Introduction to Optimization
Constrained Optimization + Descent Methods

December 9, 2016

École Centrale Paris, Châtenay-Malabry, France

Dimo Brockhoff

Inria Saclay – Ile-de-France

Course Overview

Date		Topic
Fri, 7.10.2016		Introduction
Fri, 28.10.2016	D	Introduction to Discrete Optimization + Greedy algorithms I
Fri, 4.11.2016	D	Greedy algorithms II + Branch and bound
Fri, 18.11.2016	D	Dynamic programming
Mon, 21.11.2016 in S103-S105	D	Approximation algorithms ~~and heuristics~~
Fri, 25.11.2016 in S103-S105	C	Randomized Search Heuristics + Intro. to Continuous Opt. I
Mon, 28.11.2016 in S103-S105	C	Introduction to Continuous Optimization II
Mon, 5.12.2016 in S103-S105	C	Introduction to Continuous Optimization III
Fri, 9.12.2016	C	Constrained Optimization + Descent Methods
Mon, 12.12.2016 in S103-S105	C	Derivative Free Optimization I: CMA-ES
Fri, 16.12.2016	C	Derivative Free Optimization II: Benchmarking Optimizers with the COCO platform
Wed, 4.1.2017		Exam

if not indicated otherwise, classes take place in S115-S117

Overview Continuous Optimization Part

Introduction to Continuous Optimization

- examples (from ML / black-box problems)
- typical difficulties in optimization (e.g. constraints)

Mathematical Tools to Characterize Optima

- reminders about differentiability, gradient, Hessian matrix
- unconstraint optimization
  - first and second order conditions
  - convexity
- constrained optimization

Gradient-based Algorithms

- gradient descent
- quasi-Newton method (BFGS)

Derivative Free Optimization

- stochastic adaptive algorithms (CMA-ES)
- Benchmarking Numerical Blackbox Optimizers

Convex Functions

Let U be a convex open set of $\mathbb{R}^n$ and $f: U \to \mathbb{R}$. The function f is said to be convex if for all $\boldsymbol{x}, \boldsymbol{y} \in U$ and for all $t \in [0,1]$

$$f\big((1-t)\boldsymbol{x} + t\boldsymbol{y}\big) \leq (1-t)f(\boldsymbol{x}) + tf(\boldsymbol{y})$$

Theorem

If f is differentiable, then f is convex if and only if for all $\boldsymbol{x}, \boldsymbol{y}$

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \geq \big(\nabla f(\boldsymbol{x})\big)^T (\boldsymbol{y} - \boldsymbol{x})$$

if $n = 1$, the curve is on top of the tangent

If f is twice continuously differentiable, then f is convex if and only if $\nabla^2 f(\boldsymbol{x})$ is positive semi-definite for all $\boldsymbol{x}$.

Convex Functions: Why Convexity?

Examples of Convex Functions:

- $f(\boldsymbol{x}) = a^T\boldsymbol{x} + b$
- $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} + a^T\boldsymbol{x} + b$, A symmetric positive definite
- the negative of the entropy function $(\mathrm{i.e.}\, f(\boldsymbol{x}) = -\sum_{i=1}^{n} \boldsymbol{x}_i \ln(\boldsymbol{x}_i)\,)$

Exercise:

Let $f: U \to \mathbb{R}$ be a convex and differentiable function on a convex open U.
Show that if $\nabla f(\boldsymbol{x}^*) = 0$, then $\boldsymbol{x}^*$ is a global minimum of f

Constrained Optimization

Equality Constraint

Objective:

Generalize the necessary condition of $\nabla f(x) = 0$ at the optima of f

when f is in $\mathcal{C}^1$, i.e. is differentiable and its derivative is continuous

Theorem:

Be U an open set of $(E, ||\ \ ||)$, and $f: U \to \mathbb{R}$, $g: U \to \mathbb{R}$ in $\mathcal{C}^1$.

Let $a \in E$ satisfy

$$\begin{cases} f(a) = \inf \{f(x) \mid x \in \mathbb{R}^n, g(x) = 0\} \\ g(a) = 0 \end{cases}$$

i.e. a is optimum of the problem

If $\nabla g(a) \neq 0$, then there exists a constant $\lambda \in \mathbb{R}$ called *Lagrange multiplier*, such that

$$\underbrace{\nabla f(a) + \lambda \nabla g(a) = 0}$$

i.e. gradients of f and g in a are colinear

Note: a need not be a global minimum but a local one

Geometrical Interpretation Using an Example

Exercise:

Consider the problem

$$\inf\ \{ f(x,y) \mid (x,y) \in \mathbb{R}^2, g(x,y) = 0\}$$

$f(x,y) = y - x^2 \qquad g(x,y) = x^2 + y^2 - 1$

1) Plot the level sets of f, plot $g = 0$
2) Compute ∇f and ∇g
3) Find the solutions with $\nabla f + \lambda \nabla g = 0$

   equation solving with 3 unknowns (x, y, λ)
4) Plot the solutions of 3) on top of the level set graph of 1)

Interpretation of Euler-Lagrange Equation

Intuitive way to retrieve the Euler-Lagrange equation:

- In a local minimum a of a constrained problem, the hypersurfaces (or level sets) $f = f(a)$ and $g = 0$ are necessarily tangent (otherwise we could decrease f by moving along $g = 0$).
- Since the gradients $\nabla f(a)$ and $\nabla g(a)$ are orthogonal to the level sets $f = f(a)$ and $g = 0$, it follows that $\nabla f(a)$ and $\nabla g(a)$ are colinear.

Generalization to More than One Constraint

Theorem

- Assume $f: U \to \mathbb{R}$ and $g_k: U \to \mathbb{R}$ $(1 \leq k \leq p)$ are $\mathcal{C}^1$.
- Let a be such that
  $$\begin{cases} f(a) = \inf \{ f(x) \mid x \in \mathbb{R}^n, \quad g_k(x) = 0, \quad 1 \leq k \leq p \} \\ g_k(a) = 0 \ \text{ for all } 1 \leq k \leq p \end{cases}$$
- If $\left(\nabla g_k(a)\right)_{1 \leq k \leq p}$ are linearly independent, then there exist p real constants $(\lambda_k)_{1 \leq k \leq p}$ such that

$$\nabla f(a) + \sum_{k=1}^{p} \lambda_k \nabla g_k(a) = 0$$

Lagrange multiplier

again: a does not need to be global but local minimum

The Lagrangian

- Define the Lagrangian on $\mathbb{R}^n \times \mathbb{R}^p$ as

$$\mathcal{L}(x, \{\lambda_k\}) = f(x) + \sum_{k=1}^{p} \lambda_k g_k(x)$$

- To find optimal solutions, we can solve the optimality system

$$\begin{cases} \text{Find } (x, \{\lambda_k\}) \in \mathbb{R}^n \times \mathbb{R}^p \text{ such that } \nabla f(x) + \sum_{k=1}^{p} \lambda_k \nabla g_k(x) = 0 \\ g_k(x) = 0 \ \text{ for all } 1 \leq k \leq p \end{cases}$$

$$\Leftrightarrow \begin{cases} \text{Find } (x, \{\lambda_k\}) \in \mathbb{R}^n \times \mathbb{R}^p \text{ such that } \nabla_x \mathcal{L}(x, \{\lambda_k\}) = 0 \\ \nabla_{\lambda_k} \mathcal{L}(x, \{\lambda_k\})(x) = 0 \ \text{ for all } 1 \leq k \leq p \end{cases}$$

Inequality Constraints: Definitions

Let $\mathcal{U} = \{x \in \mathbb{R}^n \mid g_k(x) = 0 \text{ (for } k \in E),\ g_k(x) \leq 0 \text{ (for } k \in I)\}$.

Definition:

The points in $\mathbb{R}^n$ that satisfy the constraints are also called *feasible* points.

Definition:

Let $a \in \mathcal{U}$, we say that the constraint $g_k(x) \leq 0$ (for $k \in I$) is *active* in a if $g_k(a) = 0$.

Inequality Constraint: Karush-Kuhn-Tucker Theorem

Theorem (Karush-Kuhn-Tucker, KKT):

Let U be an open set of $(E, || \, ||)$ and $f: U \to \mathbb{R}$, $g_k: U \to \mathbb{R}$, all $\mathcal{C}^1$

Furthermore, let $a \in U$ satisfy

$$\begin{cases} f(a) = \inf(f(x) \mid x \in \mathbb{R}^n, g_k(x) = 0 \ (\text{for } k \in E), g_k(x) \leq 0 \ (\text{for } k \in \mathrm{I}) \\ g_k(a) = 0 \ (\text{for } k \in E) \\ g_k(a) \leq 0 \ (\text{for } k \in I) \end{cases}$$

also works again for a being a local minimum

Let I_a^0 be the set of constraints that are active in a. Assume that $\left(\nabla g_k(a)\right)_{k \in E \cup I_a^0}$ are linearly independent.

Then there exist $(\lambda_k)_{1 \leq k \leq p}$ that satisfy

$$\begin{cases} \nabla f(a) + \sum_{k=1}^{p} \lambda_k \nabla g_k(a) = 0 \\ g_k(a) = 0 \ (\text{for } k \in E) \\ g_k(a) \leq 0 \ (\text{for } k \in I) \\ \lambda_k \geq 0 \ (\text{for } k \in I_a^0) \\ \lambda_k g_k(a) = 0 \ (\text{for } k \in E \cup I) \end{cases}$$

Inequality Constraint: Karush-Kuhn-Tucker Theorem

Theorem (Karush-Kuhn-Tucker, KKT):

Let U be an open set of $(E, || \, ||)$ and $f: U \to \mathbb{R}$, $g_k: U \to \mathbb{R}$, all $\mathcal{C}^1$

Furthermore, let $a \in U$ satisfy

$$\begin{cases} f(a) = \inf(f(x) \mid x \in \mathbb{R}^n, g_k(x) = 0 \ (\text{for } k \in E), g_k(x) \leq 0 \ (\text{for } k \in \mathrm{I}) \\ g_k(a) = 0 \ (\text{for } k \in E) \\ g_k(a) \leq 0 \ (\text{for } k \in I) \end{cases}$$

Let I_a^0 be the set of constraints that are active in a. Assume that $\left(\nabla g_k(a)\right)_{k \in E \cup I_a^0}$ are linearly independent.

Then there exist $(\lambda_k)_{1 \leq k \leq p}$ that satisfy

$$\begin{cases} \nabla f(a) + \sum_{k=1}^{p} \lambda_k \nabla g_k(a) = 0 \\ g_k(a) = 0 \ (\text{for } k \in E) \\ g_k(a) \leq 0 \ (\text{for } k \in I) \\ \lambda_k \geq 0 \ (\text{for } k \in I_a^0) \\ \lambda_k g_k(a) = 0 \ (\text{for } k \in E \cup I) \end{cases}$$

either active constraint or $\lambda_k = 0$

Descent Methods

Descent Methods

General principle

1. choose an initial point $\boldsymbol{x}_0$, set $t = 1$
2. while not happy
   - choose a descent direction $\boldsymbol{d}_t \neq 0$
   - line search:
     - choose a step size $\sigma_t > 0$
     - set $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \sigma_t \boldsymbol{d}_t$
   - set $t = t + 1$

Remaining questions

- how to choose $\boldsymbol{d}_t$?
- how to choose σ_t?

