Advanced Optimization
Lecture 5: Stochastic Algorithms (SGD & CMA-ES)

November 10, 2021

CentraleSupélec / ESSEC Business School

dimo.brockhoff@inria.fr

Dimo Brockhoff

Inria Saclay – Ile-de-France

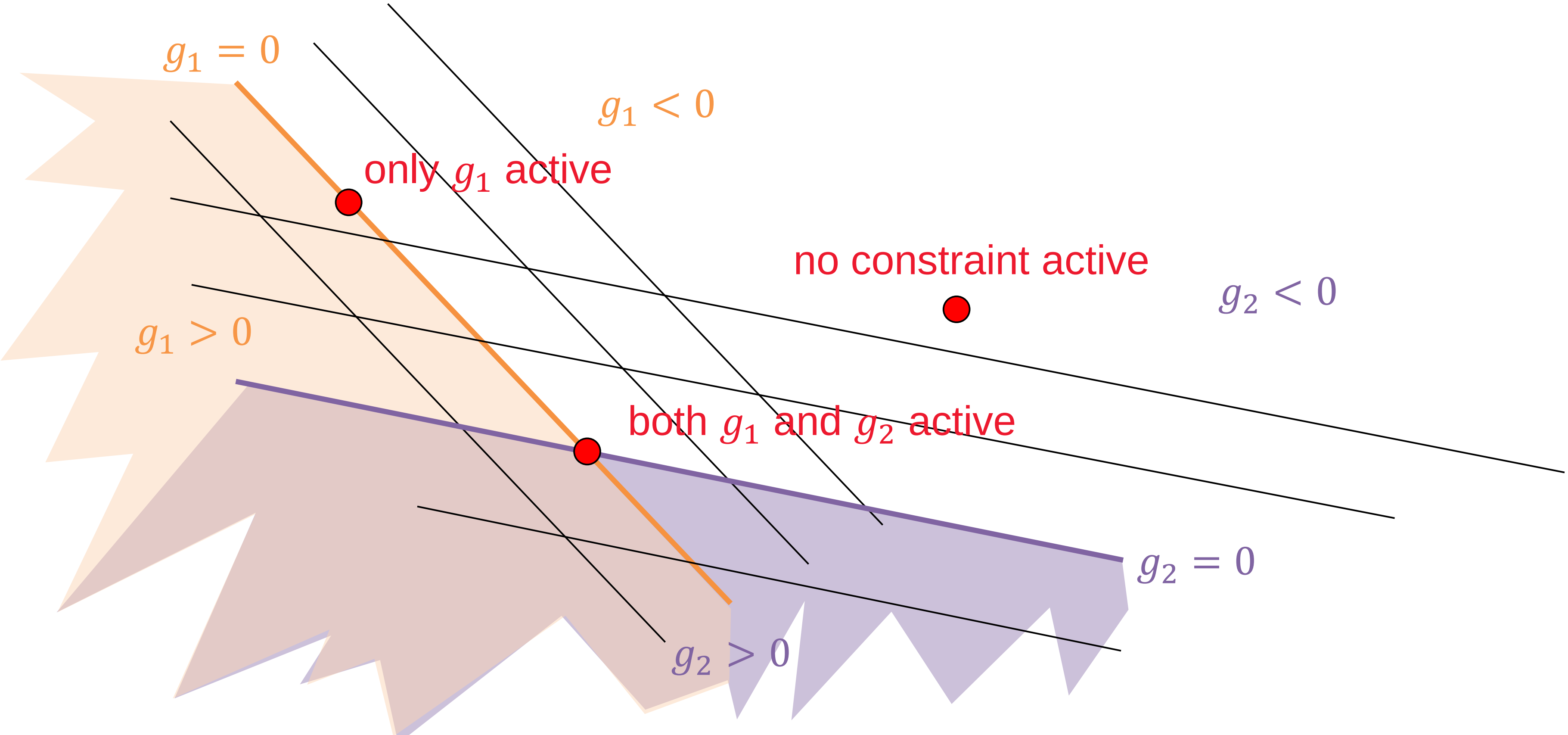
Course Overview

		Topic
Wed, 13.10.2021	PM	Introduction, examples of problems, problem types
Wed, 20.10.2021	PM	Continuous (unconstrained) optimization: convexity, gradients, Hessian, … [technical test Evalmee]
Wed, 27.10.2021	PM	Continous optimization II: [1st mini-exam] Constrained optimization: Lagrangian, optimality conditions
Wed, 03.11.2021	PM	gradient descent, Newton direction, quasi-Newton (BFGS) Linear programming: duality, maxflow/mincut, simplex algo
Wed, 10.11.2021	PM	Gradient-based and derivative-free stochastic algorithms: SGD and CMA-ES
Wed, 17.11.2021	PM	Other blackbox optimizers: Nelder-Mead, Bayesian optimization
Wed, 24.11.2021	PM	Benchmarking solvers: runtime distributions, performance profiles **[2nd mini-exam]**
Tue, 30.11.2021	23:59	Deadline open source project (PDF sent by email)
Wed, 01.12.2021	PM	Discrete optimization: branch and bound, branch and cut, k-means clustering
Wed, 15.12.2021	PM	Exam

Clarification: Active Constraints

Correct in slides, but maybe not clear enough with my examples:

An inequality constraint is active in point a if the constraint is 0 in a



List of Potential Issues for Group Project

Non-exhaustive list of course, but feasible and interesting tasks for those groups who have not yet decided on a topic:

- https://github.com/facebookresearch/nevergrad/issues/589
- https://github.com/numbbo/coco/issues/1594
- https://github.com/numbbo/coco/issues/1121
- https://github.com/numbbo/coco/issues/1836

Advanced Exercise

Also for today and next time:

- advanced exercise available
- topic: benchmarking and the COCO platform (PDF on Edunao)

Details on Continuous Optimization Lectures

Introduction to Continuous Optimization

- examples and typical difficulties in optimization

Mathematical Tools to Characterize Optima

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

Gradient-based Algorithms

- gradient descent
- quasi-Newton method (BFGS) and invariances

Linear programming, duality

Learning in Optimization / Optimization in Machine Learning

- Stochastic gradient descent (SGD) + Adam
- CMA-ES (adaptive algorithms / Information Geometry)
- Other derivative-free algorithms: Nelder-Mead, Bayesian opt.

Linear Optimization

[optimization with linear objective and linear constraints functions]

Linear programming = linear optimization

Find a vector x that

- maximizes $c^T x$
- s.t. $Ax \leq b$
- and $x \geq 0$

How to Solve Linear Programs?

Simplex method (Dantzig, 1940s)

- fast in practice, but exponential in worst case

Interior point methods

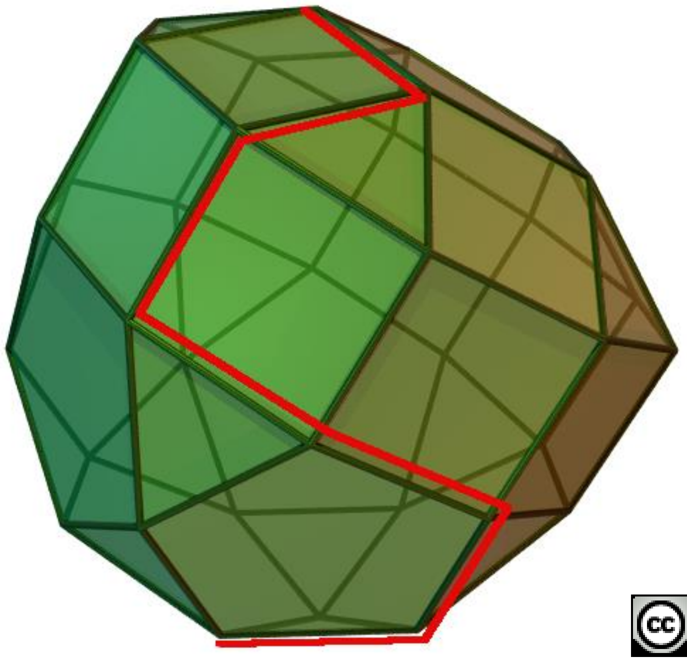
- Khachiyan, 1979: first polynomial algorithm, $O(n^6 L)$

  n: #variables, L: #input bits
- Karmarkar, 1984: $O(n^{3.5} L)$
- Vaidya, 1989: $O(n(n+d)^{1.5} L) = O(n^{2.5} L)$ for constant d

  d: #constraints

Idea Behind Simplex Algorithm

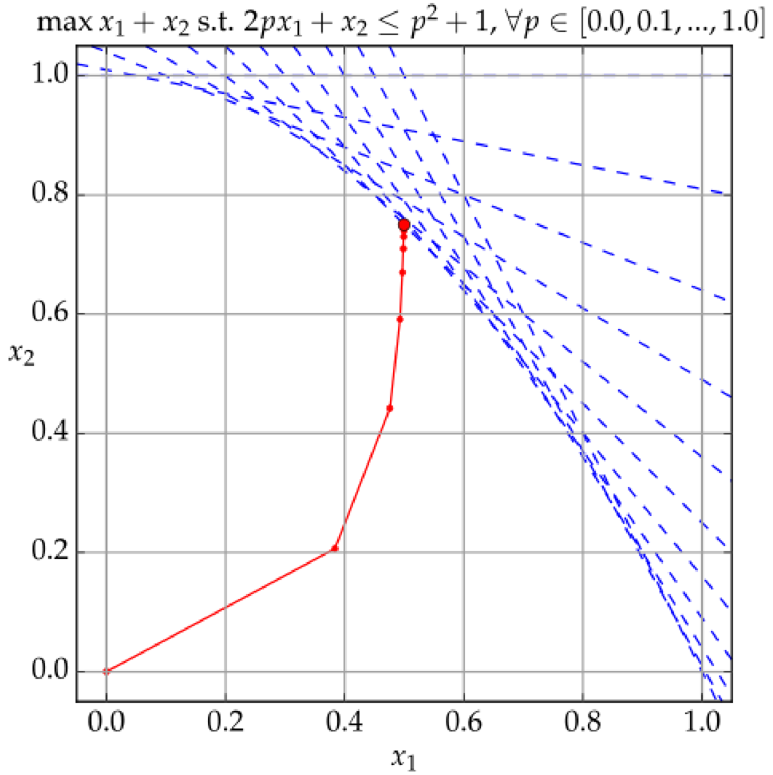
- Move along linear facets from corner to corner
- If corner not optimal, there is always a neighbor which is better
- Corresponds to equality constraints (inequality constraints need to be transformed accordingly via “slack variables”)



Idea Behind Interior Point Methods

- evaluate inside the simplex and move towards the edges
- works with inequality constraints
- solve $f(x) - 1/t \sum_{i=1}^{m} \log(g_i(x))$ iteratively with increasing t

given m inequality constraints $g_i(x) \geq 0$



Conclusions

I hope it became clear...

... what linear programming is and

... what are the ideas behind the simplex algorithm and interior point methods

Next:

idea of duality

stochastic algorithms: stochastic gradient descent and CMA-ES

Duality

[how to solve an unconstrained problem instead of a constrained one]

based on:

https://www.youtube.com/watch?v=4OifjG2kIJQ

Given: The Primal (A Constrained Opt. Problem)

Primal Problem:

- $\min f(x)$ $[f\colon \mathbb{R}^n \to \mathbb{R}]$
- such that: $h(x) = 0$ and $\mathrm{g}(x) \leq 0$

  $[h\colon \mathbb{R}^n \to \mathbb{R}^m]$ $[g\colon \mathbb{R}^n \to \mathbb{R}^p]$

Reformulate via Lagrange multipliers/penalties/dual or slack variables:

- Associate to each equality constraint a λ_i and to each inequality constraint a μ_i
- Lagrangian: $L(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x)$

What to Do With the Lagrangian?

Can be used to compute best x given a λ and a μ with less constraints:

Dual Function:

- q: $\mathbb{R}^{m+p} \to \mathbb{R}$
- $q(\lambda, \mu) = \min_{x \in \mathbb{R}^n} L(x, \lambda, \mu)$
  such that $\mu \geq 0$ [otherwise $\mu^T g(x) < 0$ for infeasible x]

And finally to compute a lower bound on $f(x^*)$:

Theorem (dual bound)

- Let x^* be the minimum of the primal. If $\lambda \in \mathbb{R}^m$ and $0 \leq \mu \in \mathbb{R}^p$, then $q(\lambda, \mu) \leq f(x^*)$.

