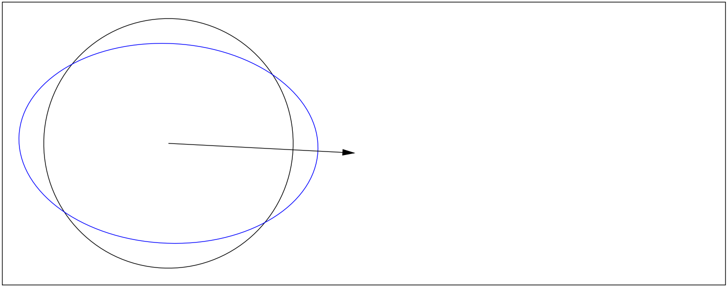


$\boldsymbol{y}_w$, movement of the population mean $\boldsymbol{m}$ (disregarding σ)

Covariance Matrix Adaptation

Rank-One Update

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



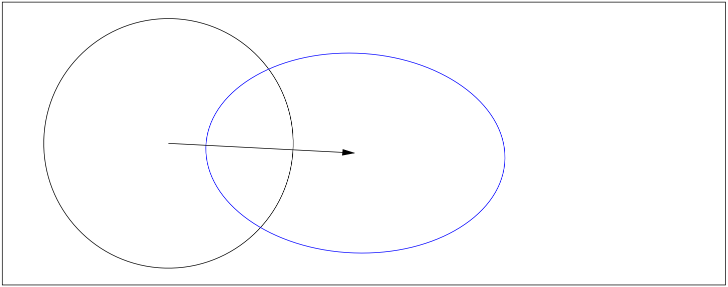
mixture of distribution $\mathbf{C}$ and step $\boldsymbol{y}_w$,

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

Covariance Matrix Adaptation

Rank-One Update

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

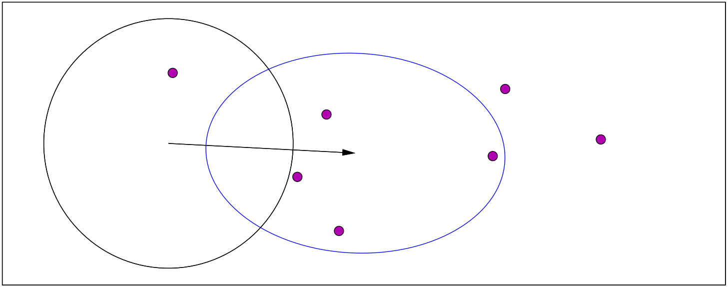


new distribution (disregarding σ)

Covariance Matrix Adaptation

Rank-One Update

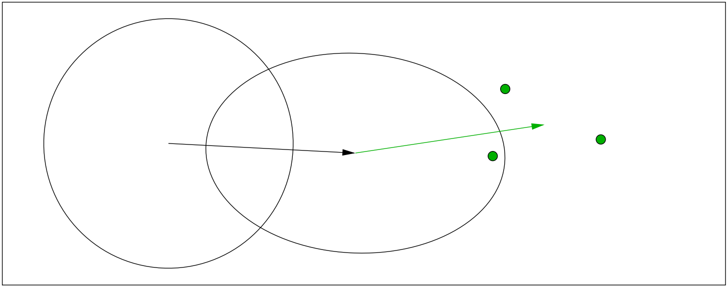
$$\mathbf{m} \leftarrow \mathbf{m} + \sigma \mathbf{y}_w, \quad \mathbf{y}_w = \sum_{i=1}^{\mu} w_i \, \mathbf{y}_{i:\lambda}, \quad \mathbf{y}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C})$$



new distribution (disregarding σ)