Gradient Descent

Rationale: $\boldsymbol{d}_t = -\nabla f(\boldsymbol{x}_t)$ is a descent direction

indeed for f differentiable

$$f(x - \sigma \nabla f(x)) = f(x) - \sigma ||\nabla f(x)||^2 + o(\sigma ||\nabla f(x)||)$$
$$< f(x) \text{ for } \sigma \text{ small enough}$$

Step-size

- optimal step-size: $\sigma_t = \underset{\sigma}{\operatorname{argmin}} f(\boldsymbol{x}_t - \sigma \nabla f(\boldsymbol{x}_t))$
- **Line Search:** total or partial optimization w.r.t. σ
  Total is however often too "expensive" (needs to be performed at each iteration step)
  Partial optimization: execute a limited number of trial steps until a loose approximation of the optimum is found. Typical rule for partial optimization: Armijo rule

see next slide and exercise

Stopping criteria:

norm of gradient smaller than ϵ

The Armijo-Goldstein Rule

Choosing the step size:

- Only a decreasing f-value is not enough to converge (quickly)
- Want to have a reasonably large decrease in f

Armijo-Goldstein rule:

- also known as backtracking line search
- starts with a (too) large estimate of σ and reduces it until f is reduced enough
- what is enough?
  - assuming a linear f e.g. $m_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k)$
  - expected decrease if step of σ_k is done in direction $\boldsymbol{d}$: $\sigma_k \nabla f(x_k)^T \boldsymbol{d}$
  - actual decrease: $f(x_k) - f(x_k + \sigma_k \boldsymbol{d})$
  - stop if actual decrease is at least constant times expected decrease (constant typically chosen in [0, 1])

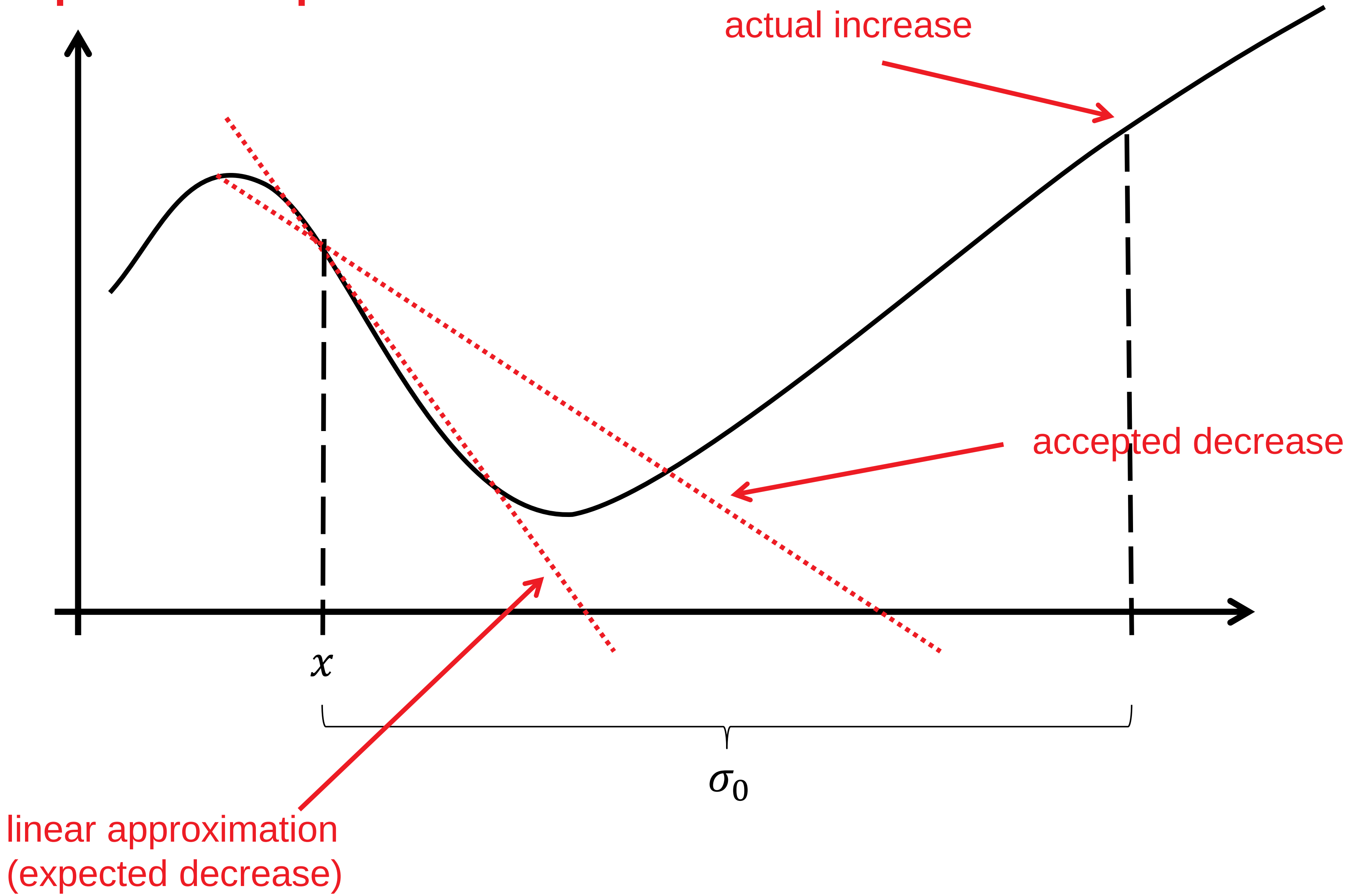
The Actual Algorithm:

Input: descent direction $\mathbf{d}$, point $\mathbf{x}$, objective function $f(\mathbf{x})$ and its gradient $\nabla f(\mathbf{x})$, parameters $\sigma_0 = 10$, $\theta \in [0, 1]$ and $\beta \in (0, 1)$
Output: step-size σ

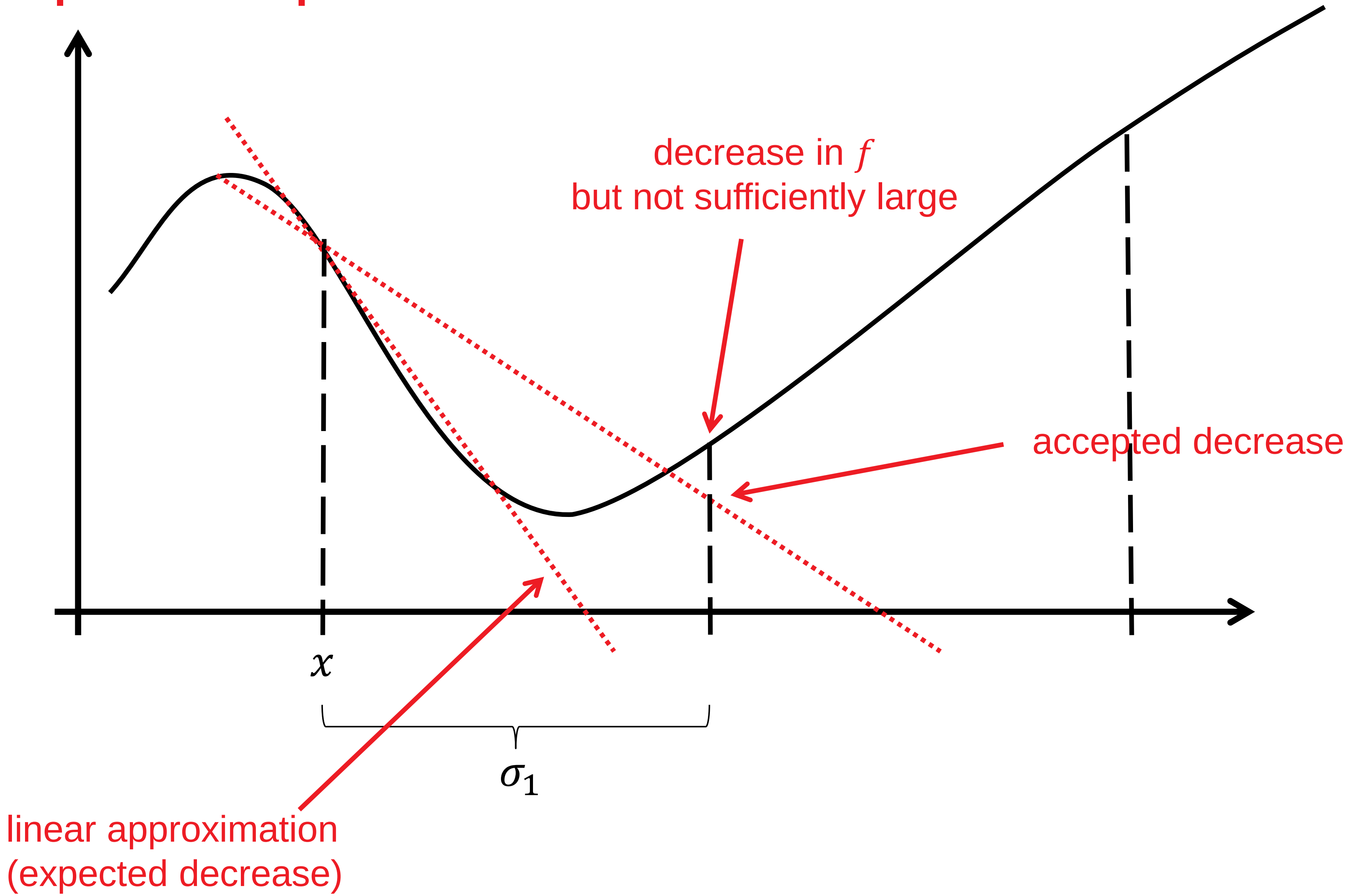
Initialize σ: $\sigma \leftarrow \sigma_0$
while $f(\mathbf{x} + \sigma\mathbf{d}) > f(\mathbf{x}) + \theta\sigma\nabla f(\mathbf{x})^T\mathbf{d}$ **do**
    $\sigma \leftarrow \beta\sigma$
end while

Armijo, in his original publication chose $\beta = \theta = 0.5$.
Choosing $\theta = 0$ means the algorithm accepts any decrease.

Graphical Interpretation

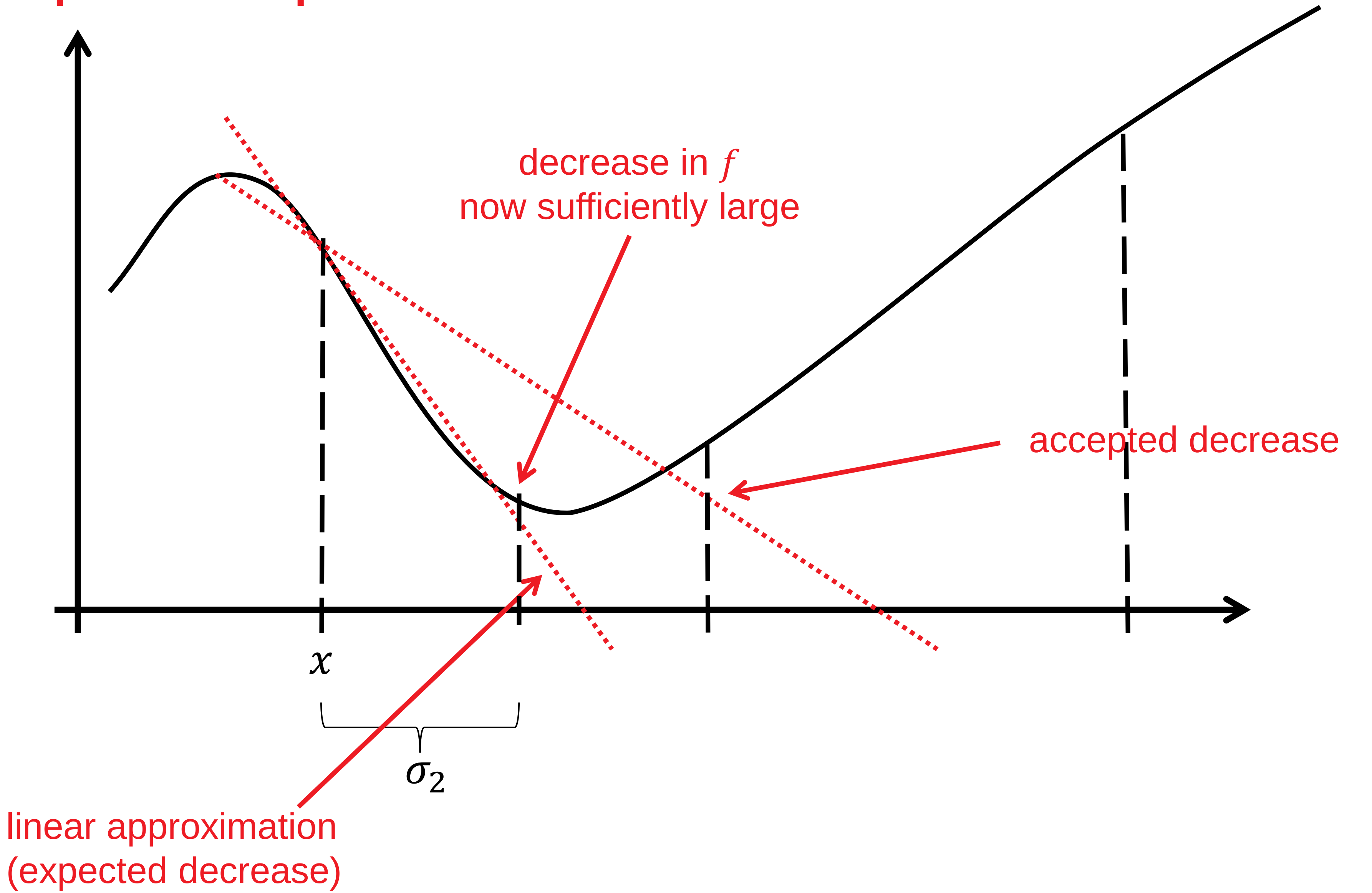


Graphical Interpretation



The Armijo-Goldstein Rule

Graphical Interpretation



Gradient Descent: Simple Theoretical Analysis

Assume f is twice continuously differentiable, convex and that $\mu I_d \preccurlyeq \nabla^2 f(x) \preccurlyeq L I_d$ with $\mu > 0$ holds, assume a fixed step-size $\sigma_t = \frac{1}{L}$