Finding the Best Lower Bound

Dual Problem:

- $\max_{\lambda,\mu} q(\lambda,\mu)$
- given $\mu \geq 0$ and $\underbrace{(\lambda,\mu) \in \{\lambda,\mu \mid q(\lambda,\mu) > -\infty\}}_{\text{do not allow unbounded solutions!}}$

Relations Between Primal and Dual

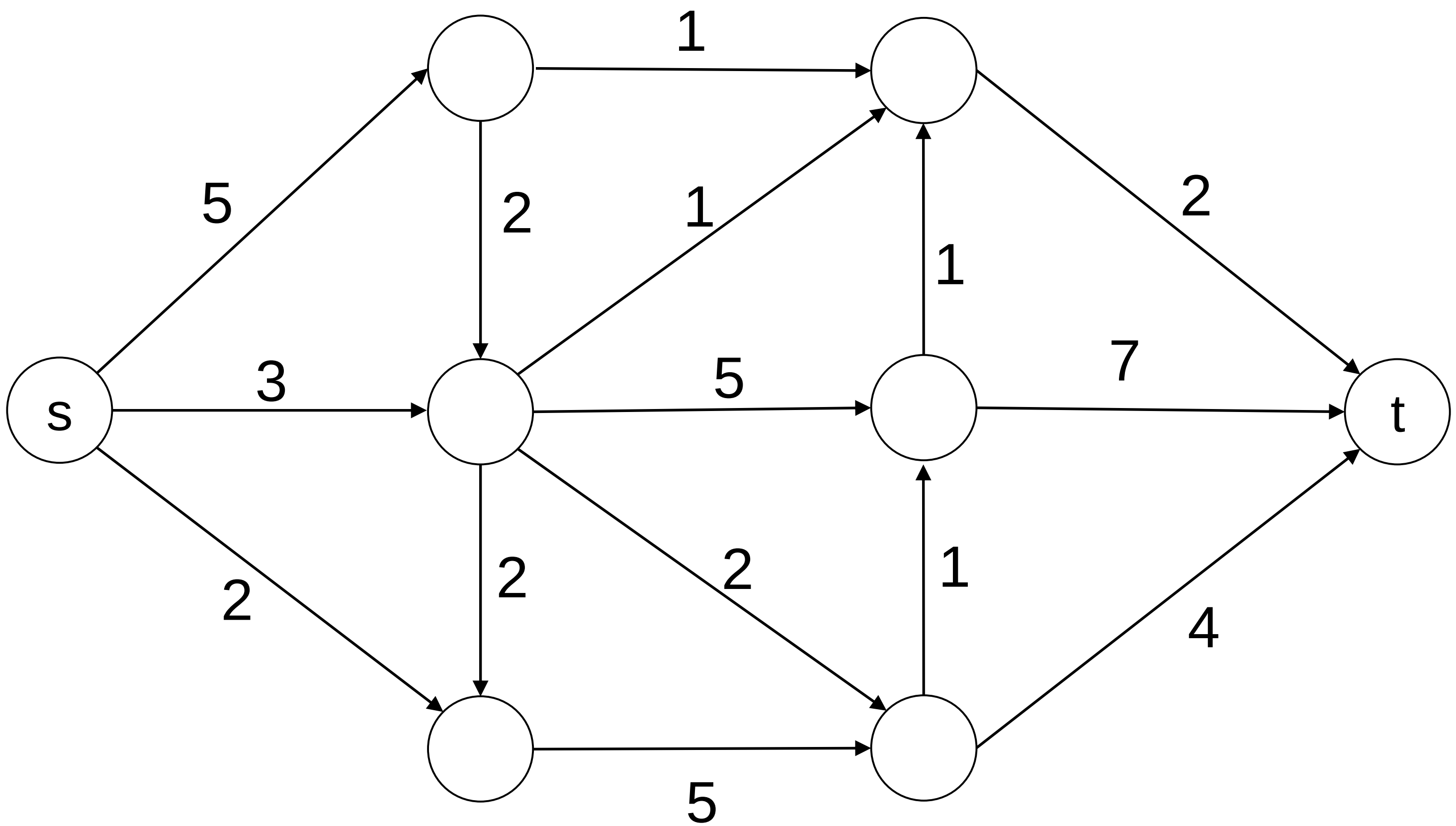
Dual problem

Primal problem

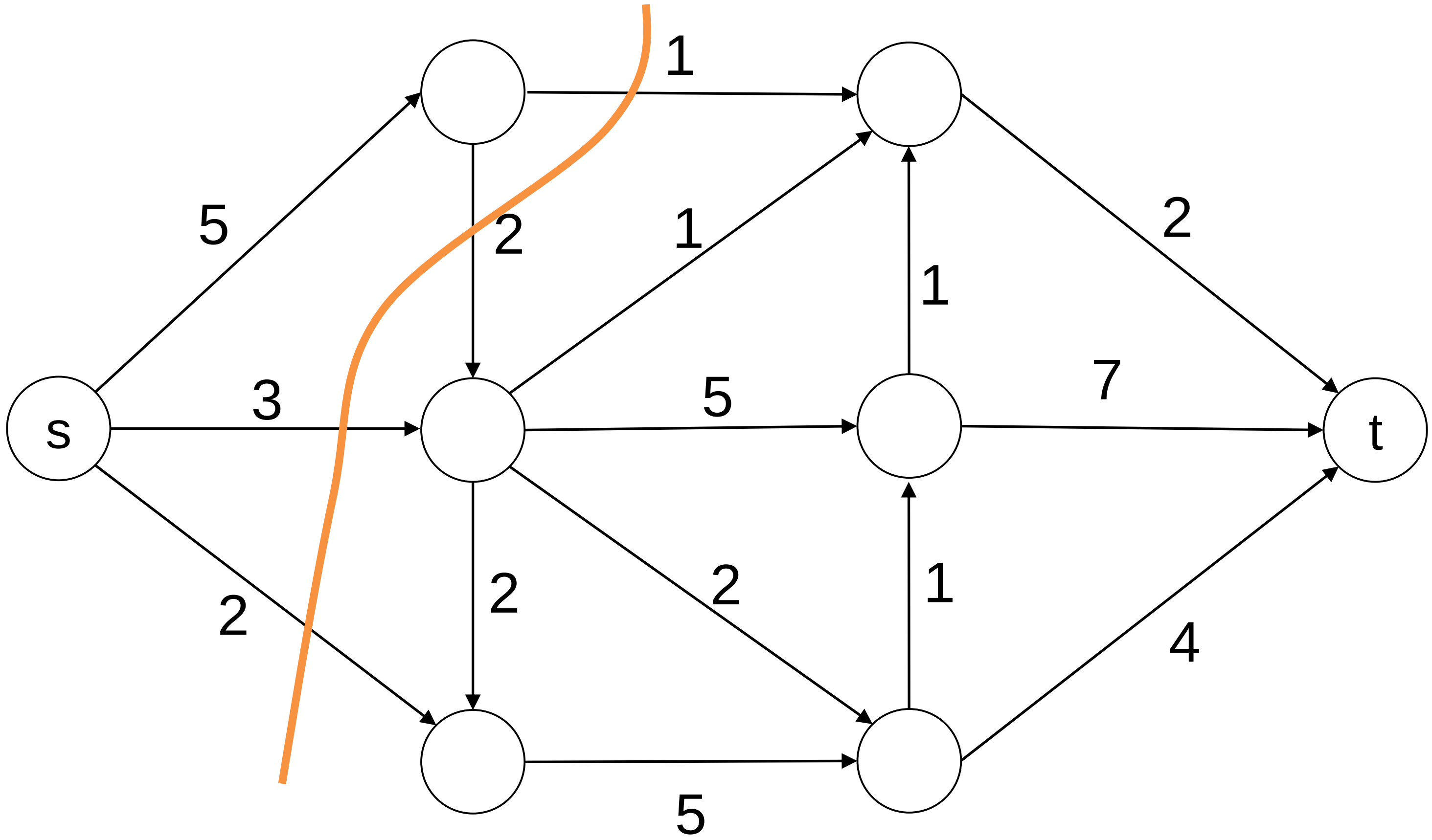
	Optimal	Unbounded	Infeasible
Optimal	YES	NO	NO
Unbounded	NO	NO	YES
Infeasible	NO	YES	YES

more details in https://www.youtube.com/watch?v=4OifjG2kIJQ

Application of Duality: Max Flow = Min Cut



Application of Duality: Max Flow = Min Cut



Stochastic Algorithms

algorithms using randomness

stochastic gradient descent (SGD)

Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

Stochastic Algorithms

algorithms using randomness

stochastic gradient descent (SGD) [next week]

Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

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
  - ...

Nelder Mead

aka simplex downhill

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

$$x_e = x_o + \gamma\,(x_r - x_o)(\gamma > 0)$$

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 $i \in \{2, \dots, n+1\}$ $(\sigma < 1)$ and go to 1)

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

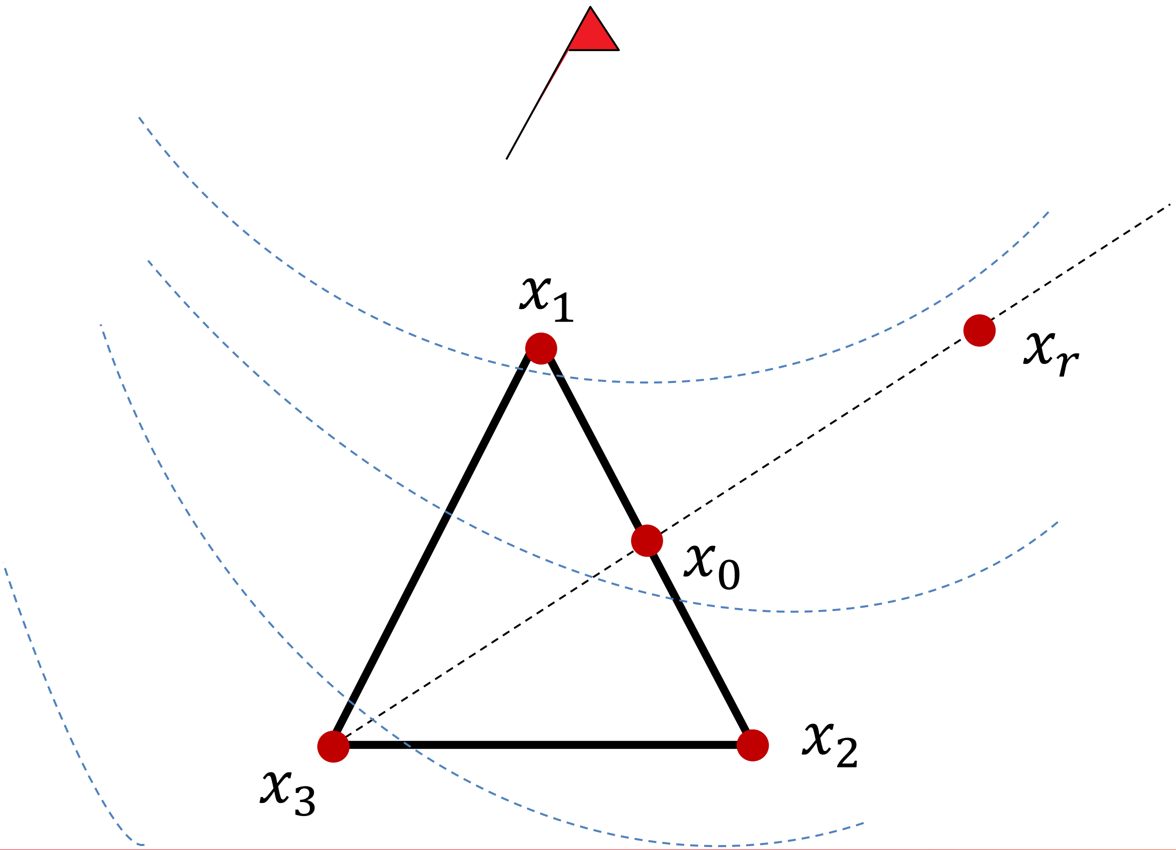
Nelder-Mead: Reflection

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)