Covariance Matrix Adaptation
Rank-One Update

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

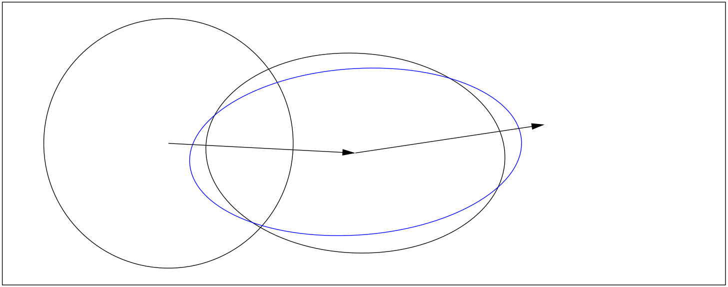


movement of the population mean $\boldsymbol{m}$

Covariance Matrix Adaptation

Rank-One Update

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



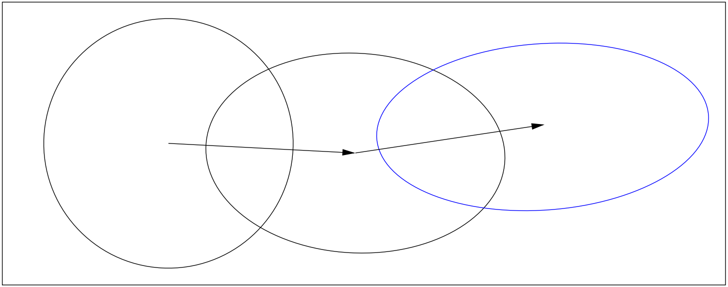
mixture of distribution $\mathbf{C}$ and step $\boldsymbol{y}_w$,

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

Covariance Matrix Adaptation

Rank-One Update

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



new distribution,

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

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

Covariance Matrix Adaptation

Rank-One Update

Initialize $\boldsymbol{m} \in \mathbb{R}^n$, and $\mathbf{C} = \mathbf{I}$, set $\sigma = 1$, learning rate $c_{\text{cov}} \approx 2/n^2$
While not terminate

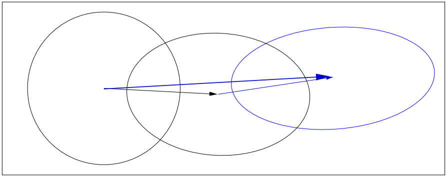
$$\boldsymbol{x}_i = \boldsymbol{m} + \sigma \, \boldsymbol{y}_i, \qquad \boldsymbol{y}_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}),$$

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

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

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 $\mathbf{m}$.

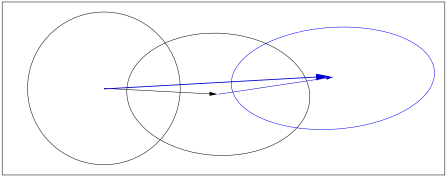


An exponentially weighted sum of steps $\mathbf{y}_w$ is used

$$\mathbf{p}_c \propto \sum_{i=0}^{g} \underbrace{(1 - c_c)^{g-i}}_{\substack{\text{exponentially} \\ \text{fading weights}}} \mathbf{y}_w^{(i)}$$

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 $\boldsymbol{m}$.



An exponentially weighted sum of steps $\boldsymbol{y}_w$ is used

$$\boldsymbol{p}_c \propto \sum_{i=0}^{g} \underbrace{(1-c_c)^{g-i}}_{\substack{\text{exponentially}\\ \text{fading weights}}} \boldsymbol{y}_w^{(i)}$$

The recursive construction of the evolution path (cumulation):

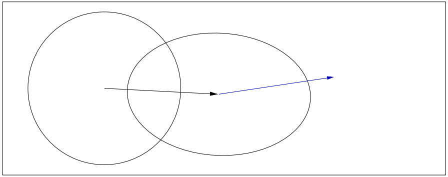
$$\boldsymbol{p}_c \quad \leftarrow \quad \underbrace{(1-c_c)}_{\text{decay factor}} \boldsymbol{p}_c + \underbrace{\sqrt{1-(1-c_c)^2}\sqrt{\mu_w}}_{\text{normalization factor}} \underbrace{\boldsymbol{y}_w}_{\text{input } = \frac{\boldsymbol{m}-\boldsymbol{m}_{\text{old}}}{\sigma}}$$