Note: $A \preccurlyeq B$ means $x^T A x \leq x^T B x$ for all x

$$x_{t+1} - x^* = x_t - x^* - \sigma_t \nabla^2 f(y_t)(x_t - x^*) \text{ for some } y_t \in [x_t, x^*]$$

$$x_{t+1} - x^* = \left(I_d - \frac{1}{L} \nabla^2 f(y_t) \right) (x_t - x^*)$$

$$\text{Hence } ||x_{t+1} - x^*||^2 \leq |||I_d - \tfrac{1}{L} \nabla^2 f(y_t)|||^2 \; ||x_t - x^*||^2$$

$$\leq \left(1 - \frac{\mu}{L}\right)^2 ||x_t - x^*||^2$$

Linear convergence: $||x_{t+1} - x^*|| \leq \left(1 - \frac{\mu}{L}\right) ||x_t - x^*||$

algorithm slower and slower with increasing condition number

Non-convex setting: convergence towards stationary point

Newton Method

- descent direction: $-[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)$ [so-called Newton direction]
- The Newton direction:
  - minimizes the best (locally) quadratic approximation of f:
    $\tilde{f}(x+\Delta x) = f(x) + \nabla f(x)^T \Delta x + \frac{1}{2}(\Delta x)^T \nabla^2 f(x) \Delta \mathrm{x}$
  - points towards the optimum on $f(x) = (x-x^*)^T A (x-x^*)$
- however, Hessian matrix is expensive to compute in general and its inversion is also not easy

quadratic convergence

(i.e. $\lim_{k\to\infty} \frac{|x_{k+1}-x^*|}{|x_k-x^*|^2} = \mu > 0$)

Remark: Affine Invariance

Affine Invariance: same behavior on $f(x)$ and $f(Ax+b)$ for $A \in \mathrm{GLn}(\mathbb{R})$

- Newton method is affine invariant

  see **http://users.ece.utexas.edu/~cmcaram/EE381V_2012F/Lecture_6_Scribe_Notes.final.pdf**

- same convergence rate on all convex-quadratic functions
- Gradient method not affine invariant

Quasi-Newton Method: BFGS

$x_{t+1} = x_t - \sigma_t H_t \nabla f(x_t)$ where H_t is an approximation of the inverse Hessian

Key idea of Quasi Newton:

successive iterates x_t, x_{t+1} and gradients $\nabla f(x_t)$, $\nabla f(x_{t+1})$ yield second order information

$$q_t \approx \nabla^2 f(x_{t+1}) p_t$$

where $p_t = x_{t+1} - x_t$ and $q_t = \nabla f(x_{t+1}) - \nabla f(x_t)$

Most popular implementation of this idea: Broyden-Fletcher-Goldfarb-Shanno (BFGS)

- default in MATLAB's **`fminunc`** and python's **`scipy.optimize.minimize`**

Conclusions

I hope it became clear...

...what are the difficulties to cope with when solving numerical optimization problems

in particular dimensionality, non-separability and ill-conditioning

...what are gradient and Hessian

...what is the difference between gradient and Newton direction

...and that adapting the step size in descent algorithms is crucial.

Exercise: Comparing Gradient-Based Algorithms on Convex Quadratic Functions

http://researchers.lille.inria.fr/~brockhof/introoptimization/

Derivative-Free Optimization

DFO = blackbox optimization

Why blackbox scenario?

- gradients are not always available (binary code, no analytical model, ...)
- or not useful (noise, non-smooth, ...)

- problem domain specific knowledge is used only within the black box, e.g. within an appropriate encoding
- some algorithms are furthermore function-value-free, i.e. *invariant* wrt. monotonous transformations of f.

Derivative-Free Optimization Algorithms

- (gradient-based algorithms which approximate the gradient by finite differences)

- coordinate descent
- pattern search methods, e.g. Nelder-Mead
- surrogate-assisted algorithms, e.g. NEWUOA or other trust-region methods
- other function-value-free algorithms
  - typically stochastic
  - evolution strategies (ESs) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES)
  - differential evolution
  - particle swarm optimization
  - simulated annealing
  - ...

Downhill Simplex Method by Nelder and Mead

While not happy do:

[assuming minimization of f and that $x_1, \dots, x_{n+1} \in \mathbb{R}^n$ form a simplex]

1) Order according to the values at the vertices: $f(x_1) \leq f(x_2) \leq \cdots \leq f(x_{n+1})$

2) Calculate x_o, the centroid of all points except x_{n+1}.

3) Reflection

Compute reflected point $x_r = x_o + \alpha (x_o - x_{n+1})$ $(\alpha > 0)$

If x_r better than second worst, but not better than best: $x_{n+1} := x_r$, and go to 1)

4) Expansion

If x_r is the best point so far: compute the expanded point

$$x_e = e_o + \gamma (x_r - x_o)(\text{\textbackslash gamma} > 0)$$

If x_e better than x_r then $x_{n+1} \coloneqq x_e$ and go to 1)

Else $x_{n+1} \coloneqq x_r$ and go to 1)

Else (i.e. reflected point is not better than second worst) continue with 5)

5) Contraction (here: $f(x_r) \geq f(x_n)$)

Compute contracted point $x_c = x_o + \rho(x_{n+1} - x_o)$ $(0 < \rho \leq 0.5)$

If $f(x_c) < f(x_{n+1})$: $x_{n+1} \coloneqq x_c$ and go to 1)

Else go to 6)

6) Shrink

$x_i = x_1 + \sigma(x_i - x_1)$ for all $i \in \{2, \dots, n+1\}$ and go to 1)

*Nelder, John A.; R. Mead (1965). "A simplex method for function minimization". Computer Journal. **7**: 308–313. doi:10.1093/comjnl/7.4.308*

Stochastic Search Template

A stochastic blackbox search template to minimize $\boldsymbol{f}: \mathbb{R}^{\boldsymbol{n}} \to \mathbb{R}$

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

While happy do:

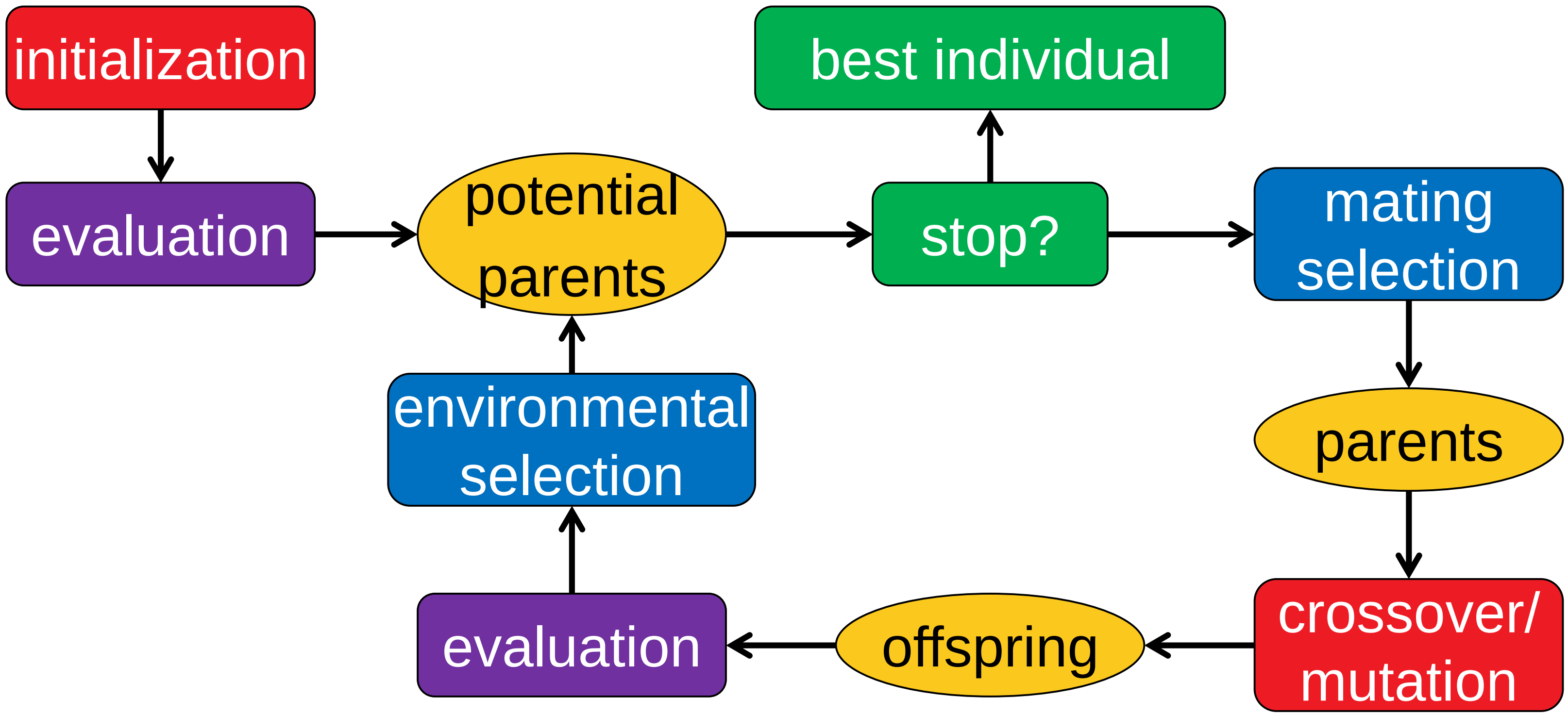
- Sample distribution $P(\boldsymbol{x}|\theta) \to \boldsymbol{x}_1, \dots, \boldsymbol{x}_\lambda \in \mathbb{R}^n$
- Evaluate $\boldsymbol{x}_1, \dots, \boldsymbol{x}_\lambda$ on f
- Update parameters $\theta \leftarrow F_\theta(\theta, \boldsymbol{x}_1, \dots, \boldsymbol{x}_\lambda, f(\boldsymbol{x}_1),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

- All depends on the choice of P and F_θ

deterministic algorithms are covered as well

- In Evolutionary Algorithms, P and F_θ are often defined implicitly via their operators.

Generic Framework of an EA



stochastic operators

"Darwinism"

stopping criteria

Nothing else: just interpretation change

The CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_c = \mathbf{0}$, $\boldsymbol{p}_\sigma = \mathbf{0}$,
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 $w_{i=1\ldots\lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} \approx 0.3\,\lambda$

While not terminate

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

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

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

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

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

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

Not covered on this slide: termination, restarts, useful output, boundaries and encoding

The CMA-ES

Input: $m \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $p_c = \mathbf{0}$, $p_\sigma = \mathbf{0}$,
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 $w_{i=1\ldots\lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i{}^2} \approx 0.3\,\lambda$

While not terminate

$x_i = m + \sigma y_i, \quad y_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), \quad$ for $i = 1, \ldots, \lambda$ — sampling

$m \leftarrow \sum_{i=1}^{\mu} w_i x_{i:\lambda} = m + \sigma y_w$ where $y_w = \sum_{i=1}^{\mu} w_i y_{i:\lambda}$ — update mean

$p_c \leftarrow (1 - c_c)\, p_c + \mathbb{1}_{\{\|p_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w}\, y_w$ — cumulation for $\mathbf{C}$

$p_\sigma \leftarrow (1 - c_\sigma)\, p_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w}\, \mathbf{C}^{-\frac{1}{2}} y_w$ — cumulation for σ

$\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\, \mathbf{C} + c_1\, p_c p_c{}^{\mathrm{T}} + \ldots$

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

Not covered on this slide: terminati
encoding

Goal:
Understand the main principles of this state-of-the-art algorithm.

Copyright Notice

- Last slide was taken from **`https://www.lri.fr/~hansen/copenhagen-cma-es.pdf`** (copyright by Nikolaus Hansen, one of the main inventors of the CMA-ES algorithms)
- In the following, I will borrow more slides from there and from **`http://researchers.lille.inria.fr/~brockhof/optimizationSaclay/slides/20151106-continuousoptIV.pdf`** (by Anne Auger)

- In the following and the online material in particular, I refer to these pdfs as [Hansen, p. X] and [Auger, p. Y] respectively.