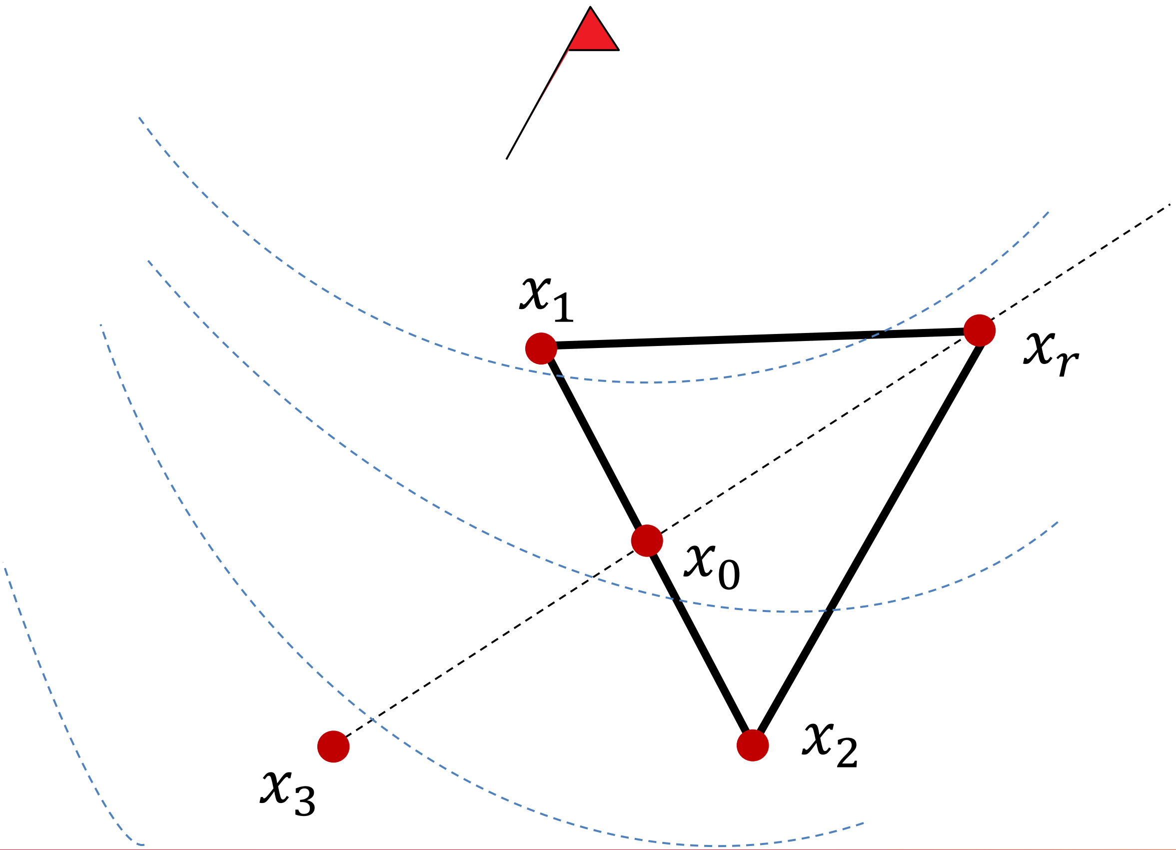
Nelder-Mead: Reflection

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

3) Reflection

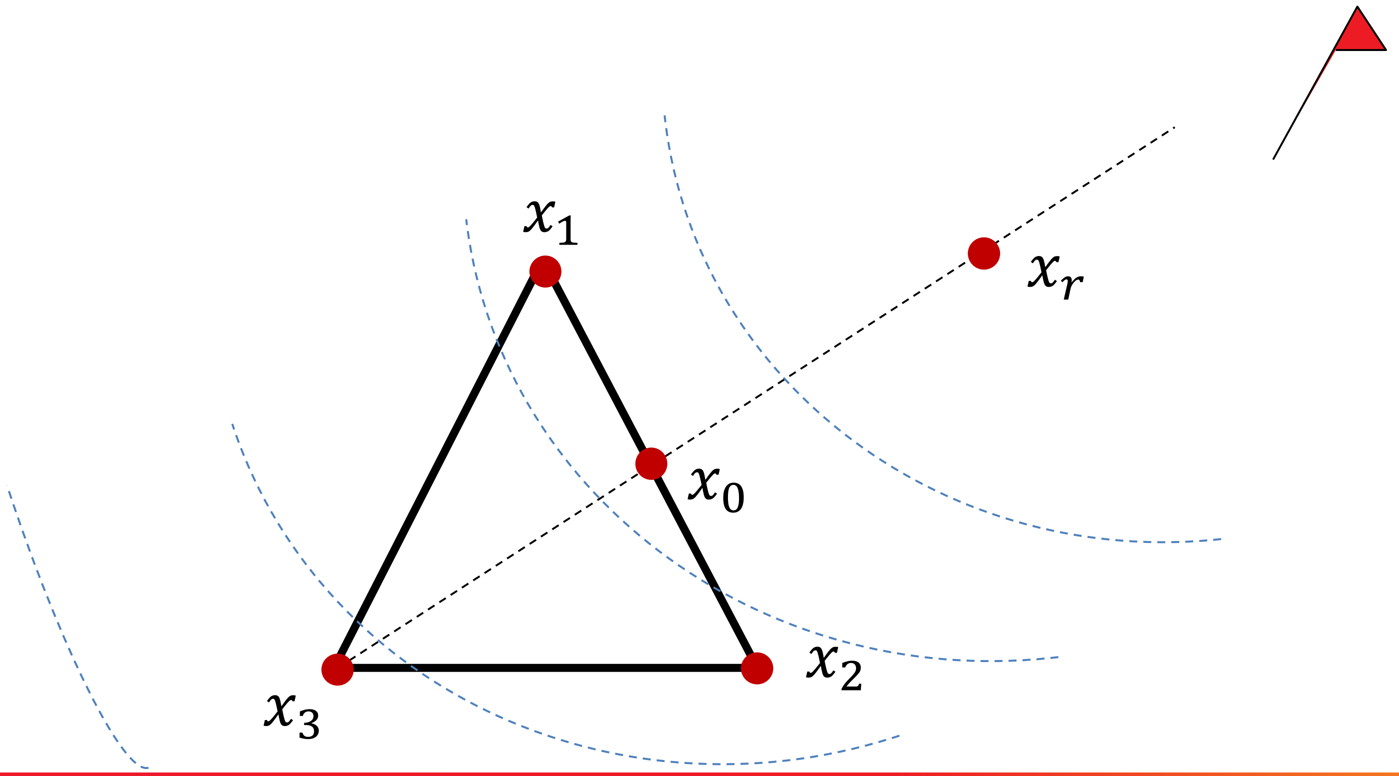
Compute reflected point $x_r = x_o + \alpha\,(x_o - x_{n+1})\ (\alpha > 0)$

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



Nelder-Mead: Reflection

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)



Nelder-Mead: Expansion

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

4) Expansion

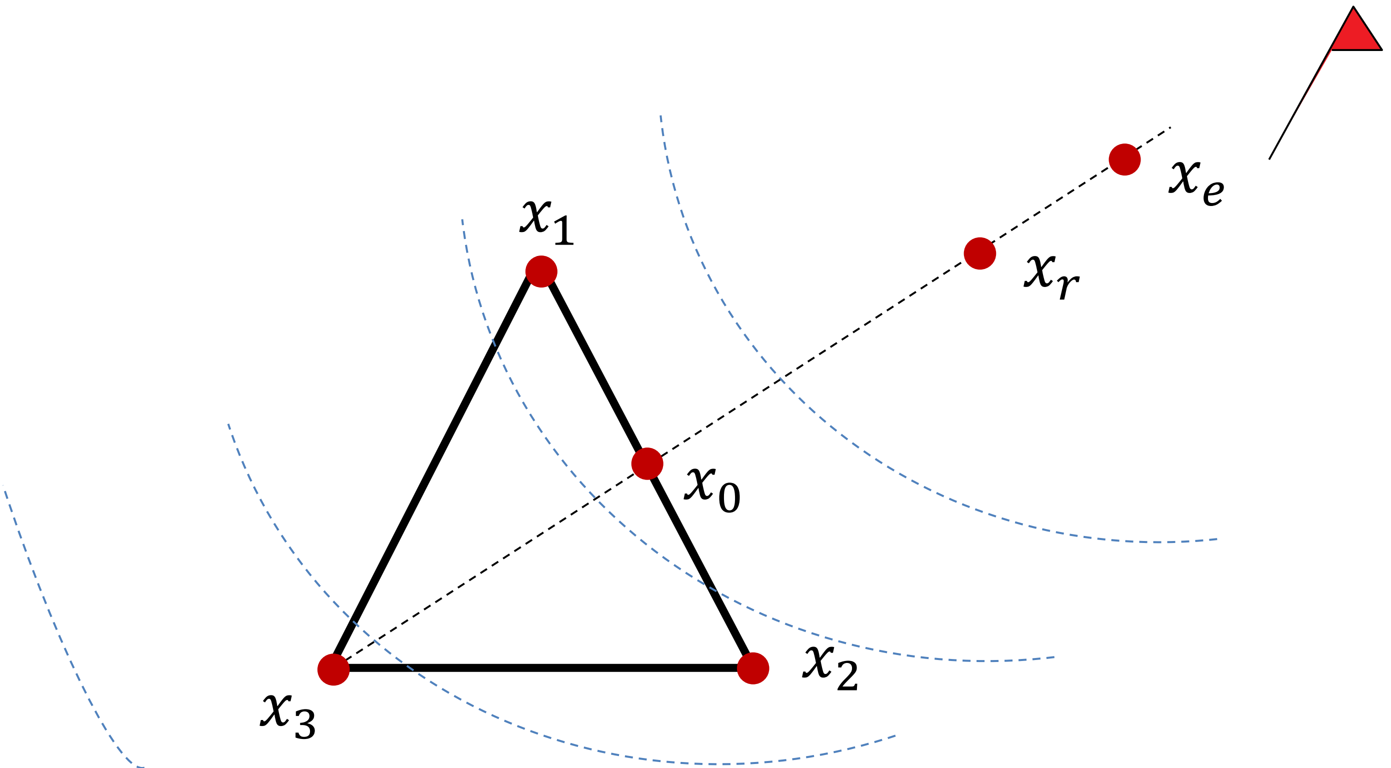
If x_r is the best point so far: compute the expanded point

$$x_e = x_o + \gamma (x_r - x_o) (\gamma > 0)$$

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

Else $x_{n+1} := x_r$ and go to 1)

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



Nelder-Mead: Expansion

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

4) Expansion

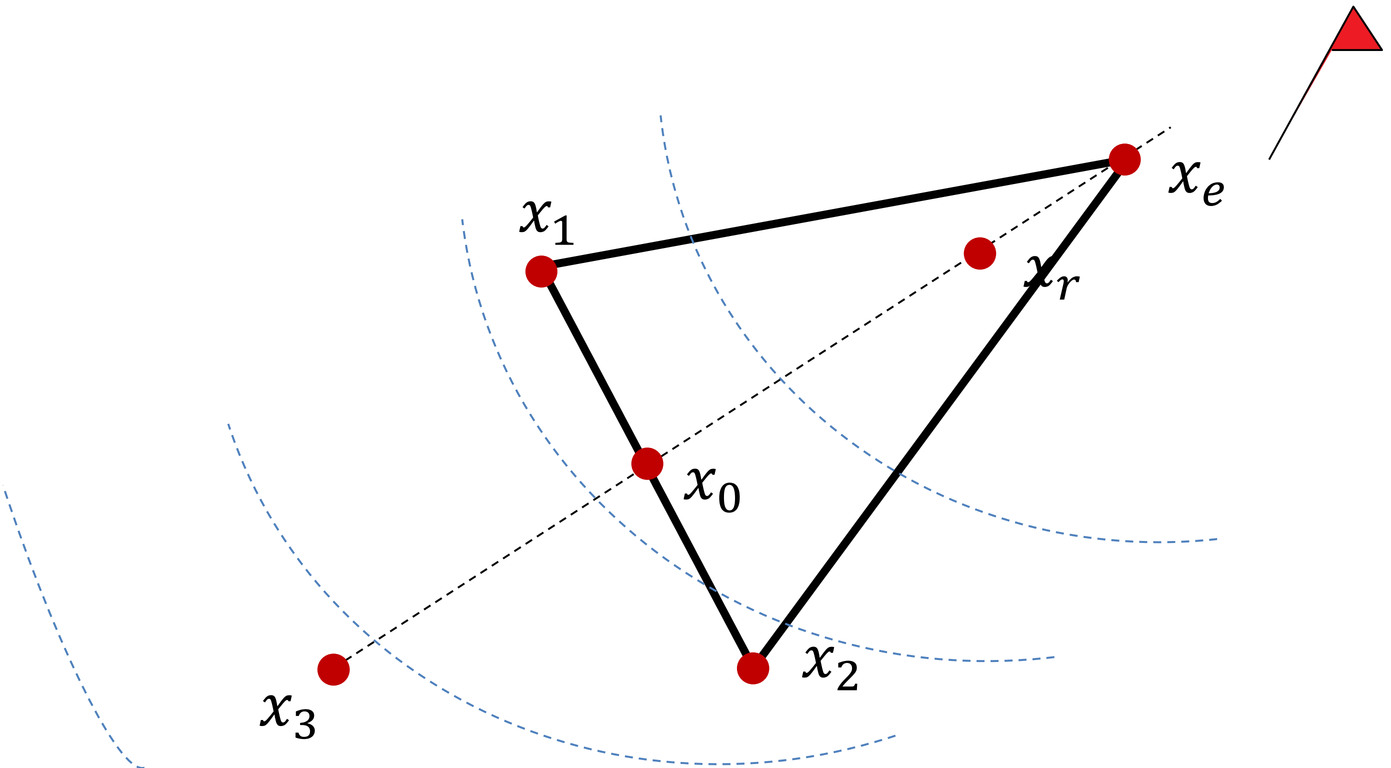
If x_r is the best point so far: compute the expanded point

$$x_e = x_o + \gamma\,(x_r - x_o)(\gamma > 0)$$

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)