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

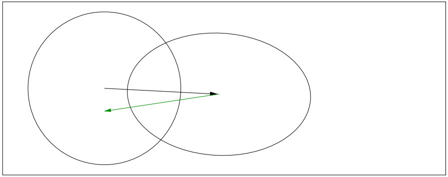
We used $\boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}}$ for updating **C**. Because $\boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}} = -\boldsymbol{y}_w (-\boldsymbol{y}_w)^{\mathrm{T}}$ the sign of $\boldsymbol{y}_w$ is lost.



Cumulation

Utilizing the Evolution Path

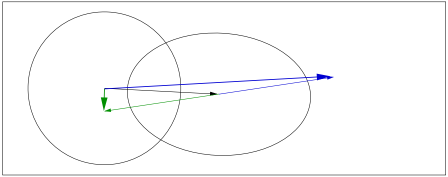
We used $\boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}}$ for updating $\mathbf{C}$. Because $\boldsymbol{y}_w \boldsymbol{y}_w^{\mathrm{T}} = -\boldsymbol{y}_w (-\boldsymbol{y}_w)^{\mathrm{T}}$ the sign of $\boldsymbol{y}_w$ is lost.



Cumulation

Utilizing the Evolution Path

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



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

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

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

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

[3] Hansen, 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

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

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

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

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

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

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

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

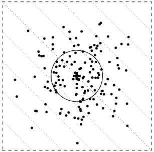
The rank-μ update extends the update rule for large population sizes λ using $\mu > 1$ vectors to update $\mathbf{C}$ at each iteration step.
The matrix

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

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

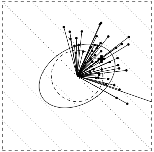
$$\mathbf{C} \leftarrow (1 - c_{\mathrm{cov}})\, \mathbf{C} + c_{\mathrm{cov}}\, \mathbf{C}_\mu$$

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



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

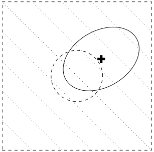
sampling of $\lambda = 150$ solutions where $\mathbf{C} = \mathbf{I}$ and $\sigma = 1$



$\mathbf{C}_\mu = \frac{1}{\mu} \sum \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^{\mathrm{T}}$

$\mathbf{C} \leftarrow (1 - 1) \times \mathbf{C} + 1 \times \mathbf{C}_\mu$

calculating $\mathbf{C}$ where $\mu = 50$, $w_1 = \cdots = w_\mu = \frac{1}{\mu}$, and $c_{\mathrm{cov}} = 1$



$\mathbf{m}_{\mathbf{new}} \leftarrow \mathbf{m} + \frac{1}{\mu} \sum \mathbf{y}_{i:\lambda}$

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 $\lambda \geq 3\,n + 10$

[4] Hansen, 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 $\lambda \geq 3\,n + 10$

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 $\mathcal{O}(n)$.

[4] Hansen, 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 $\lambda \geq 3\,n + 10$

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

[4] Hansen, 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: $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_\mathbf{c} = \mathbf{0}$, $\boldsymbol{p}_\sigma = \mathbf{0}$,
Set: $c_\mathbf{c} \approx 4/n$, $c_\sigma \approx 4/n$, $c_1 \approx 2/n^2$, $c_\mu \approx \mu_w/n^2$, $c_1 + c_\mu \leq 1$,
$d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}$, and $w_{i=1\ldots\lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i{}^2} \approx 0.3\,\lambda$

While not terminate

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

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

$$\boldsymbol{p}_\mathbf{c} \leftarrow (1 - c_\mathbf{c})\, \boldsymbol{p}_\mathbf{c} + \mathbb{1}_{\{\|\boldsymbol{p}_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_\mathbf{c})^2} \sqrt{\mu_w}\, \boldsymbol{y}_w \qquad \text{cumulation for } \mathbf{C}$$