The CMA-ES

Input: $m \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $p_c = \mathbf{0}$, $p_\sigma = \mathbf{0}$,
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 $w_{i=1\ldots\lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i{}^2} \approx 0.3\,\lambda$

While not terminate

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

$$m \leftarrow \sum_{i=1}^{\mu} w_i x_{i:\lambda} = m + \sigma y_w \quad \text{where } y_w = \sum_{i=1}^{\mu} w_i y_{i:\lambda} \qquad \text{update mean}$$

$$p_c \leftarrow (1 - c_c)\, p_c + 1\!\!1_{\{\|p_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w}\, y_w \qquad \text{cumulation for } \mathbf{C}$$

$$p_\sigma \leftarrow (1 - c_\sigma)\, p_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w}\, \mathbf{C}^{-\frac{1}{2}} y_w \qquad \text{cumulation for } \sigma$$

$$\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\, \mathbf{C} + c_1\, p_c {p_c}^{\mathrm{T}} + \ldots$$

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

Not covered on this slide: terminati…
encoding

Goal:
Understand the main principles of this state-of-the-art algorithm.

CMA-ES: Stochastic Search Template

A stochastic blackbox search template to minimize $\boldsymbol{f}: \mathbb{R}^{\boldsymbol{n}} \to \mathbb{R}$

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

While happy do:

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

For CMA-ES and evolution strategies in general:

sample distributions = multivariate Gaussian distributions

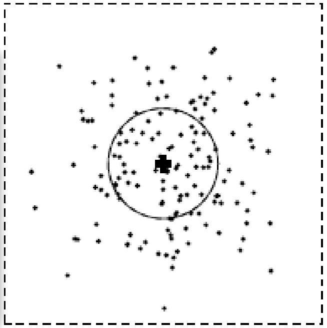
Sampling New Candidate Solutions (Offspring)

Evolution Strategies

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

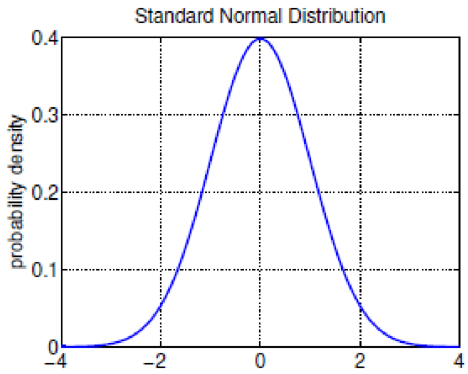
here, all new points are sampled with the same parameters

it remains to show how to adapt the parameters, but for now: normal distributions

from [Auger, p. 10]

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

from [Auger, p. 11]

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

from [Auger, p. 12]

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 $\mathbf{C}$.

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

from [Auger, p. 13]

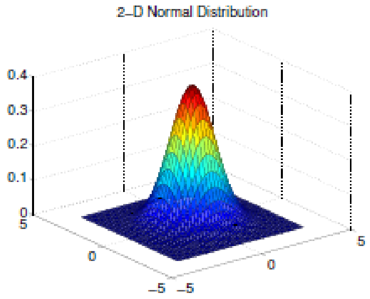
The Multi-Variate (n-Dimensional) Normal Distribution

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



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

from [Auger, p. 13]

The Multi-Variate (n-Dimensional) Normal Distribution

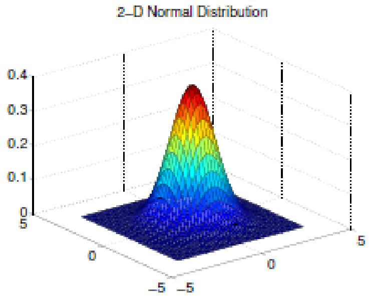
Any multi-variate normal distribution $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$ is uniquely determined by its mean value $\boldsymbol{m} \in \mathbb{R}^n$ and its symmetric positive definite $n \times n$ covariance matrix $\mathbf{C}$.

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

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 $\mathbf{C}$

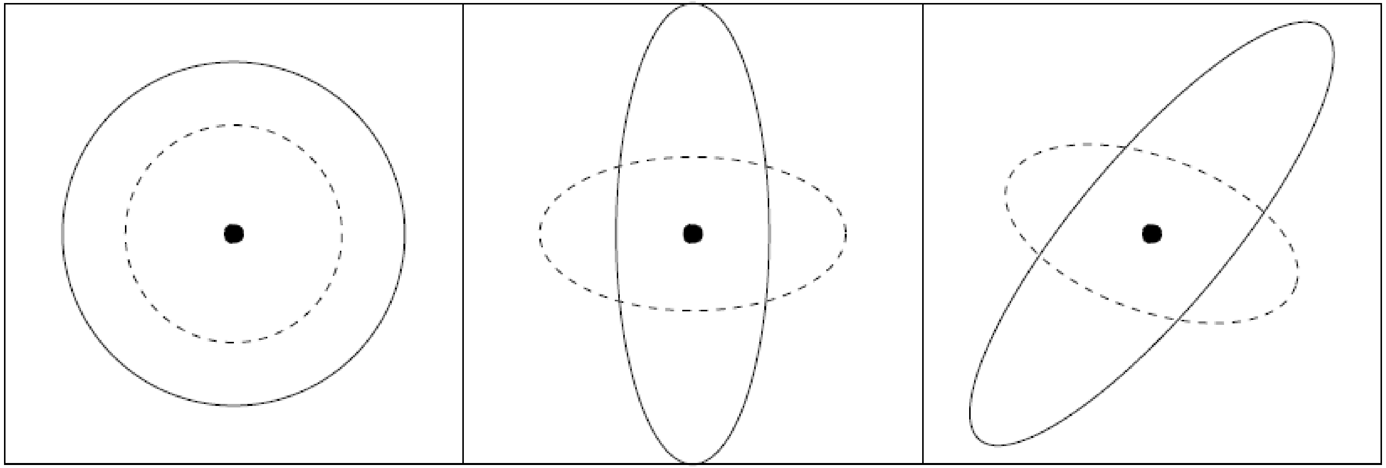
- determines the shape
- geometrical interpretation: any covariance matrix can be uniquely identified with the iso-density ellipsoid $\{\boldsymbol{x} \in \mathbb{R}^n \,|\, (\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}\mathbf{C}^{-1}(\boldsymbol{x}-\boldsymbol{m}) = 1\}$

from [Auger, p. 13]

Covariance Matrix: Lines of Equal Density

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

Lines of Equal Density



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

one degree of freedom σ

components are independent standard normally distributed

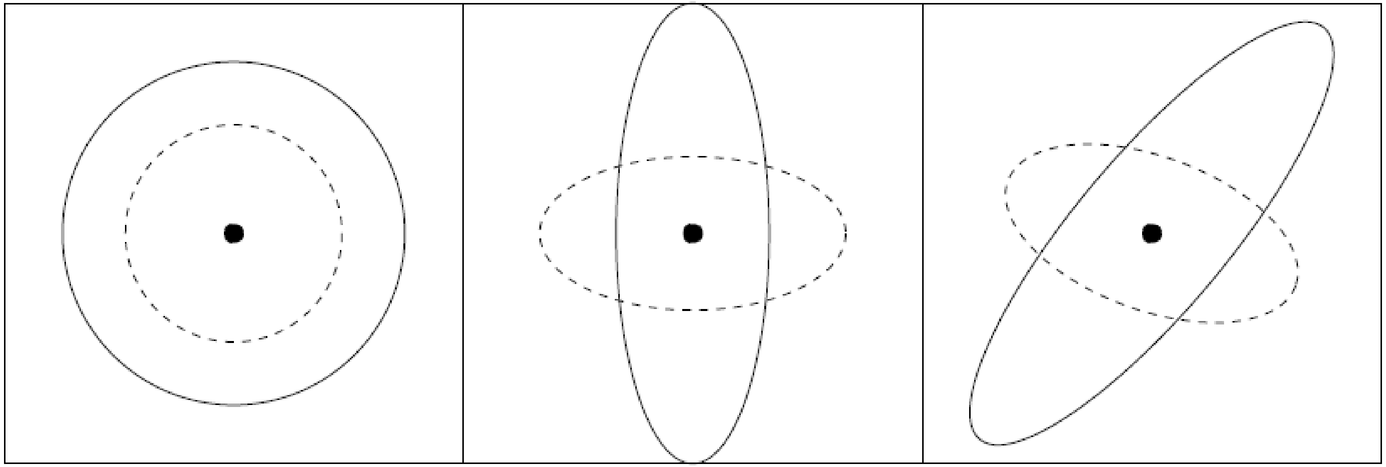
where $\mathbf{I}$ is the identity matrix (isotropic case) and $\mathbf{D}$ is a diagonal matrix (reasonable for separable problems) and $\mathbf{A} \times \mathcal{N}(\mathbf{0}, \mathbf{I}) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}\mathbf{A}^{\mathrm{T}})$ holds for all $\mathbf{A}$.

from [Auger, p. 14]

Covariance Matrix: Lines of Equal Density

. . . 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

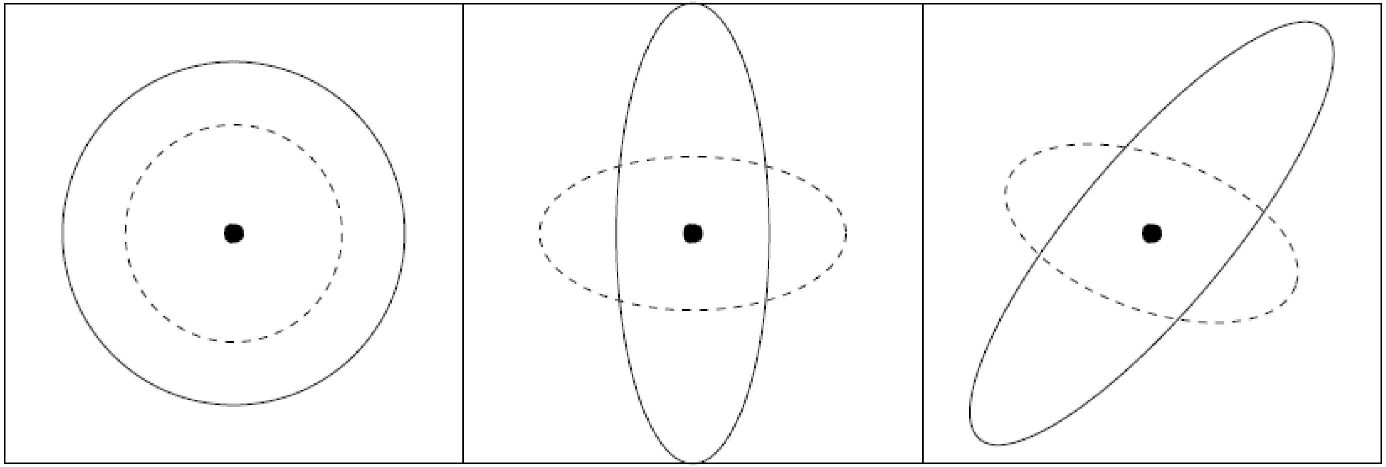
where $\mathbf{I}$ is the identity matrix (isotropic case) and $\mathbf{D}$ is a diagonal matrix (reasonable for separable problems) and $\mathbf{A} \times \mathcal{N}(\mathbf{0}, \mathbf{I}) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}\mathbf{A}^{\mathrm{T}})$ holds for all $\mathbf{A}$.

from [Auger, p. 14]

Covariance Matrix: Lines of Equal Density

...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



$\mathcal{N}(\boldsymbol{m}, \sigma^2 \mathbf{I}) \sim \boldsymbol{m} + \sigma \mathcal{N}(\mathbf{0}, \mathbf{I})$	$\mathcal{N}(\boldsymbol{m}, \mathbf{D}^2) \sim \boldsymbol{m} + \mathbf{D} \mathcal{N}(\mathbf{0}, \mathbf{I})$	$\mathcal{N}(\boldsymbol{m}, \mathbf{C}) \sim \boldsymbol{m} + \mathbf{C}^{\frac{1}{2}} \mathcal{N}(\mathbf{0}, \mathbf{I})$
one degree of freedom σ	n degrees of freedom	$(n^2 + n)/2$ degrees of freedom
components are independent standard normally distributed	components are independent, scaled	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 $\mathbf{A}$.

from [Auger, p. 14]

Adaptation: What do we want to achieve?