Nelder-Mead: Expansion

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

4) Expansion

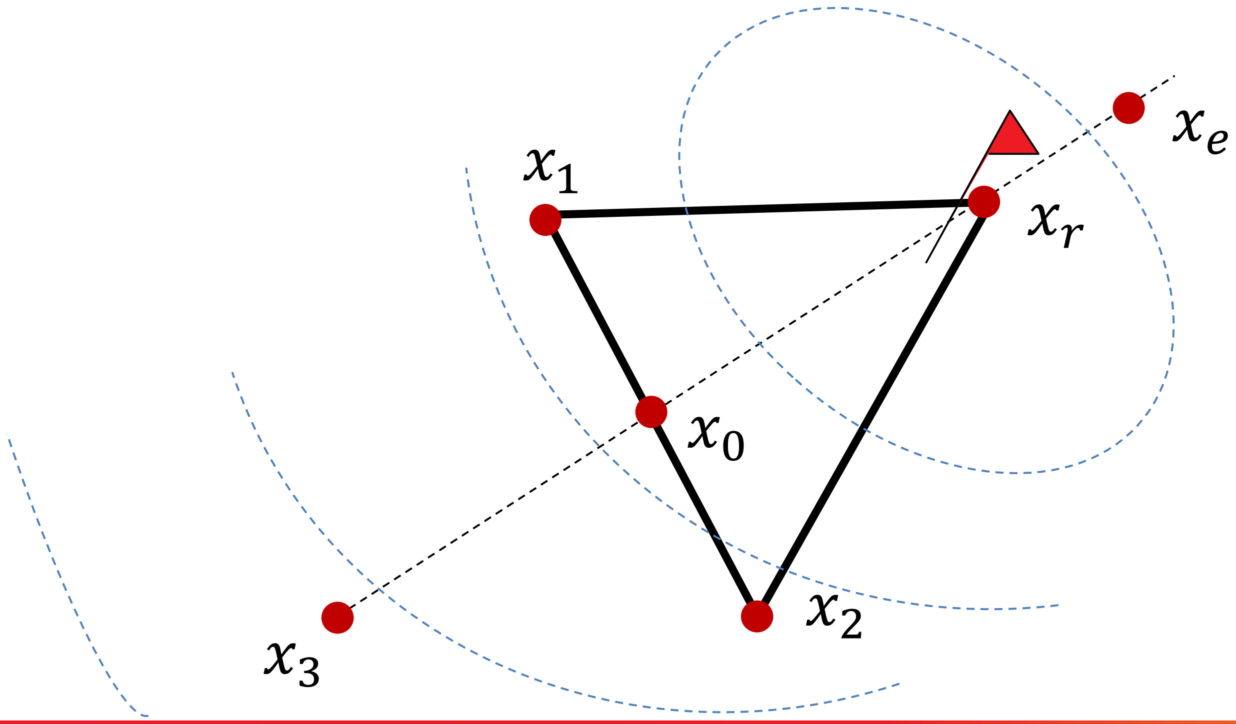
If x_r is the best point so far: compute the expanded point

$$x_e = x_o + \gamma\,(x_r - x_o)(\gamma > 0)$$

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)



Nelder-Mead: Expansion

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

4) Expansion

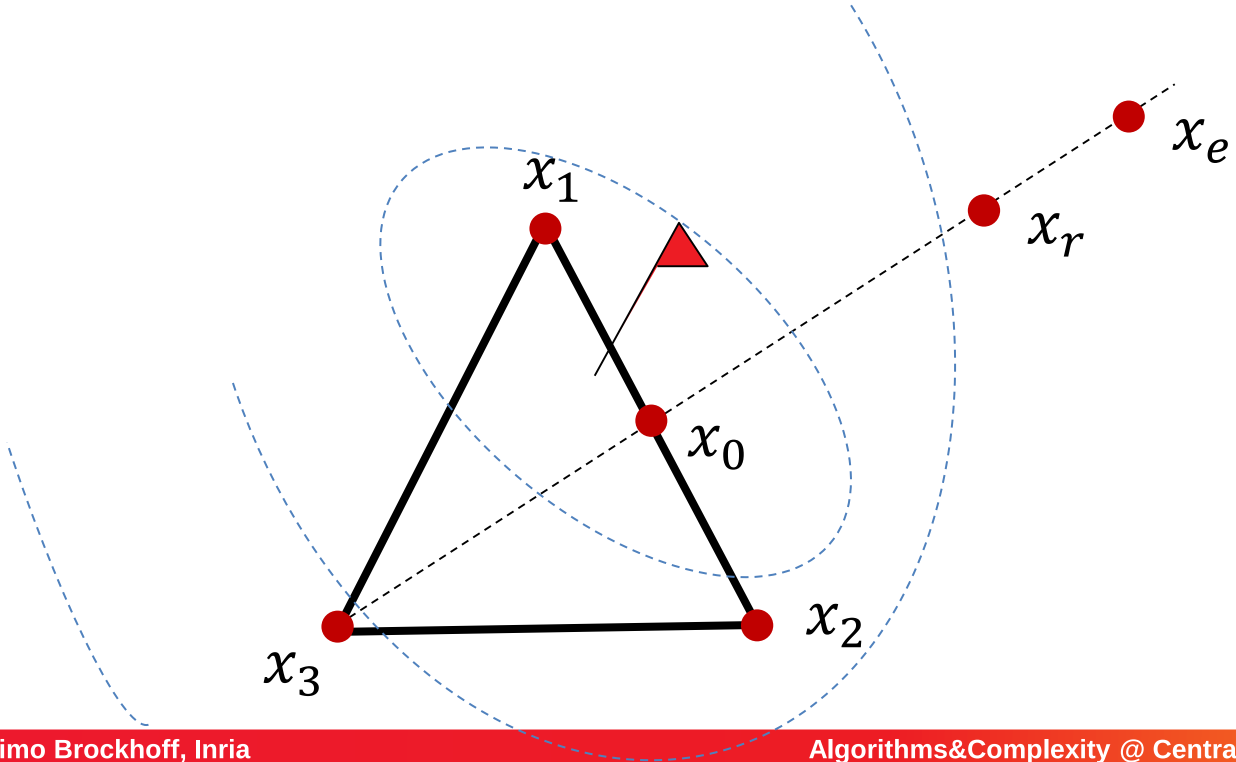
If x_r is the best point so far: compute the expanded point

$$x_e = x_o + \gamma\,(x_r - x_o)(\gamma > 0)$$

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

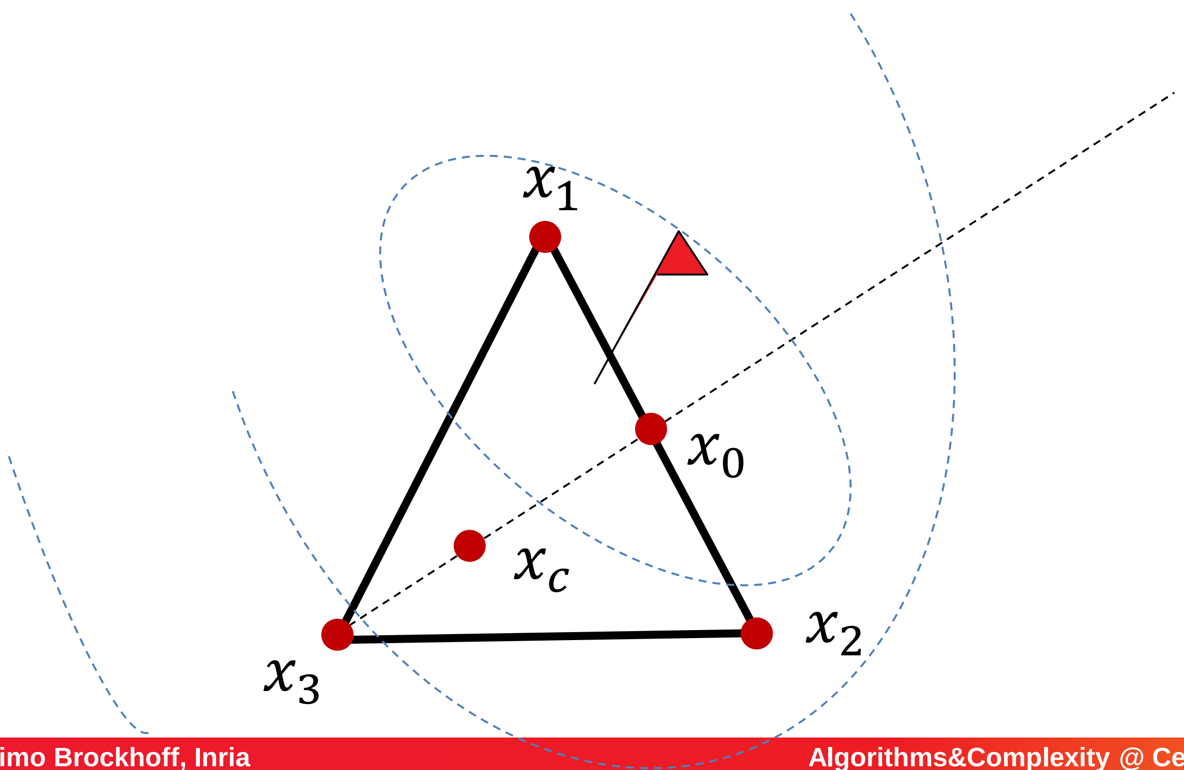
Else $x_{n+1} := x_r$ and go to 1)