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

$$\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\, \mathbf{C} + c_1\, \boldsymbol{p}_\mathbf{c} \boldsymbol{p}_\mathbf{c}{}^\mathrm{T} + c_\mu \sum_{i=1}^{\mu} w_i\, \boldsymbol{y}_{i:\lambda} \boldsymbol{y}_{i:\lambda}^\mathrm{T} \qquad \text{update } \mathbf{C}$$

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

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

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

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

to the sphere model

$$f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$$

without use of derivatives

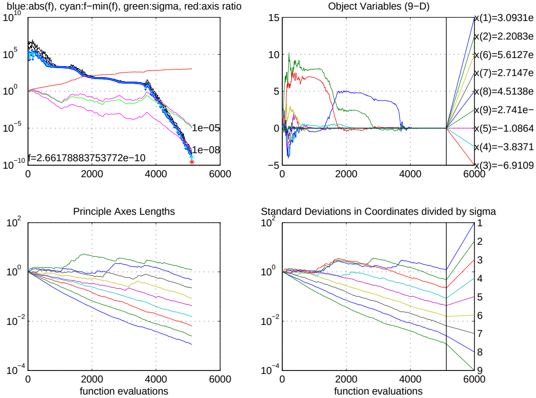
- lines of equal density align with lines of equal fitness

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

in a stochastic sense

Experimentum Crucis (1)

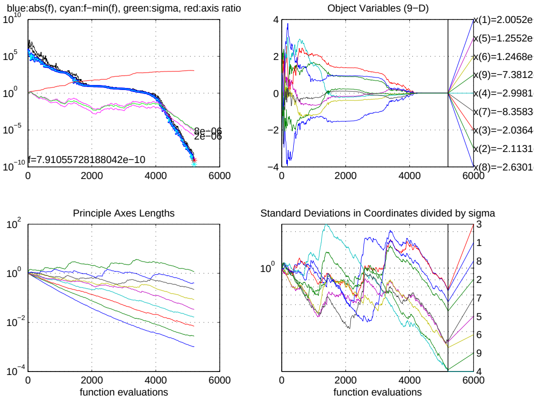
f convex quadratic, separable



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

Experimentum Crucis (2)

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

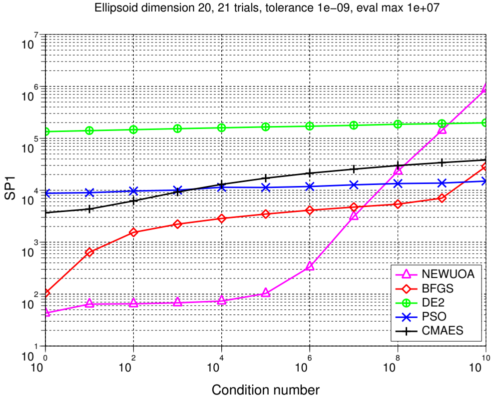


$\mathbf{C} \propto \boldsymbol{H}^{-1}$ for all $g, \mathbf{H}$

$$f(\boldsymbol{x}) = g\left(\boldsymbol{x}^{\mathrm{T}}\mathbf{H}\boldsymbol{x}\right),\ g : \mathbb{R} \to \mathbb{R} \text{ stricly increasing}$$

Comparison to BFGS, NEWUOA, PSO and DE

f convex quadratic, separable with varying condition number α



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

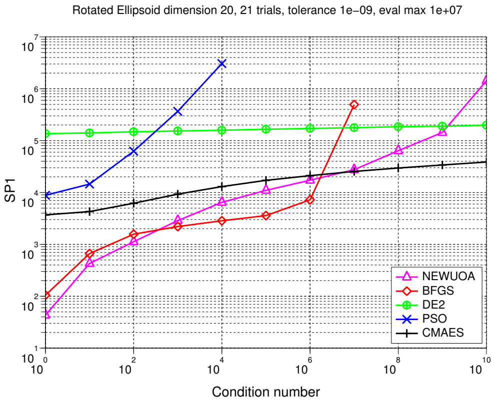
$f(\boldsymbol{x}) = g(\boldsymbol{x}^{\mathrm{T}}\mathbf{H}\boldsymbol{x})$ with
$\boldsymbol{H}$ diagonal
g identity (for BFGS and NEWUOA)
g any order-preserving = strictly increasing function (for all other)