New search points are sampled normally distributed

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

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

from [Auger, p. 16]

Adaptation of the Mean

Evolution Strategies

Terminology

μ: # of parents, λ: # of offspring

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

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

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

$(1 + 1)$-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

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

why?

from [Hansen, p. 35]

Non-Elitism and Weighted Recombination

The $(\mu/\mu, \lambda)$-ES

Non-elitist selection and intermediate (weighted) recombination

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

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

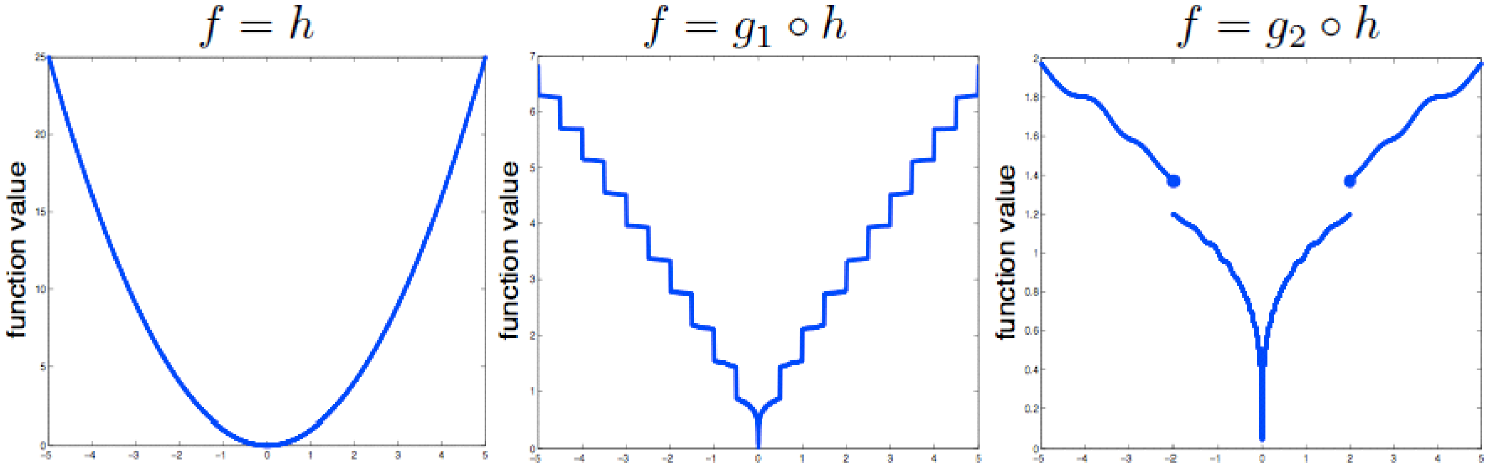
$$w_1 \geq \cdots \geq w_\mu > 0, \quad \sum_{i=1}^{\mu} w_i = 1, \quad \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} =: \mu_w \approx \frac{\lambda}{4}$$

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

from [Hansen, p. 34]

Invariance Against Order-Preserving f-Transformations

Invariance: Function-Value Free Property



Three functions belonging to the same equivalence class

A *function-value free search algorithm* is invariant under the transformation with any order preserving (strictly increasing) g.

Invariances make

- observations meaningful
- algorithms predictable and/or "robust"

as a rigorous notion of generalization

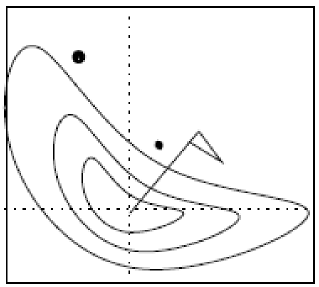
from [Hansen, p. 37]

Invariance Against Translations in Search Space

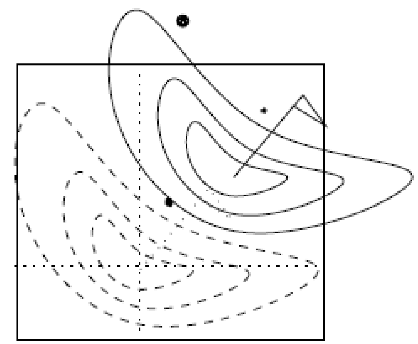
Basic Invariance in Search Space

- translation invariance

is true for most optimization algorithms



$$f(\boldsymbol{x}) \leftrightarrow f(\boldsymbol{x} - \boldsymbol{a})$$



Identical behavior on f and f_a

$$\begin{aligned} f: \quad & \boldsymbol{x} \mapsto f(\boldsymbol{x}), & \boldsymbol{x}^{(t=0)} &= \boldsymbol{x}_0 \\ f_a: \quad & \boldsymbol{x} \mapsto f(\boldsymbol{x} - \boldsymbol{a}), & \boldsymbol{x}^{(t=0)} &= \boldsymbol{x}_0 + \boldsymbol{a} \end{aligned}$$

No difference can be observed w.r.t. the argument of f

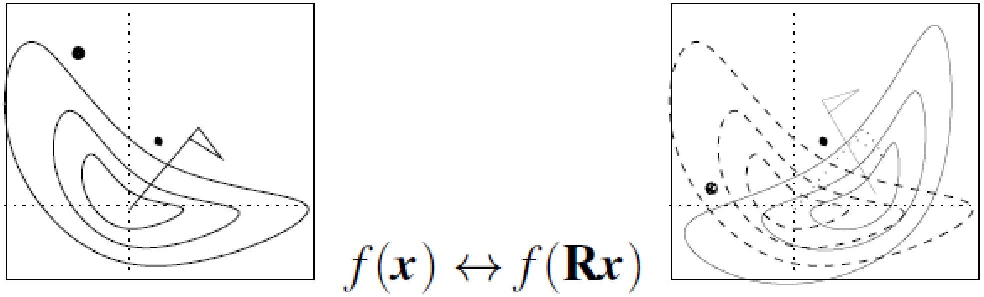
from [Hansen, p. 38]

Rotational Invariance in Search Space

- invariance to orthogonal (rigid) transformations $\mathbf{R}$, where $\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{I}$

e.g. true for simple evolution strategies

recombination operators might jeopardize rotational invariance



$f(\boldsymbol{x}) \leftrightarrow f(\mathbf{R}\boldsymbol{x})$

Identical behavior on f and $f_{\mathbf{R}}$

$$\begin{aligned} f: \quad & \boldsymbol{x} \mapsto f(\boldsymbol{x}), & \boldsymbol{x}^{(t=0)} &= \boldsymbol{x}_0 \\ f_{\mathbf{R}}: \quad & \boldsymbol{x} \mapsto f(\mathbf{R}\boldsymbol{x}), & \boldsymbol{x}^{(t=0)} &= \mathbf{R}^{-1}(\boldsymbol{x}_0) \end{aligned}$$

45

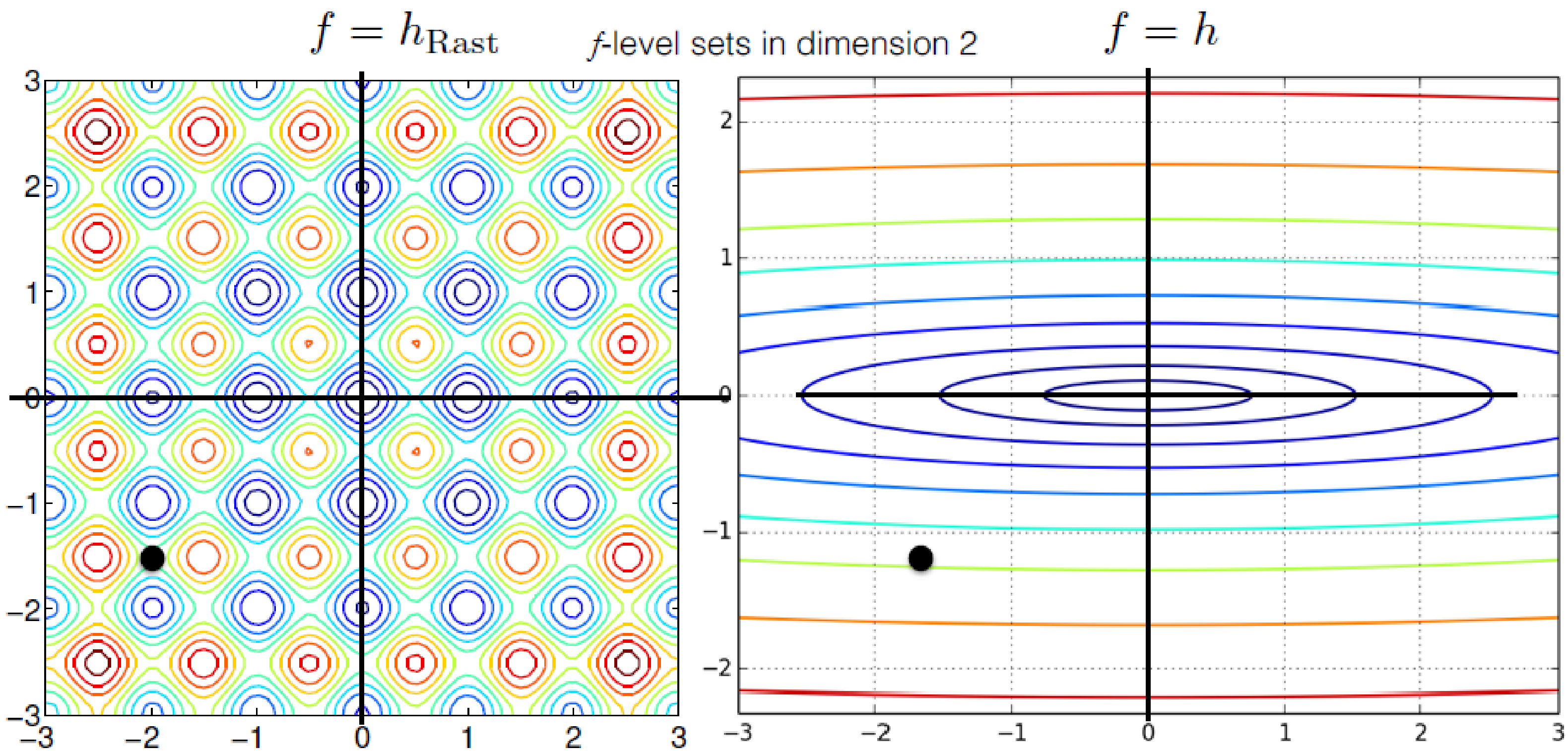
No difference can be observed w.r.t. the argument of f

[4] 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

[5] Hansen 2000. Invariance, Self-Adaptation and Correlated Mutations in Evolution Strategies. *Parallel Problem Solving from Nature PPSN VI*

from [Hansen, p. 39]

Invariance Under Rigid Search Space Transformations

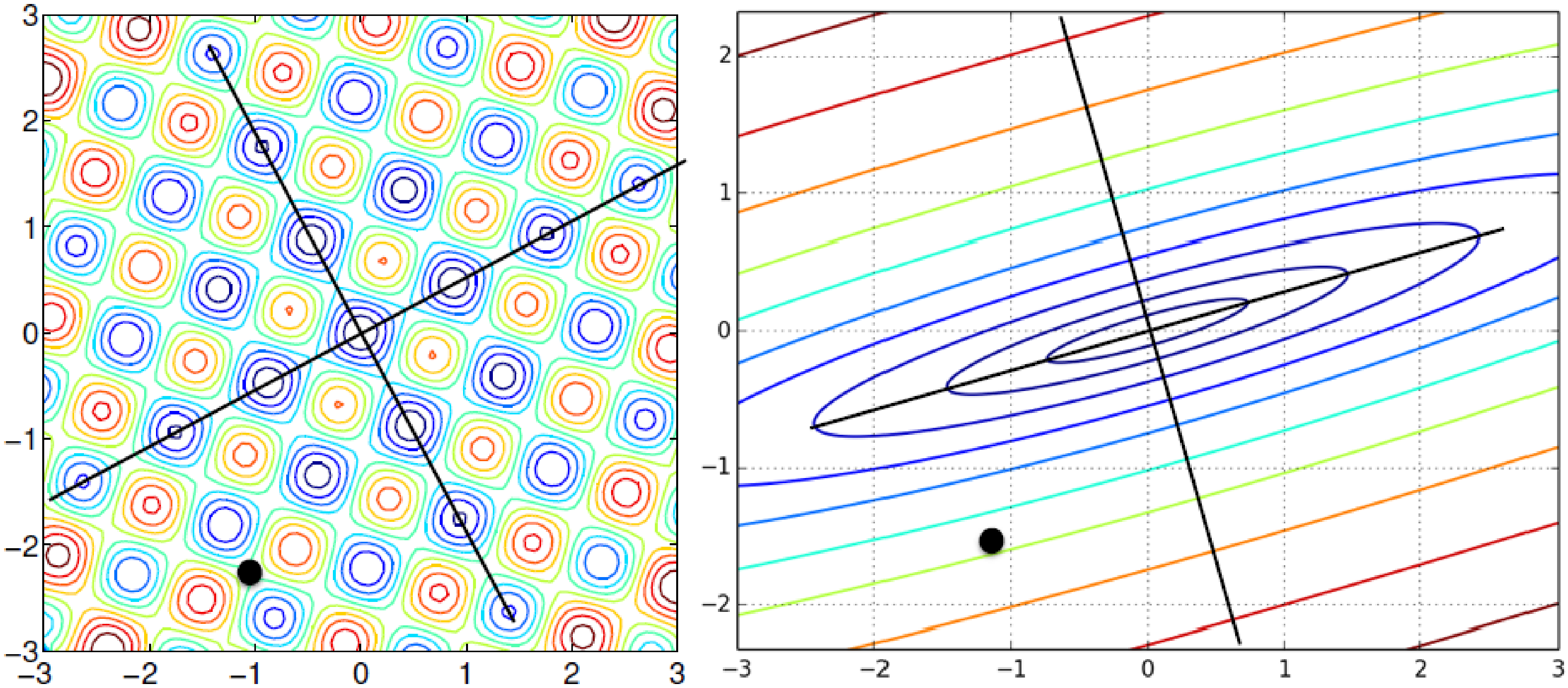


for example, invariance under search space rotation
(separable $\Leftrightarrow$ non-separable)

from [Hansen, p. 40]