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



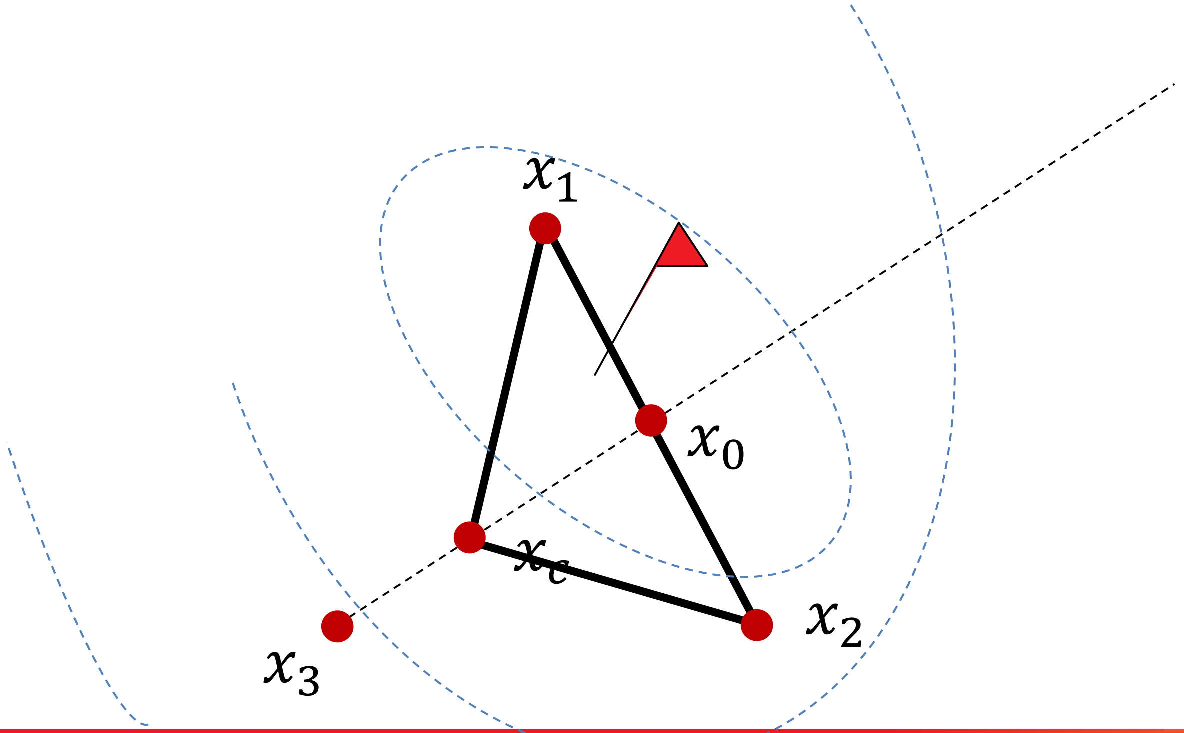
Nelder-Mead: Expansion

2) Calculate x_o, the centroid of all points except x_{n+1}.
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 $f(x_c) < f(x_{n+1})$: $x_{n+1} := x_c$ and go to 1)
Else go to 6)



Nelder-Mead: Expansion

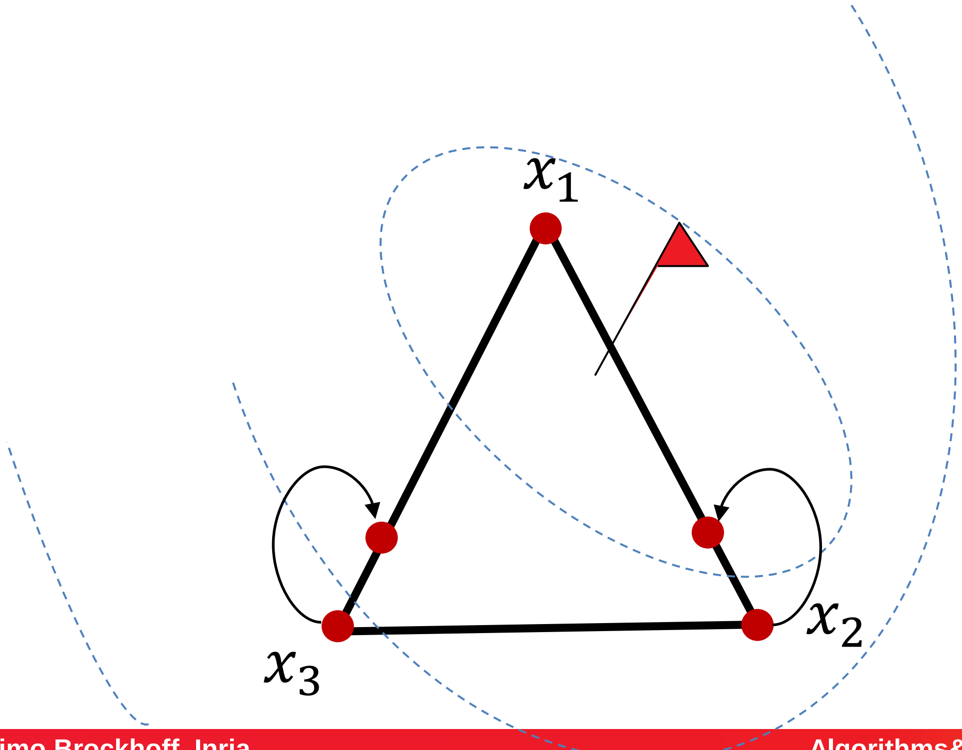
2) Calculate x_o, the centroid of all points except x_{n+1}.
5) Contraction (here: $f(x_r) \geq f(x_n)$)
  Compute contracted point $x_c = x_o + \rho(x_{n+1} - x_o)$ $(0 < \rho \leq 0.5)$
  If $f(x_c) < f(x_{n+1})$: $x_{n+1} := x_c$ and go to 1)
  Else go to 6)



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

6) Shrink

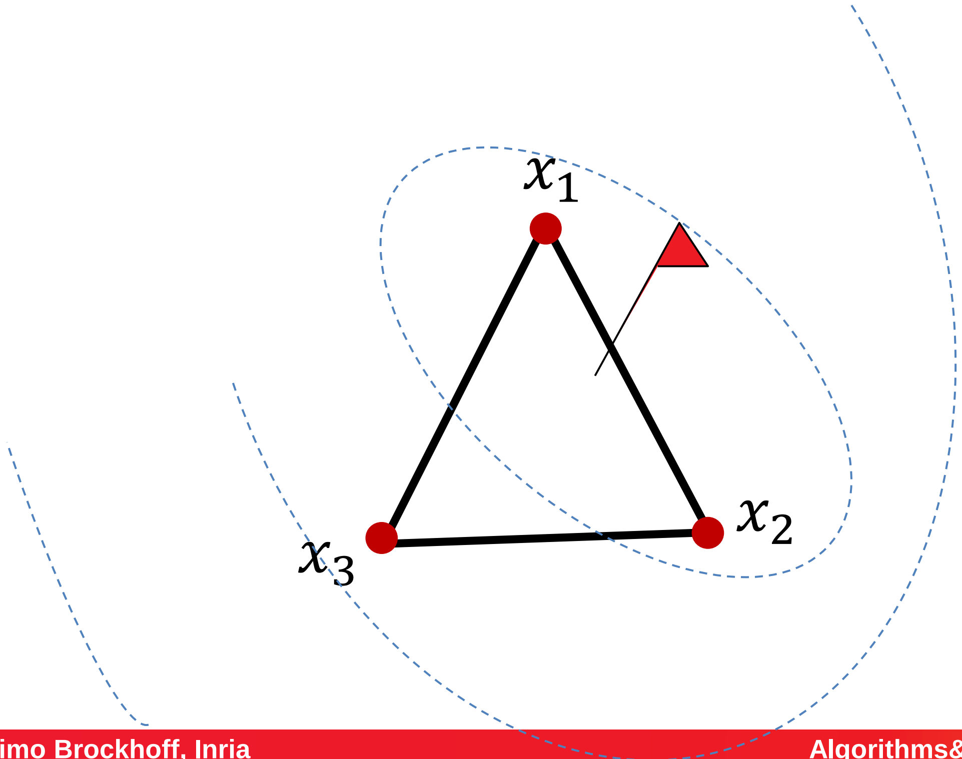
$x_i = x_1 + \sigma(x_i - x_1)$ for all $i \in \{2, \ldots, n+1\}$ and go to 1)



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

6) Shrink

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



Nelder-Mead: Standard Parameters

- reflection parameter : $\alpha = 1$
- expansion parameter: $\gamma = 2$
- contraction parameter: $\rho = \frac{1}{2}$
- shrink parameter: $\sigma = \frac{1}{2}$

some visualizations of example runs can be found here:
https://en.wikipedia.org/wiki/Nelder%E2%80%93Mead_method

Important to Note

- Nelder-Mead mainly good in (very) low dimension

  we’ll see this in the benchmarking lecture

- originally proposed algorithm shows shrinking simplex behavior

  hence, newer implementations try to avoid this:

  [Hansen 2009] [Doerr et al. 2009], COBYLA [Powell 1994]

[Hansen 2009] Nikolaus Hansen: "Benchmarking the Nelder-Mead downhill simplex algorithm with many local restarts“. In: *Genetic and Evolutionary Computation Conference Companion*, 2009.

[Doerr et al. 2009] Benjamin Doerr, Mahmoud Fouz, Martin Schmidt, and Magnus Wahlstrom: “BBOB: Nelder-Mead with resize and halfruns”. In: *Genetic and Evolutionary Computation Conference Companion*, 2009.

[Powell 1994] Michael J. D. Powell: “A direct search optimization method that models the objective and constraint functions by linear interpolation“. In: Advances in Optimization and Numerical Analysis, Kluwer Academic, Dordrecht, pp 51–67, 1994.

Covariance Matrix Adaptation Evolution Strategy (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:

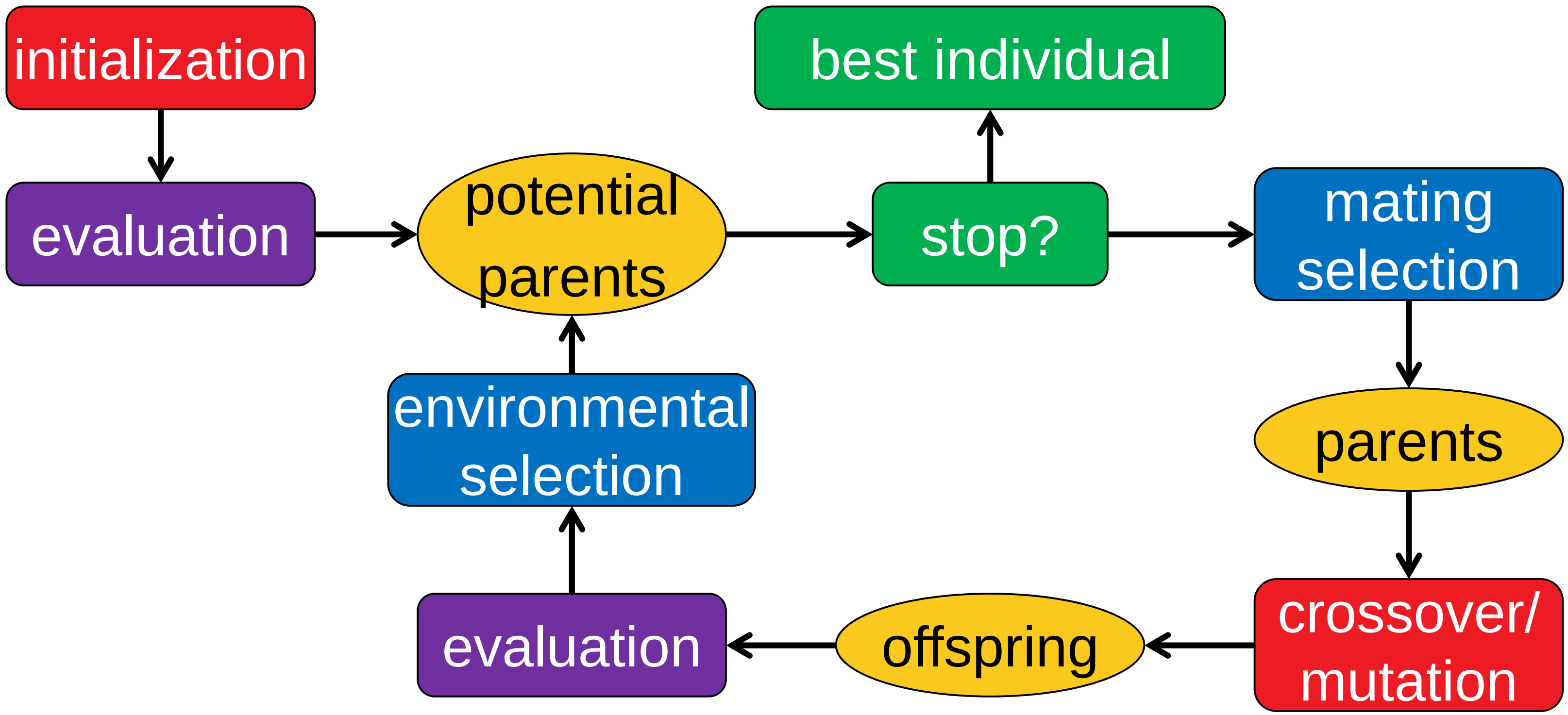
- Sample distribution $P(\boldsymbol{x}|\theta) \to \boldsymbol{x}_1, \dots, \boldsymbol{x}_\lambda \in \mathbb{R}^n$
- Evaluate $\boldsymbol{x}_1, \dots, \boldsymbol{x}_\lambda$ on f
- Update parameters $\theta \leftarrow F_\theta(\theta, \boldsymbol{x}_1, \dots, \boldsymbol{x}_\lambda, f(\boldsymbol{x}_1),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

- 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

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

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

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

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

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

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

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 **`http://www.cmap.polytechnique.fr/~nikolaus.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://www.cmap.polytechnique.fr/~dimo.brockhoff/optimizationSaclay/2015/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.