SP1 = average number of objective function evaluations[5] to reach the target function value of $g^{-1}(10^{-9})$

[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



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

$f(\boldsymbol{x}) = g(\boldsymbol{x}^{\mathrm{T}}\mathbf{H}\boldsymbol{x})$ with

$\boldsymbol{H}$ full

g identity (for BFGS and NEWUOA)

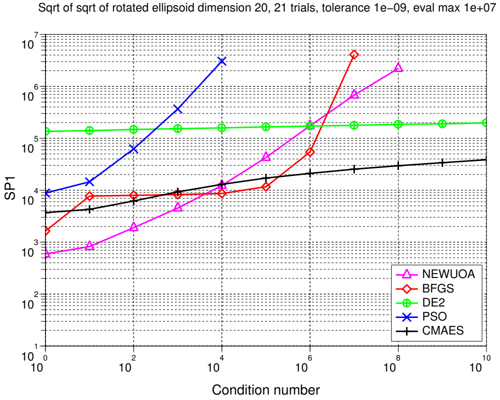
g any order-preserving = strictly increasing function (for all other)

SP1 = average number of objective function evaluations[6] to reach the target function value of $g^{-1}(10^{-9})$

[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)
NEWUAO (Powell 2004)
DE (Storn & Price 1996)
PSO (Kennedy & Eberhart 1995)
CMA-ES (Hansen & Ostermeier 2001)

$f(\boldsymbol{x}) = g(\boldsymbol{x}^{\mathrm{T}}\mathbf{H}\boldsymbol{x})$ with

$\boldsymbol{H}$ full

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

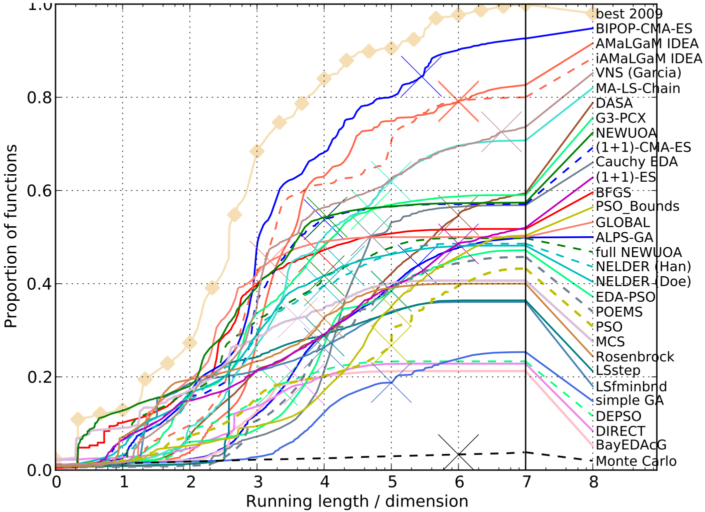
g any order-preserving = strictly increasing function (for all other)

SP1 = average number of objective function evaluations[7] to reach the target function value of $g^{-1}(10^{-9})$