Invariance Under Rigid Search Space Transformations

$f = h_{\mathrm{Rast}} \circ R$ f-level sets in dimension 2 $f = h \circ R$

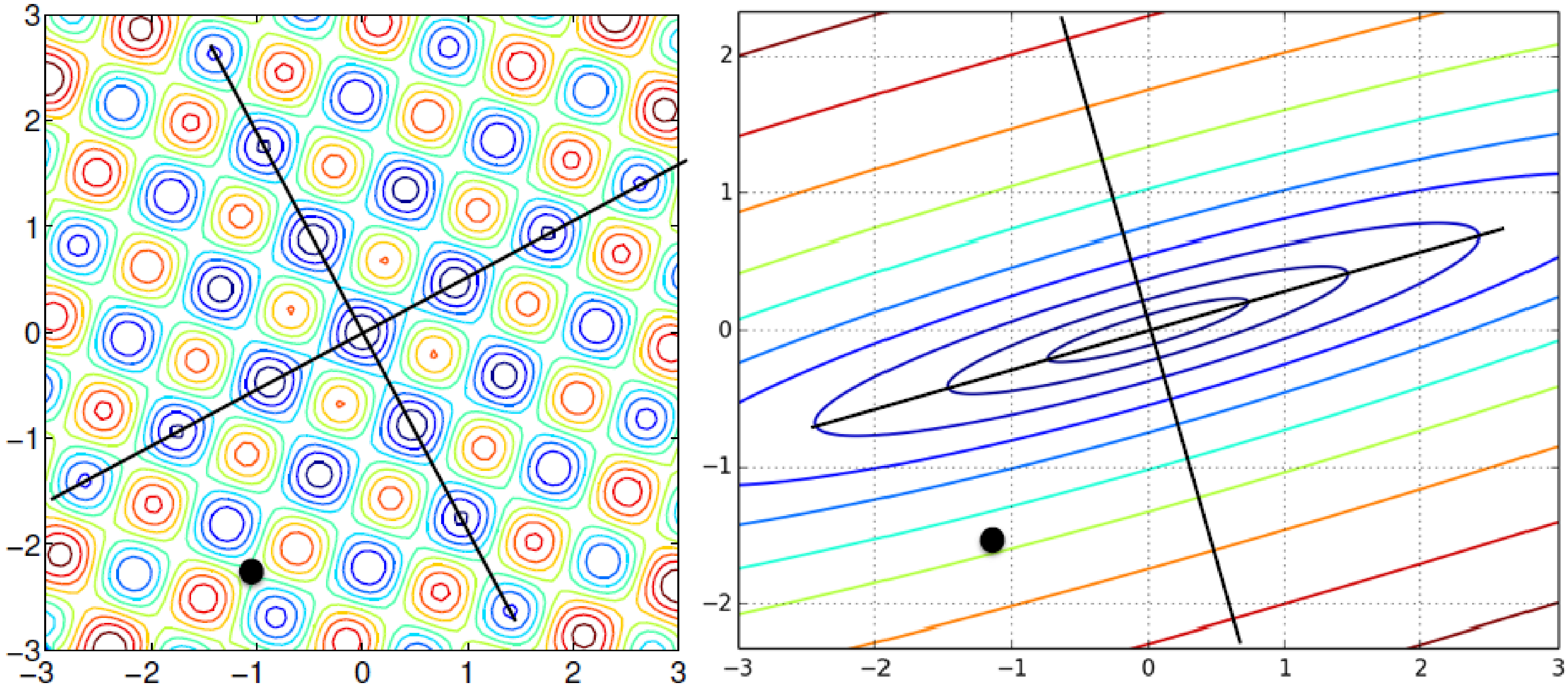


for example, invariance under search space rotation
(separable ⇔ non-separable)

from [Hansen, p. 41]

Invariance Under Rigid Search Space Transformations

$f = h_{\mathrm{Rast}} \circ R$ f-level sets in dimension 2 $f = h \circ R$



for example, invariance un
(separable $\Leftrightarrow$ non-separab

mainly Nelder-Mead and CMA-ES
have this property

Invariance

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

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

  are only useful if they do generalize to other problems

- Invariance is a strong non-empirical statement about generalization

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

consequently, invariance is important for the evaluation of search algorithms

from [Hansen, p. 43]

Step-Size Adaptation

Recap CMA-ES: What We Have So Far

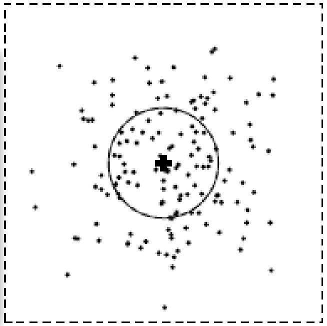
Evolution Strategies

Recalling

New search points are sampled normally distributed

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

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



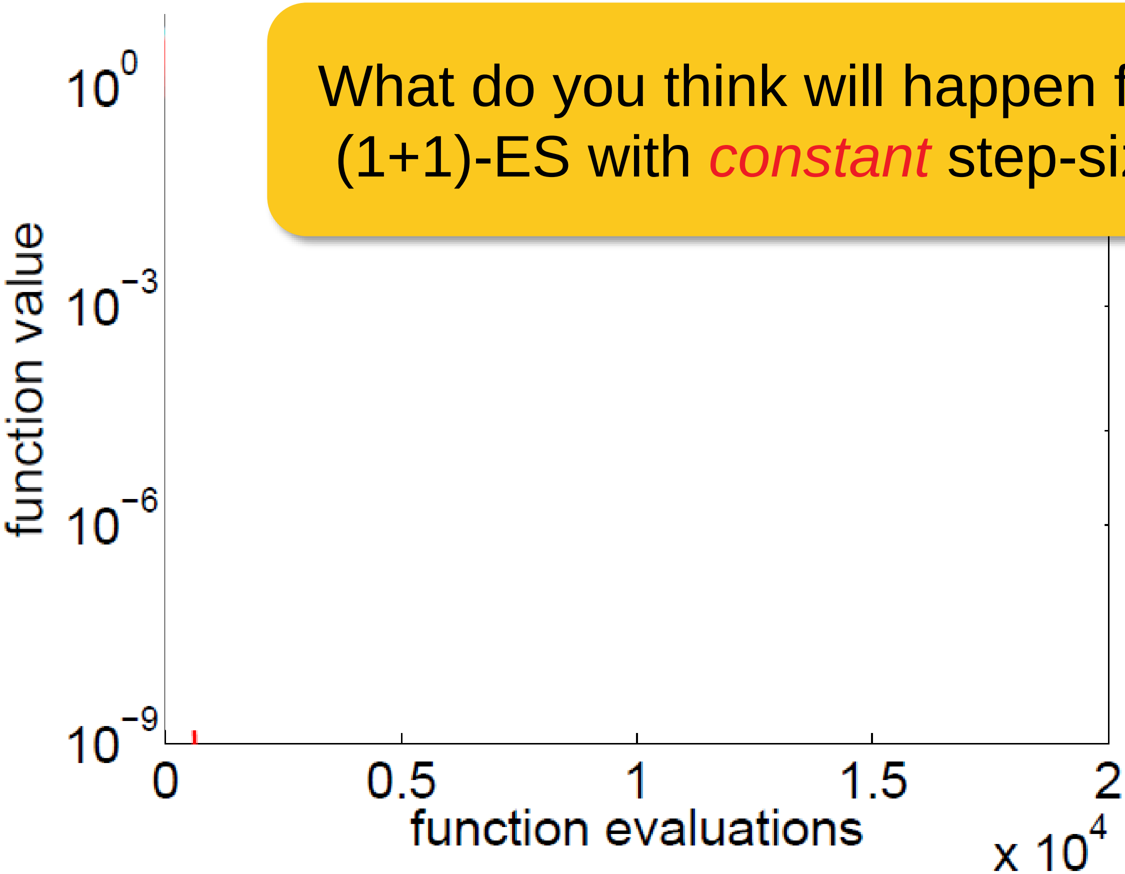
where

- the mean vector $m \in \mathbb{R}^n$ represents the favorite solution and $m \leftarrow \sum_{i=1}^{\mu} w_i x_{i:\lambda}$
- 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 σ and $\mathbf{C}$.

from [Hansen, p. 45]

Why Step-Size Control?



What do you think will happen for a (1+1)-ES with *constant* step-size?

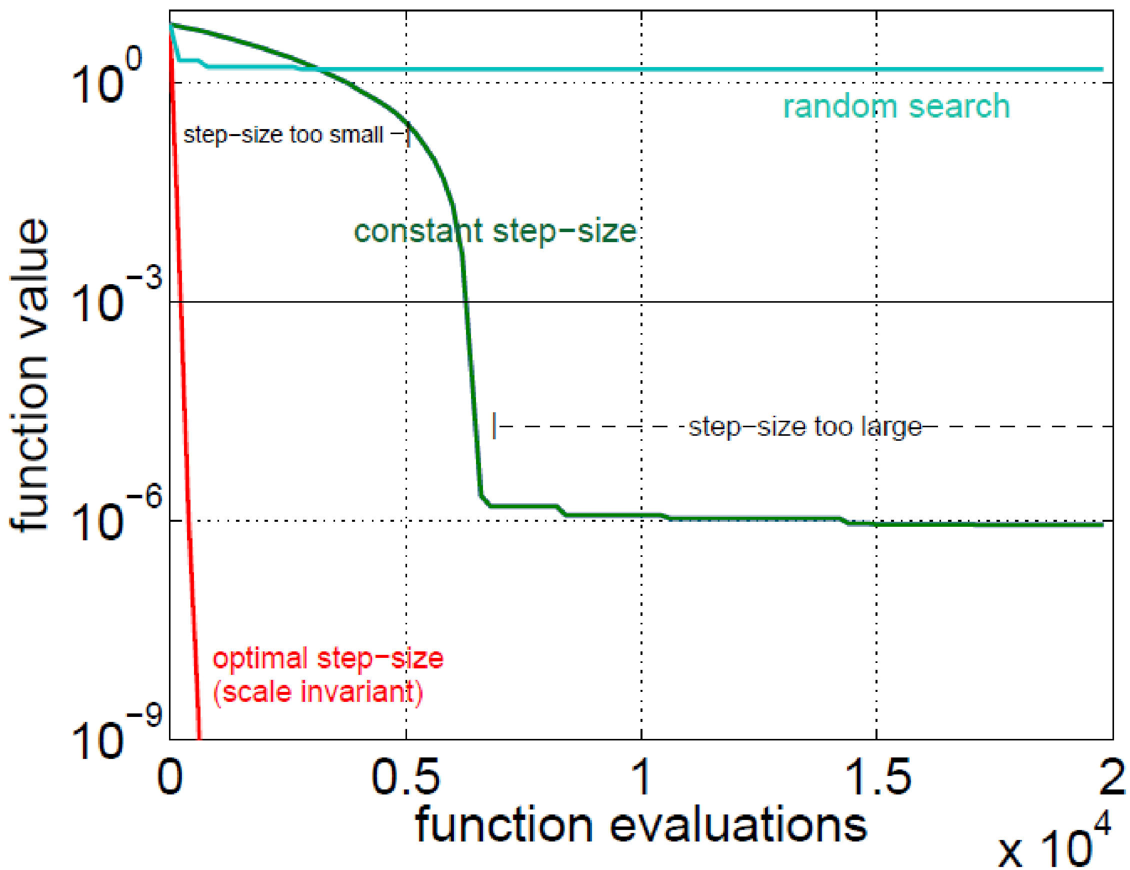
$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$$

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

from [Auger, p. 22]

Why Step-Size Adaptation?

Why Step-Size Control?