- There is also a tutorial available on Youtube by Y. Akimoto and N. Hansen: https://www.youtube.com/watch?v=7VBKLH3oDuw

Back to CMA-ES

The CMA-ES

Input: $\boldsymbol{m} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ
Initialize: $\mathbf{C} = \mathbf{I}$, and $\boldsymbol{p}_\mathrm{c} = \mathbf{0}$, $\boldsymbol{p}_\sigma = \mathbf{0}$,
Set: $c_\mathrm{c} \approx 4/n$, $c_\sigma \approx 4/n$, $c_1 \approx 2/n^2$, $c_\mu \approx \mu_w/n^2$, $c_1 + c_\mu \leq 1$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}$,
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

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

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

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

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

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

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

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 $f: \mathbb{R}^n \to \mathbb{R}$

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

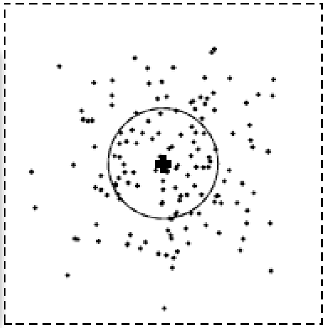
Sampling New Candidate Solutions (Offspring)

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

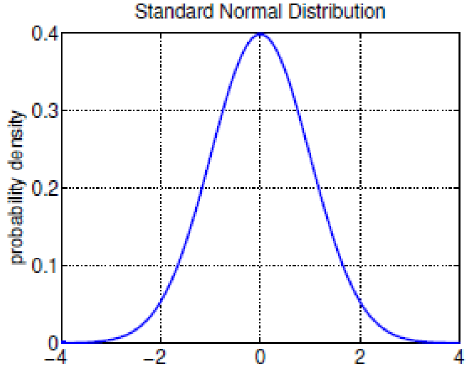
here, all new points are sampled with the same parameters

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 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

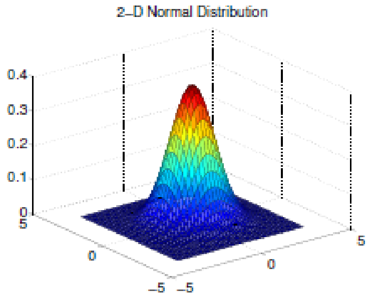
Any multi-variate normal distribution $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$ is uniquely determined by its mean value $\boldsymbol{m} \in \mathbb{R}^n$ and its symmetric positive definite $n \times n$ covariance matrix $\mathbf{C}$.

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

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



from [Auger, p. 13]

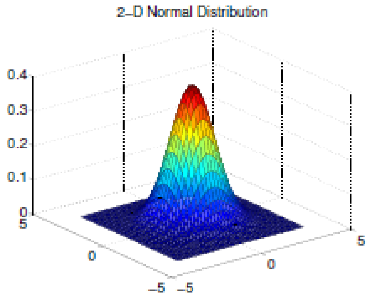
The Multi-Variate (n-Dimensional) Normal Distribution

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 $\mathbf{C}$

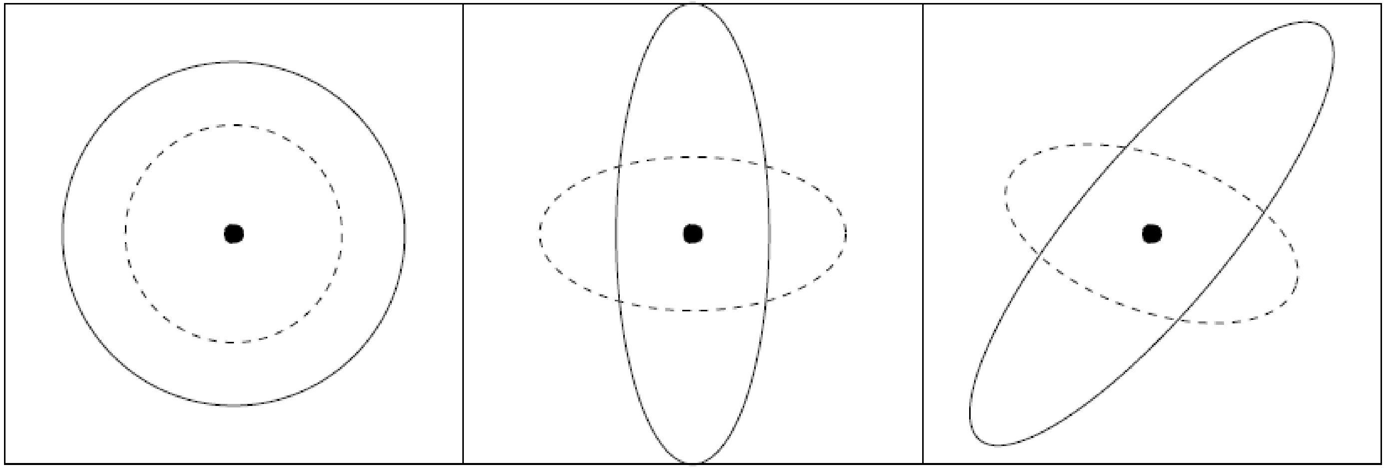
- determines the shape
- geometrical interpretation: any covariance matrix can be uniquely identified with the iso-density ellipsoid $\{\boldsymbol{x} \in \mathbb{R}^n \mid (\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}\mathbf{C}^{-1}(\boldsymbol{x}-\boldsymbol{m}) = 1\}$

from [Auger, p. 13]

Covariance Matrix: Lines of Equal Density

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

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

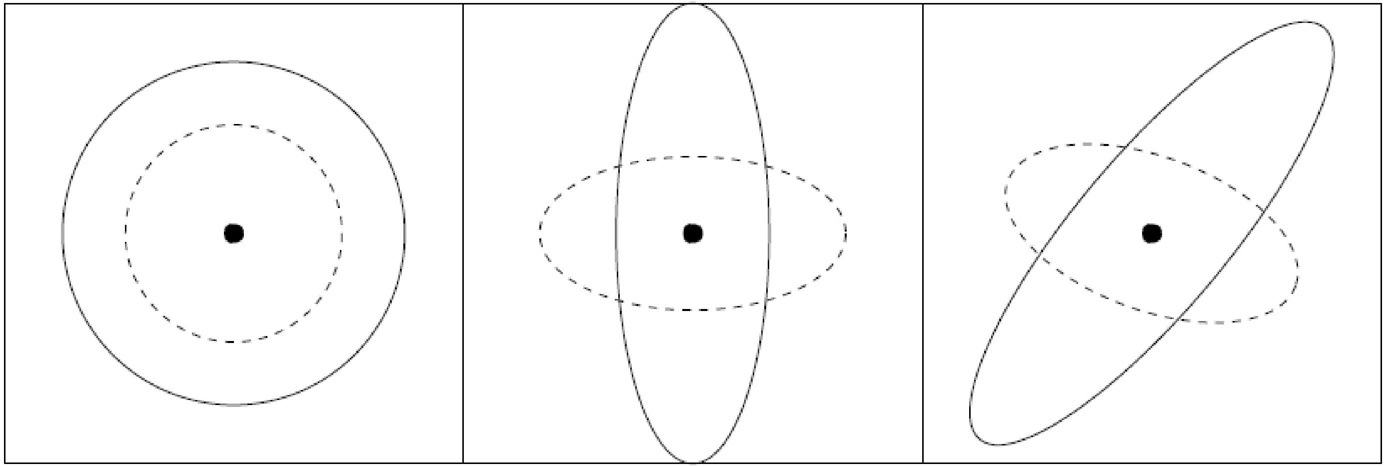
components are independent standard normally distributed

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]

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

Lines of Equal Density



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

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

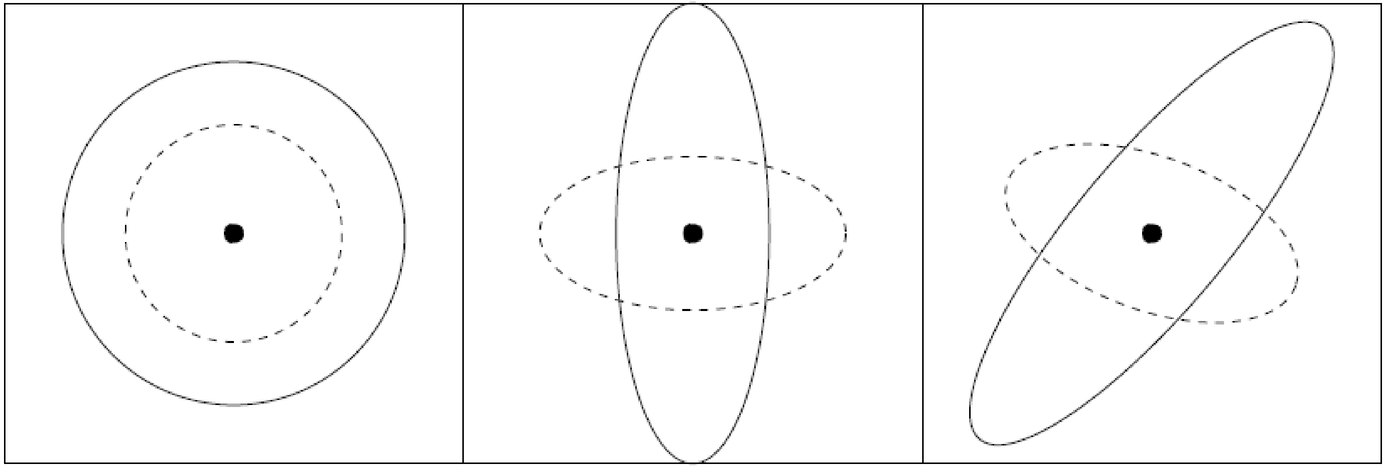
where $\mathbf{I}$ is the identity matrix (isotropic case) and $\mathbf{D}$ is a diagonal matrix (reasonable for separable problems) and $\mathbf{A} \times \mathcal{N}(\mathbf{0}, \mathbf{I}) \sim \mathcal{N}(\mathbf{0}, \mathbf{AA}^{\mathrm{T}})$ holds for all $\mathbf{A}$.

from [Auger, p. 14]

Covariance Matrix: Lines of Equal Density

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

Lines of Equal Density



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

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

$\mathcal{N}(m, \mathbf{C}) \sim m + \mathbf{C}^{\frac{1}{2}} \mathcal{N}(\mathbf{0}, \mathbf{I})$
$(n^2 + n)/2$ 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 $\mathbf{A}$.