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

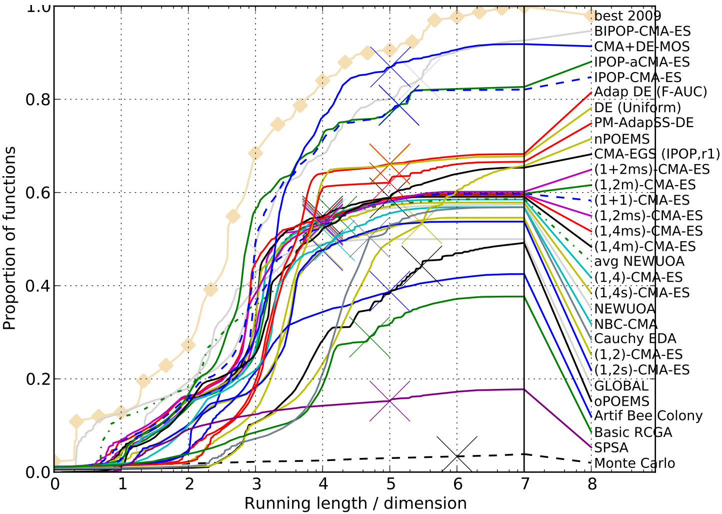
Comparison during BBOB at GECCO 2009

24 functions and 31 algorithms in 20-D



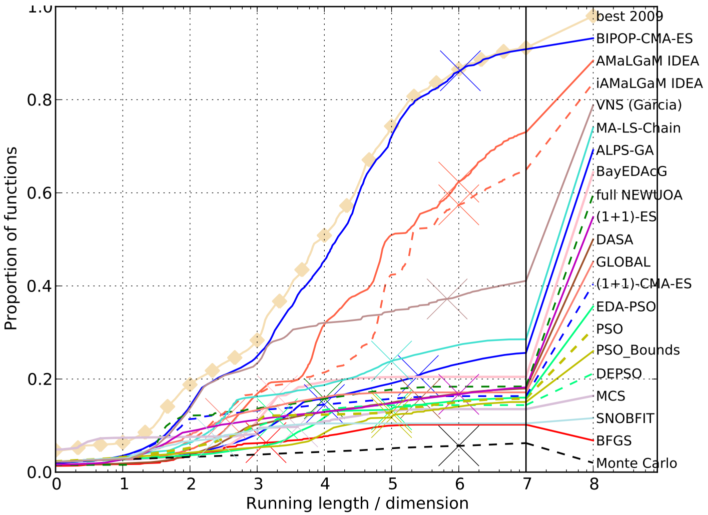
Comparison during BBOB at GECCO 2010

24 functions and 20+ algorithms in 20-D



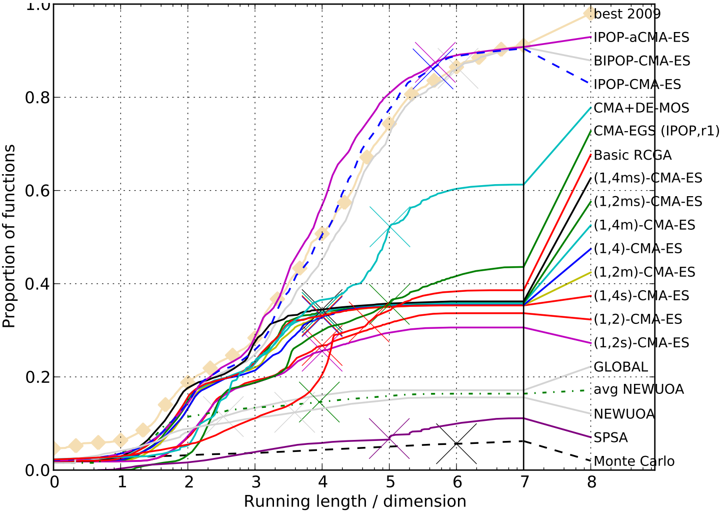
Comparison during BBOB at GECCO 2009

30 noisy functions and 20 algorithms in 20-D



Comparison during BBOB at GECCO 2010

30 noisy functions and 10+ algorithms in 20-D



The Continuous Search Problem

Difficulties of a non-linear optimization problem are

- dimensionality and non-separabitity
  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 $g(f(\boldsymbol{x}))$ 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

   $\mathbf{C} \propto \boldsymbol{H}^{-1} \iff$ adapts a variable metric

   $\iff$ new (rotated) problem representation

   $\implies f(\boldsymbol{x}) = g(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x})$ reduces to $g(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x})$

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