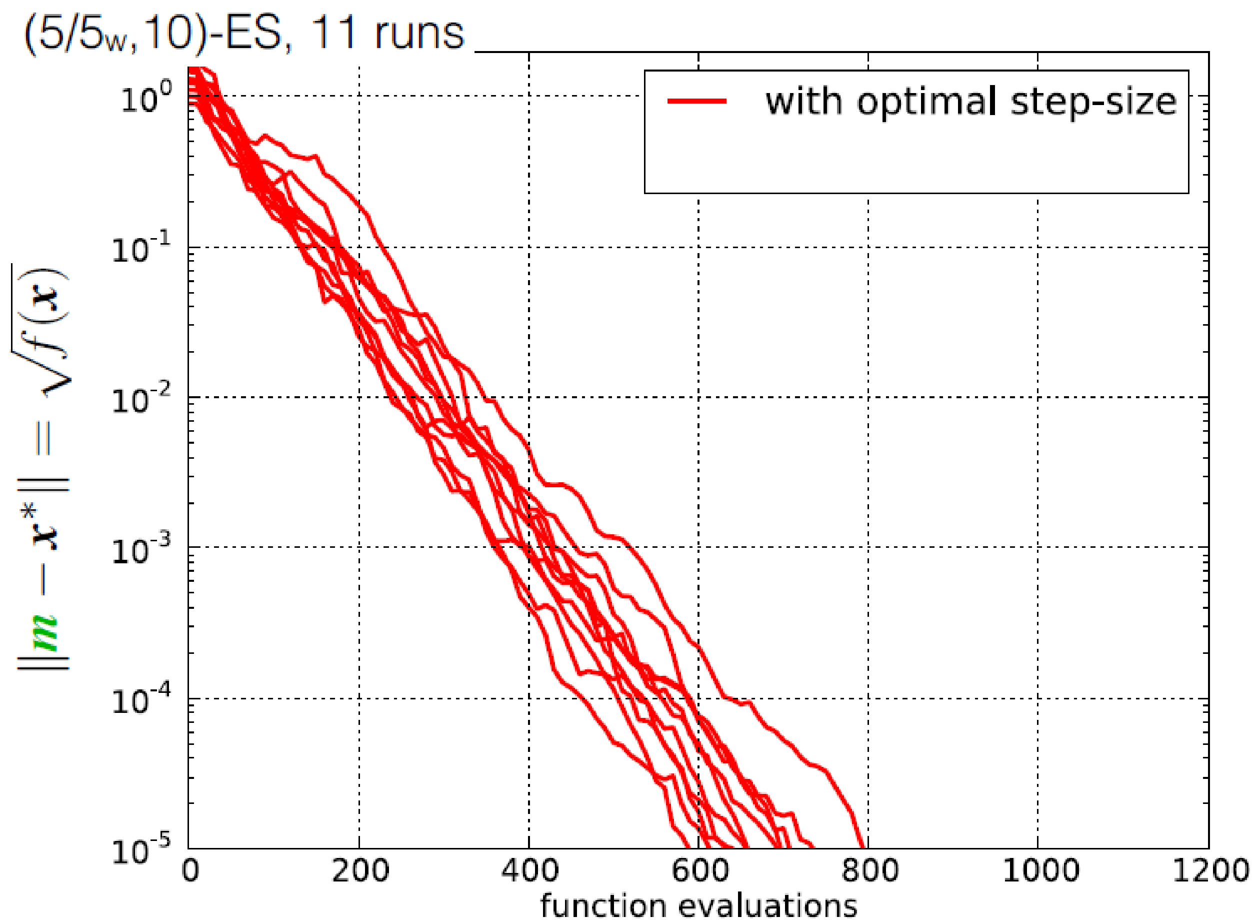


$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$$

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

from [Auger, p. 22]

Why Step-Size Control?



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

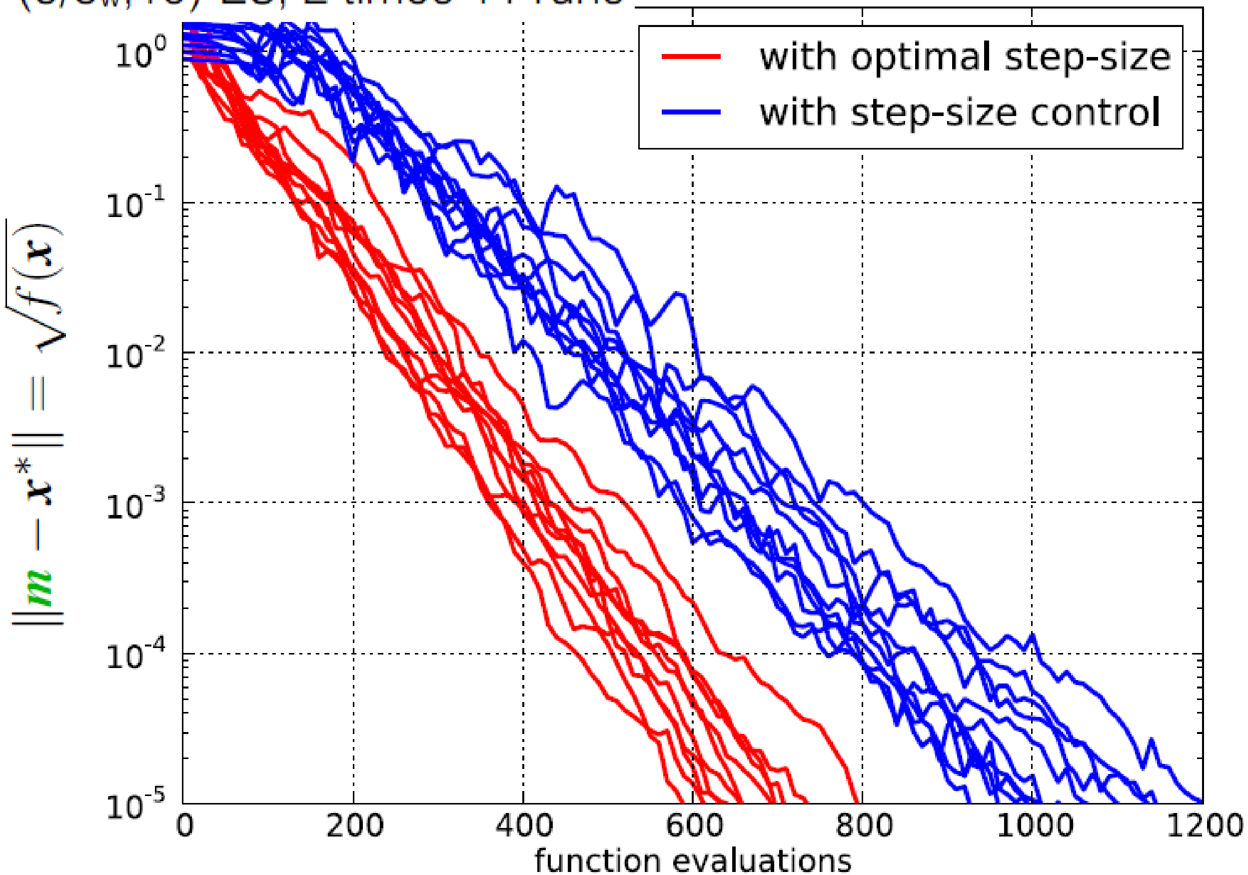
for $n = 10$ and
$\boldsymbol{x}^0 \in [-0.2, 0.8]^n$

with optimal step-size σ

from [Hansen, p. 47]

Why Step-Size Control?

(5/5w,10)-ES, 2 times 11 runs



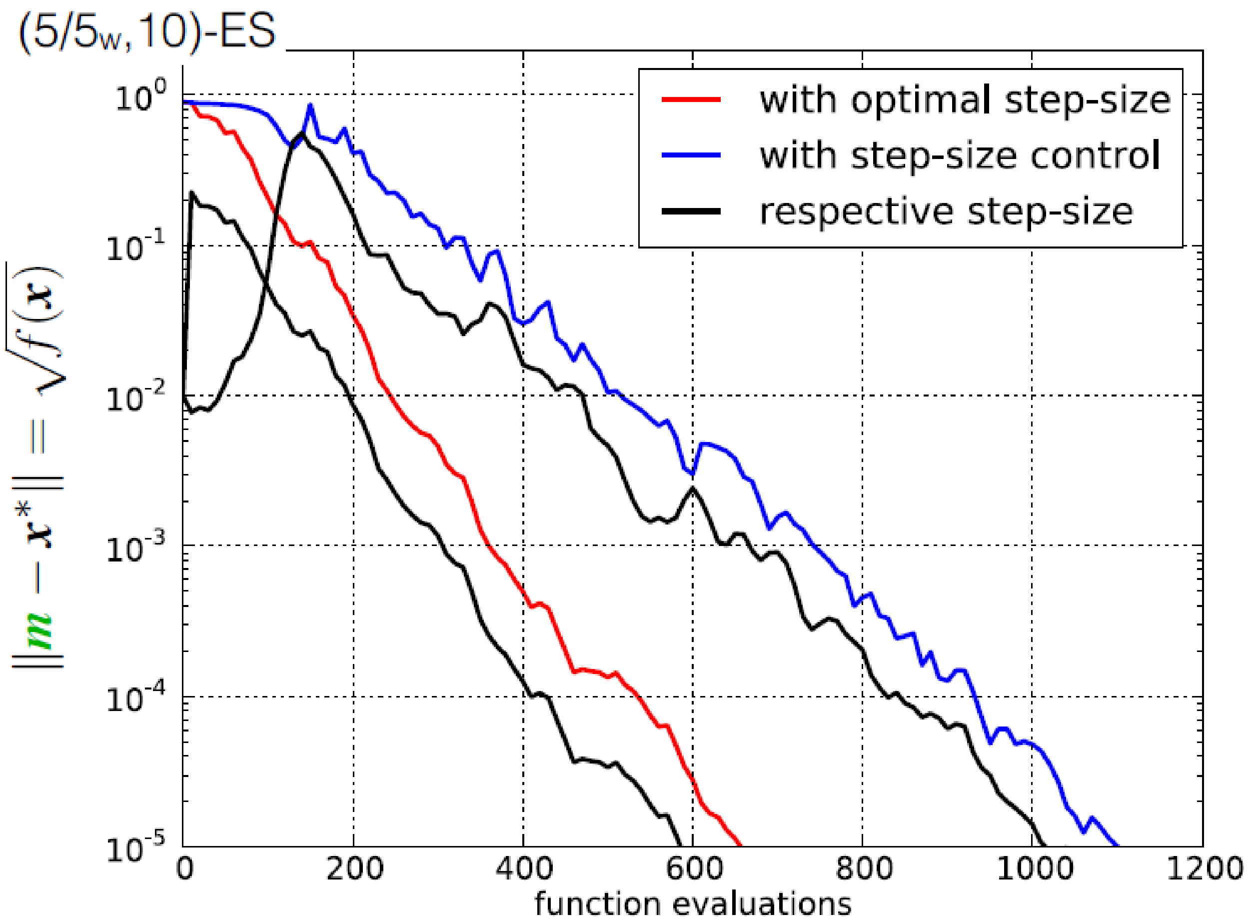
$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$$

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

from [Hansen, p. 48]

Why Step-Size Control?