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 {parents} $\cup$ {offspring}

(μ, λ)-ES: selection in {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]

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

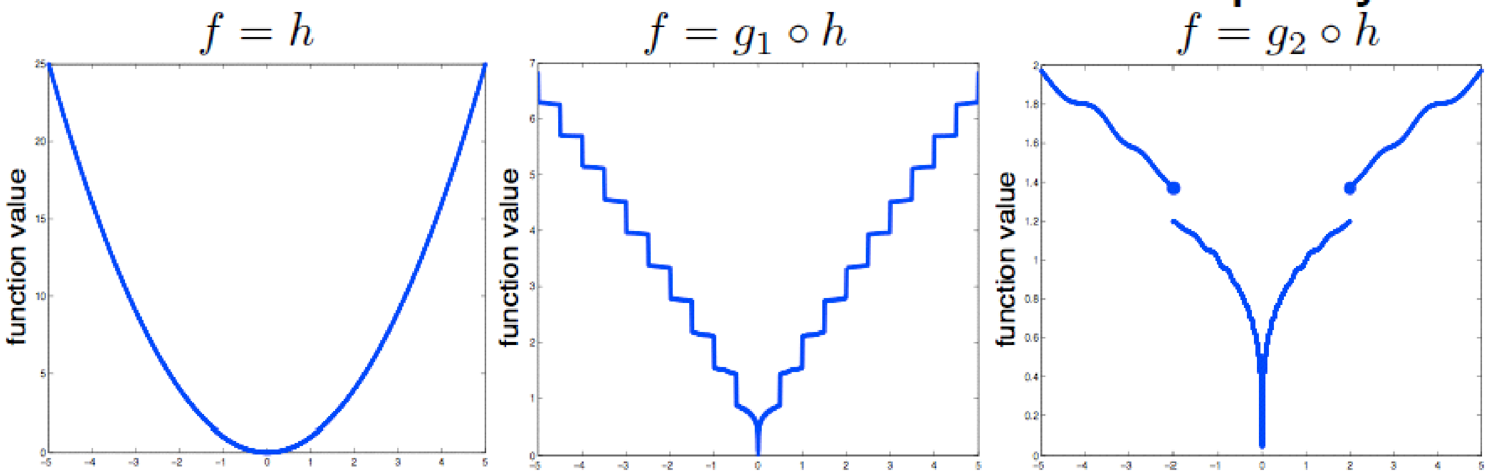
$$w_1 \geq \cdots \geq w_\mu > 0, \quad \sum_{i=1}^{\mu} w_i = 1, \quad \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} =: \mu_w \approx \frac{\lambda}{4}$$

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

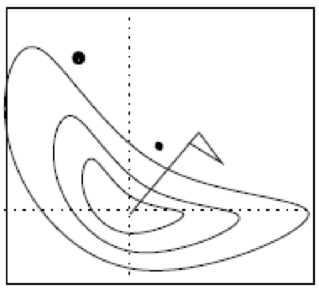
from [Hansen, p. 37]

Invariance Against Translations in Search Space

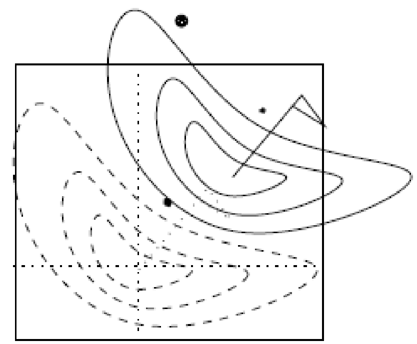
Basic Invariance in Search Space

- translation invariance

is true for most optimization algorithms



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



Identical behavior on f and f_a

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

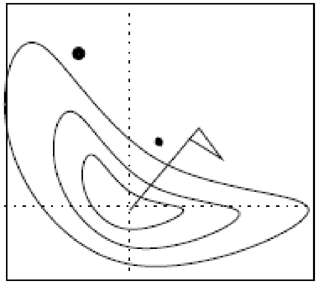
No difference can be observed w.r.t. the argument of f

from [Hansen, p. 38]

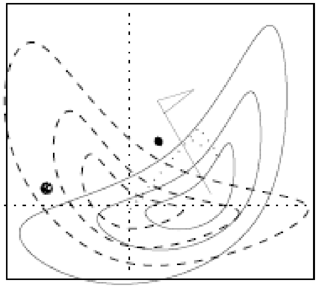
Rotational Invariance in Search Space

- 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

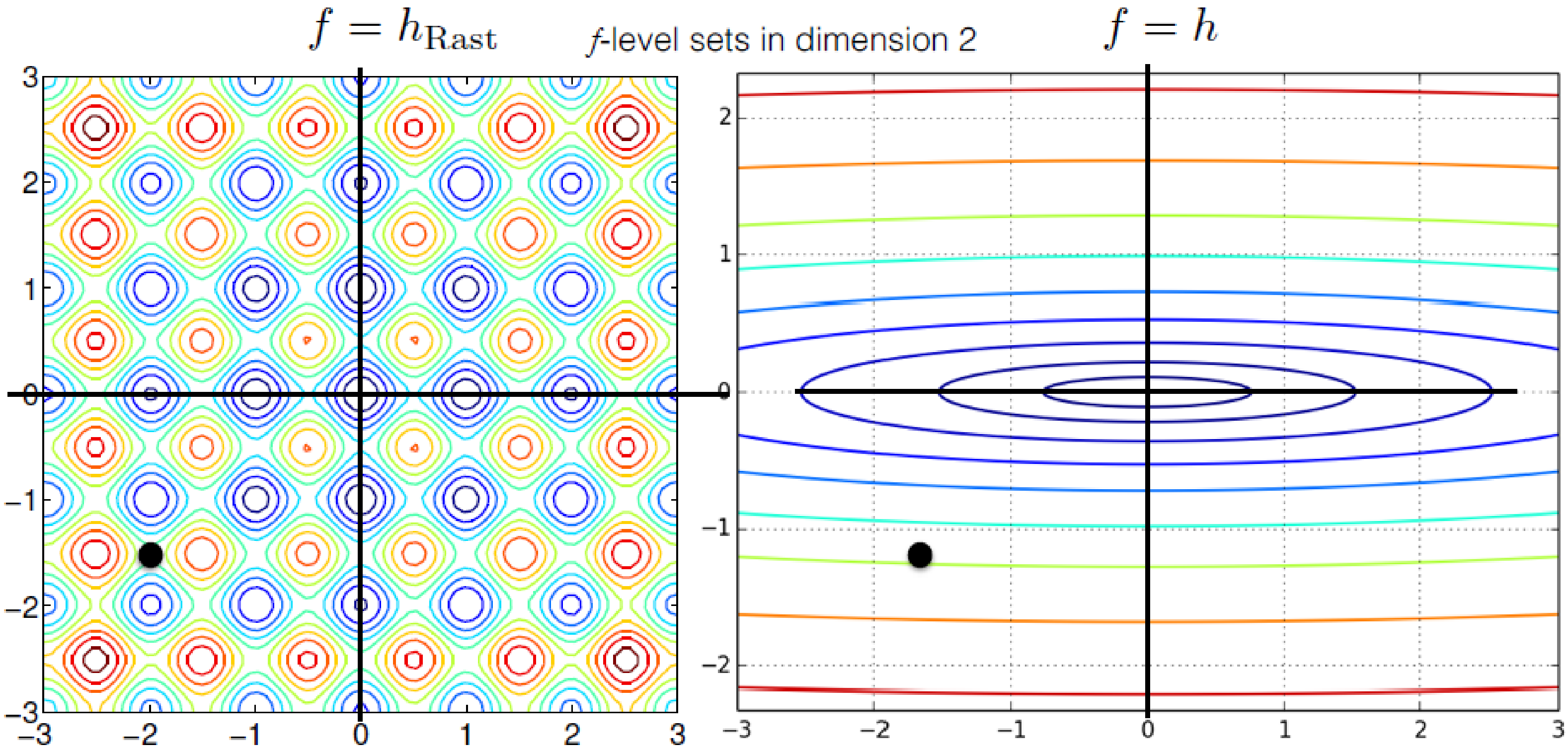
No difference can be observed w.r.t. the argument of f

[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

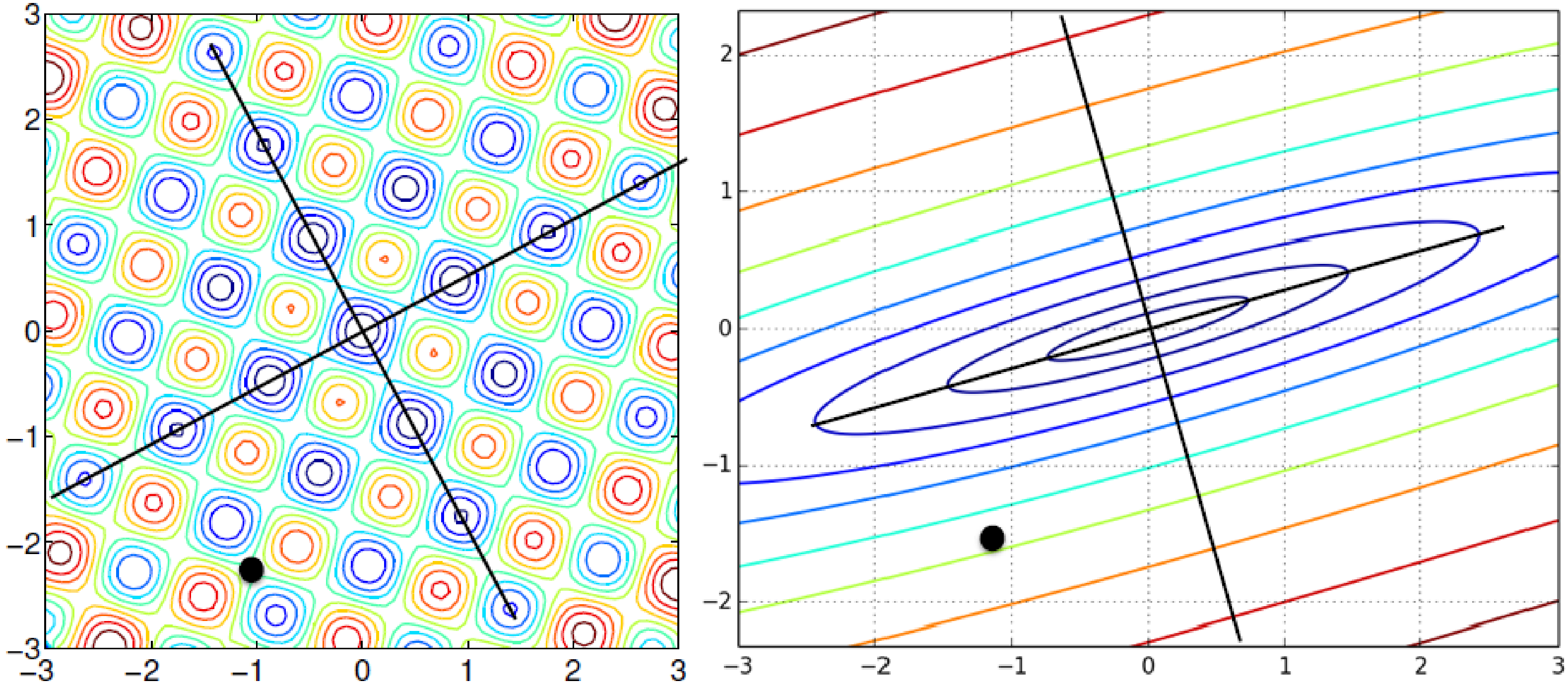


for example, invariance under search space rotation
(separable $\Leftrightarrow$ non-separable)

from [Hansen, p. 40]

Invariance Under Rigid Search Space Transformations

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

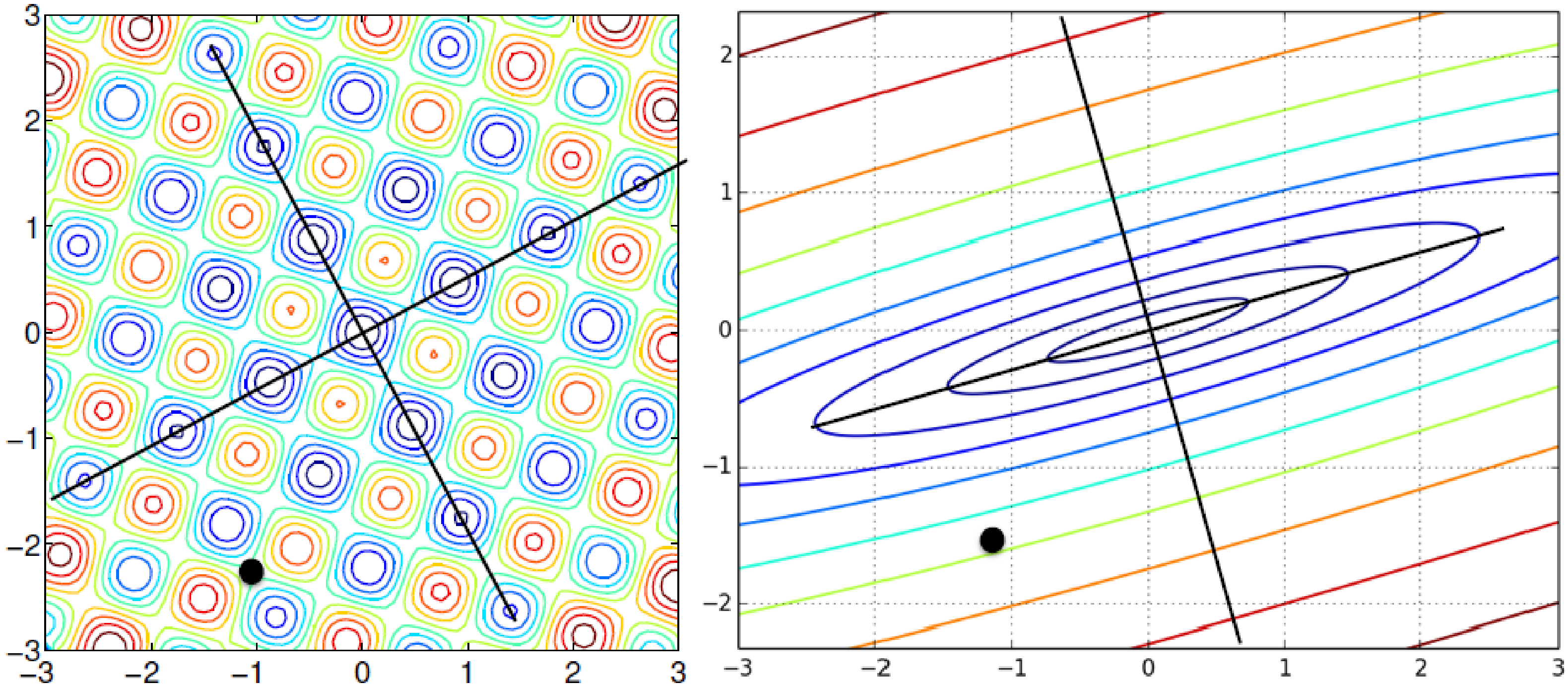


for example, invariance under search space rotation
(separable $\Leftrightarrow$ non-separable)

from [Hansen, p. 41]