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

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

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

from [Hansen, p. 49]

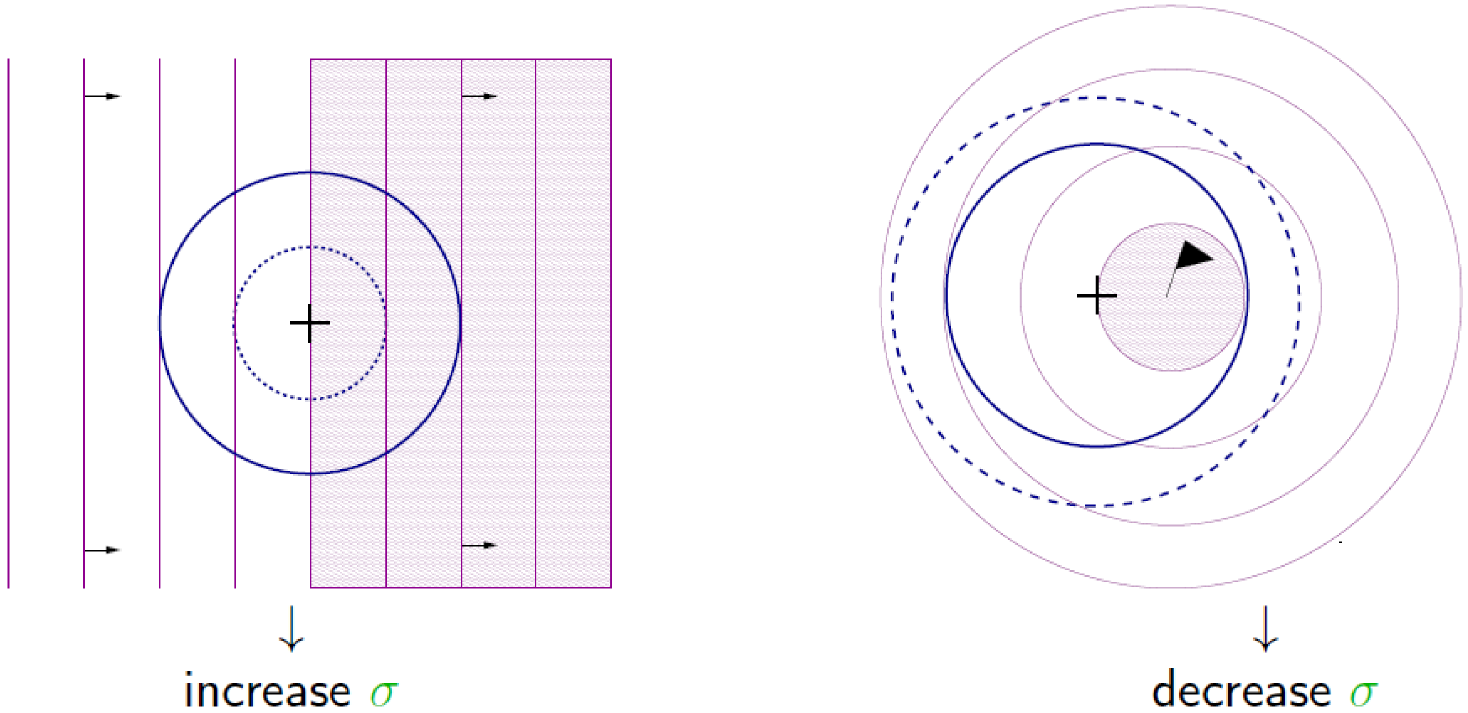
Adapting the Step-Size

- How to actually adapt the step-size during the optimization?

Most common:

- 1/5 success rule
- Cumulative Step-Size Adaptation (CSA, as in standard CMA-ES)
- others possible (Two-Point Adaptation, self-adaptive step-size, ...)

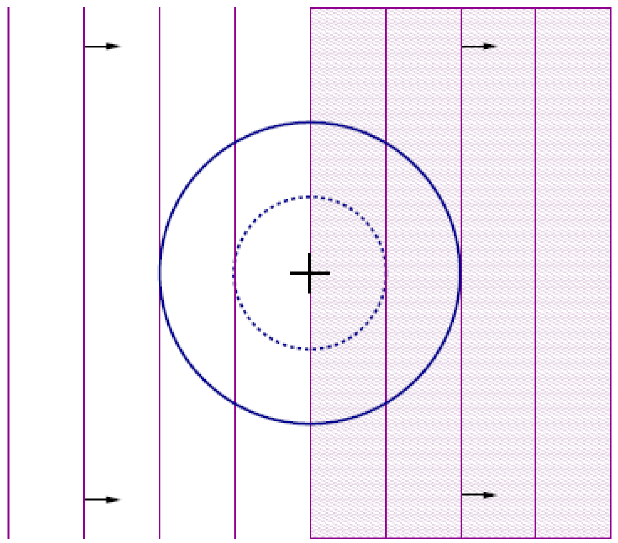
One-fifth success rule

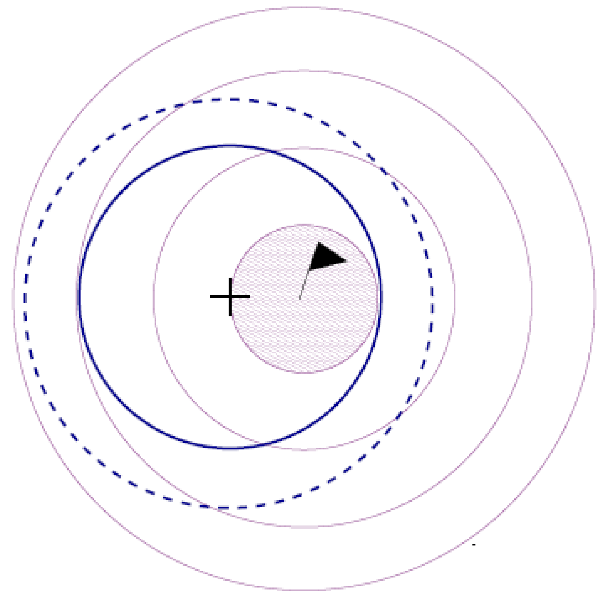


from [Auger, p. 32]

One-Fifth Success Rule

One-fifth success rule





Probability of success (p_s) — Probability of success (p_s)

$1/2$ — $1/5$ — "too small"

from [Auger, p. 33]

One-fifth success rule

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

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

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*

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

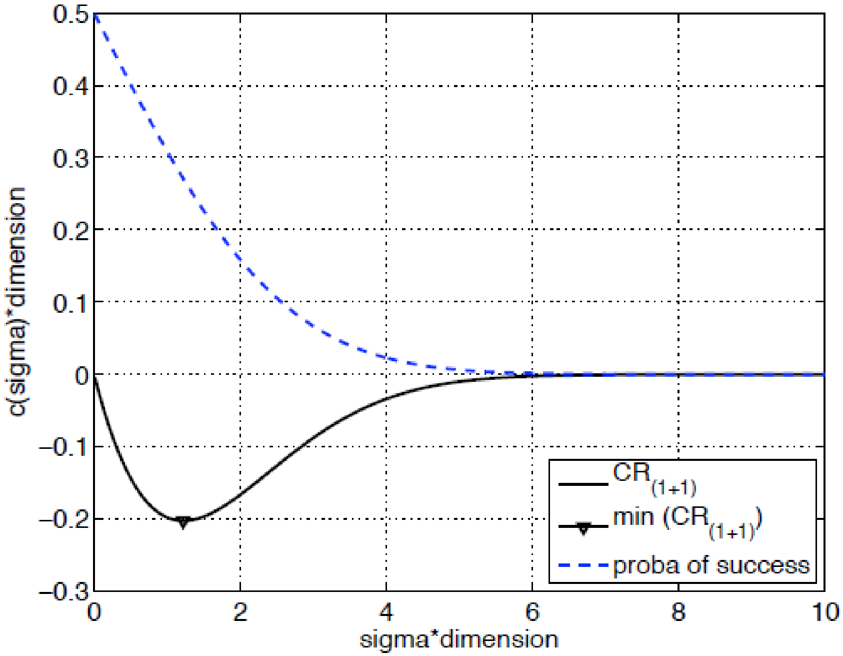
ELSE

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

from [Auger, p. 34]

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

from [Auger, p. 35]

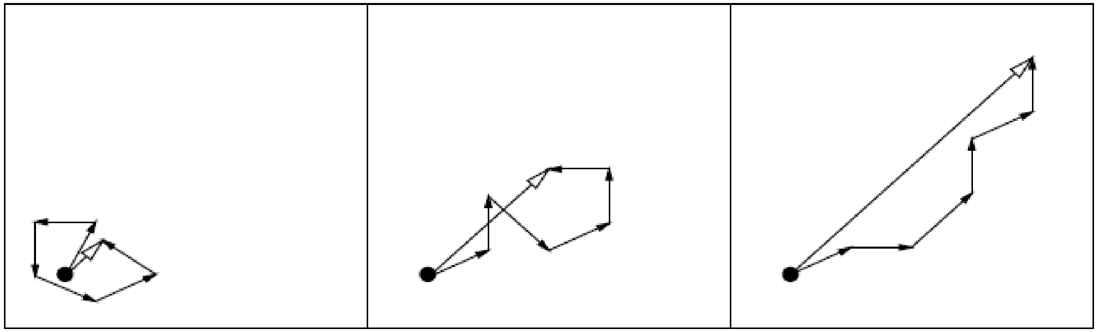
Path Length Control (CSA)

The Concept of Cumulative Step-Size Adaptation

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

Measure the length of the *evolution path*

the pathway of the mean vector $\boldsymbol{m}$ in the generation sequence



⇓		⇓
decrease σ		increase σ

from [Auger, p. 36]

Path Length Control (CSA)

The Equations

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

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

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

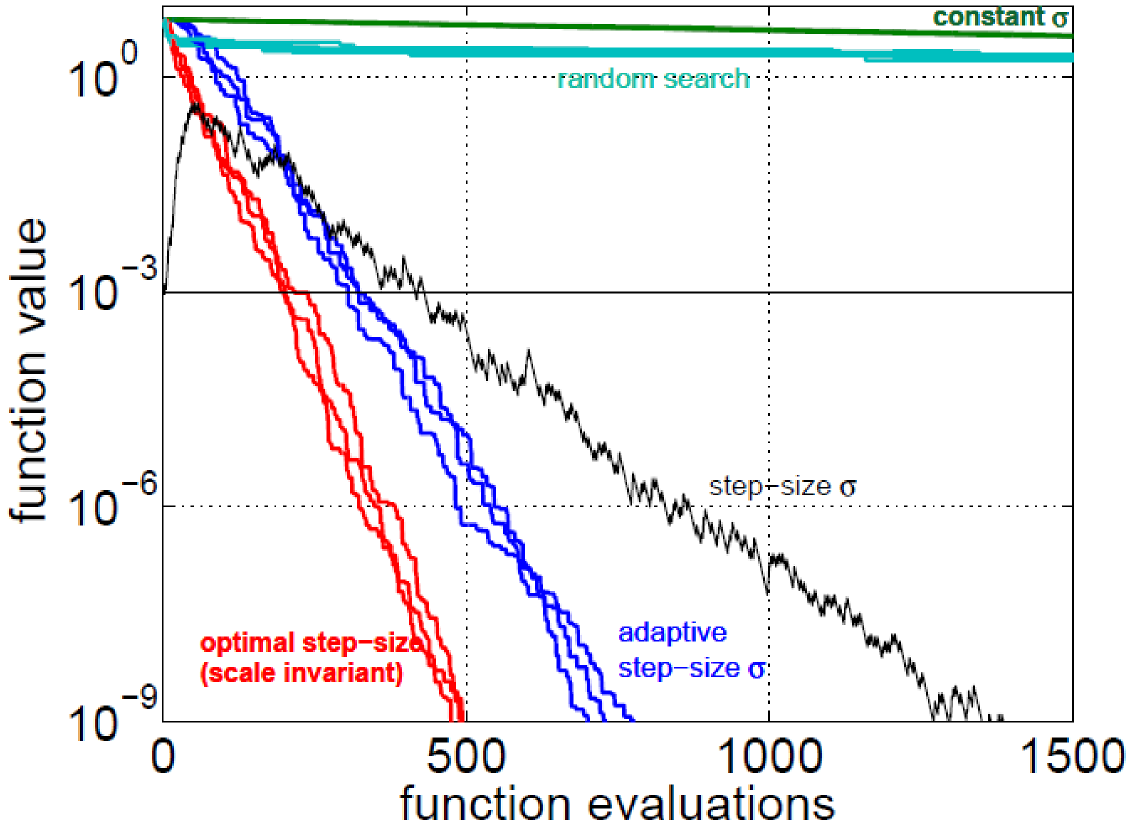
$$\sigma \leftarrow \sigma \times \underbrace{\exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p}_\sigma\|}{\mathsf{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)\right)}_{>1 \iff \|\boldsymbol{p}_\sigma\| \text{ is greater than its expectation}} \qquad \text{update step-size}$$

from [Auger, p. 37]

Cumulative Step-Size Adaptation (CSA)

Step-size adaptation

What is achived



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

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

Linear convergence

from [Auger, p. 38]

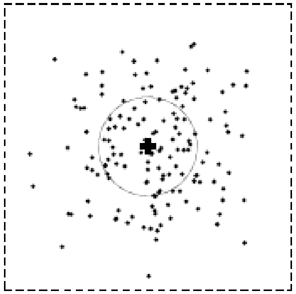
Covariance Matrix Adaptation

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 $\mathbf{C}$.

from [Auger, p. 40]

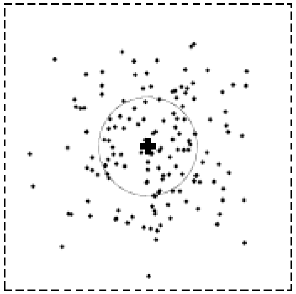
Evolution Strategies

Recalling

New search points are sampled normally distributed

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

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



where

- the mean vector $m \in \mathbb{R}^n$ represents the favorite solution
- the so-called step-size
- the covariance matrix
  of the distribution elli

...which is what we will see in the last lecture next Friday

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

from [Auger, p. 40]