Invariance Under Rigid Search Space Transformations

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



for example, invariance 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

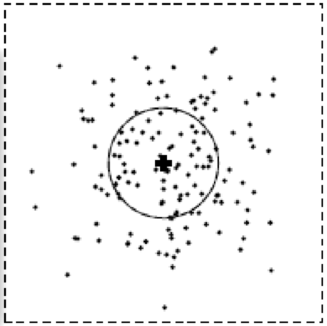
Evolution Strategies

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



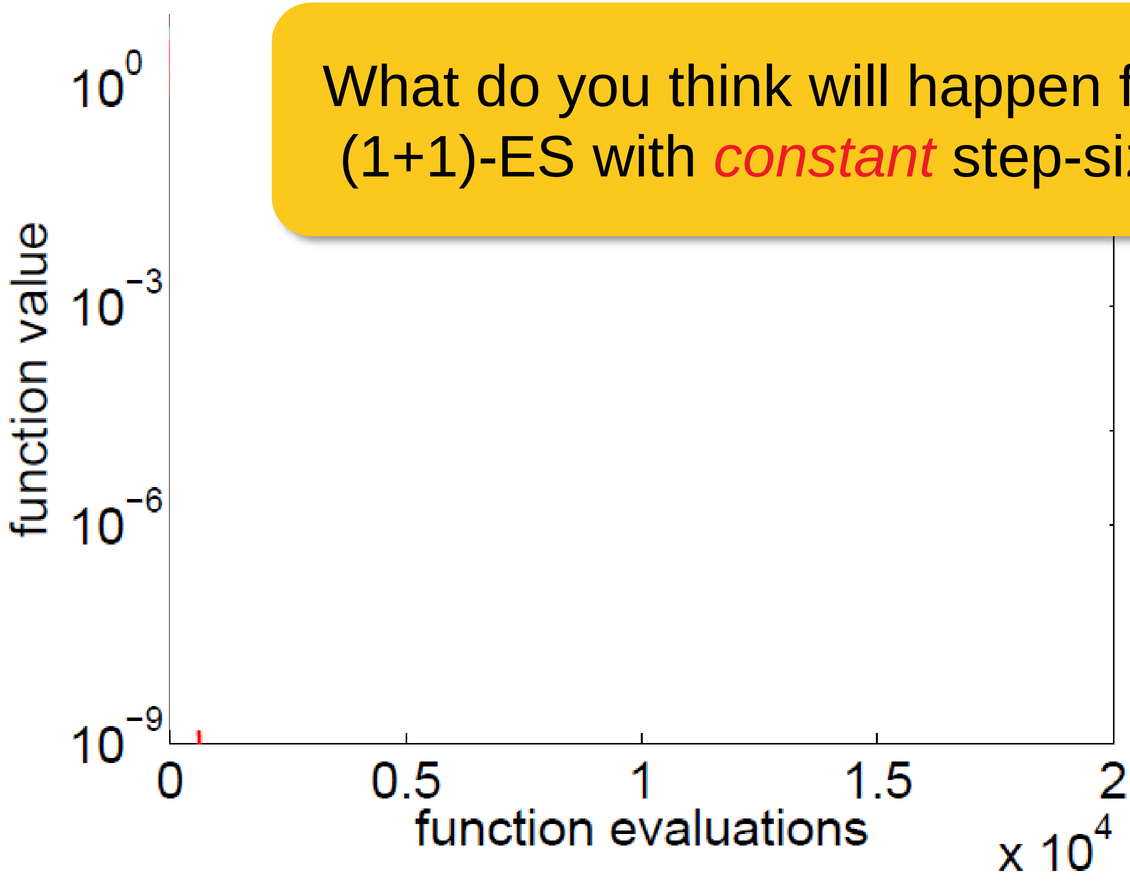
where

- 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?



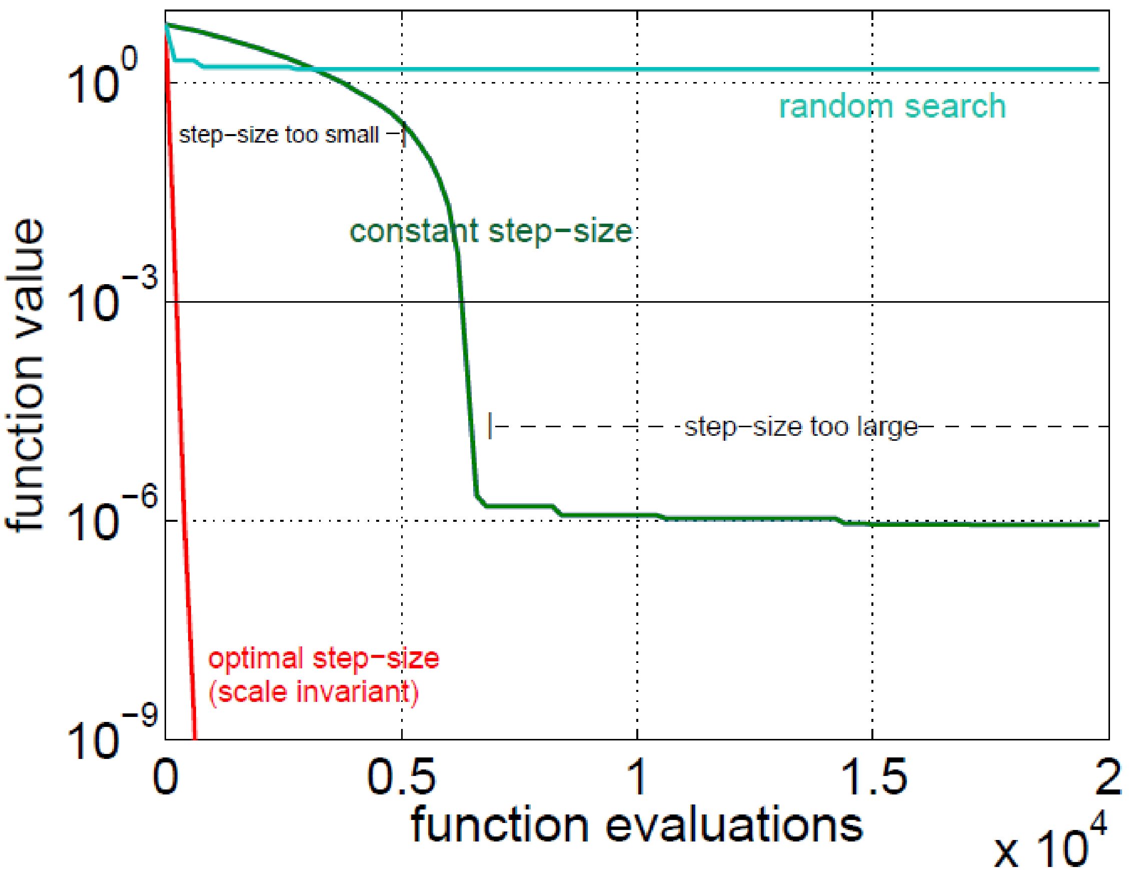
What do you think will happen for a (1+1)-ES with *constant* step-size?

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

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

from [Auger, p. 22]

Why Step-Size Control?

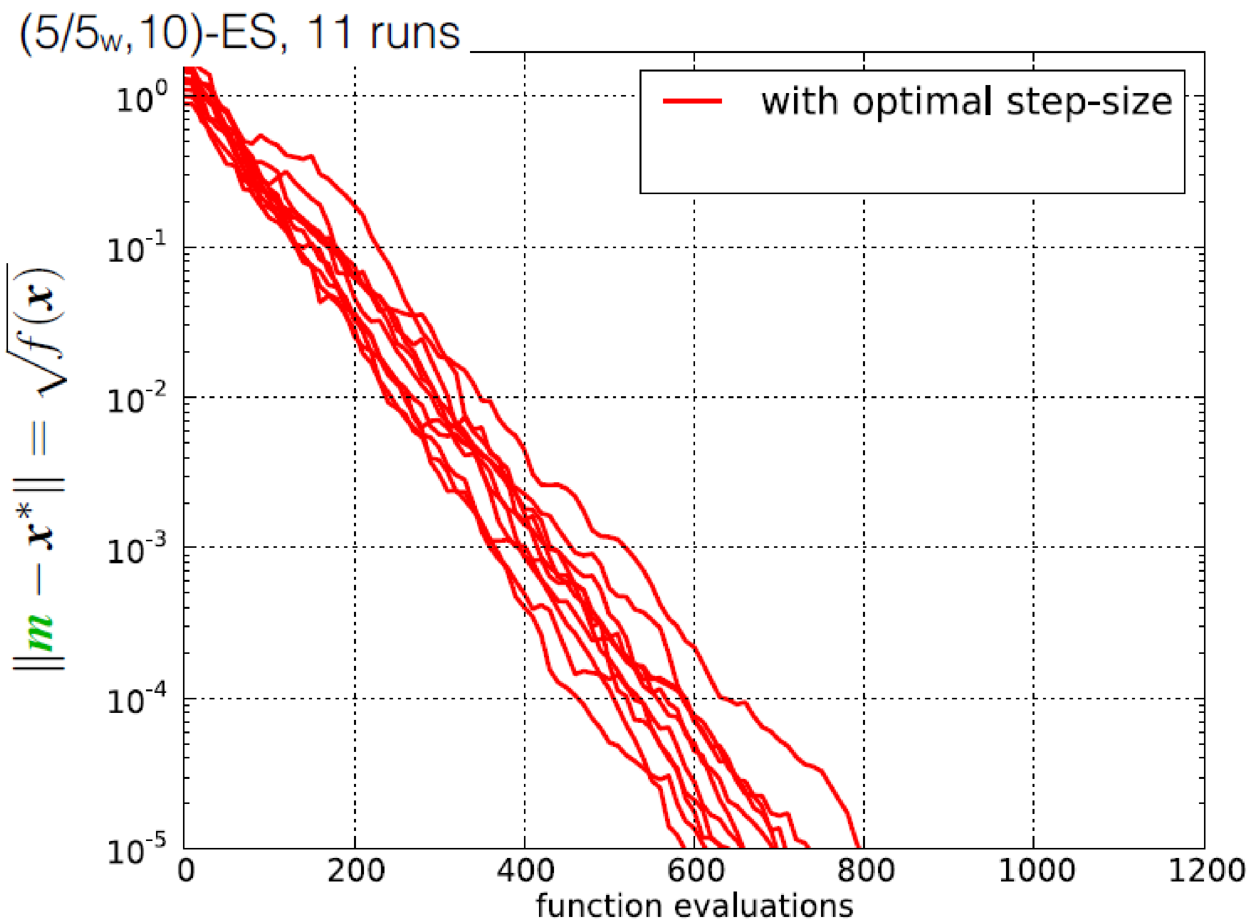


$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$$

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

from [Auger, p. 22]

Why Step-Size Control?



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

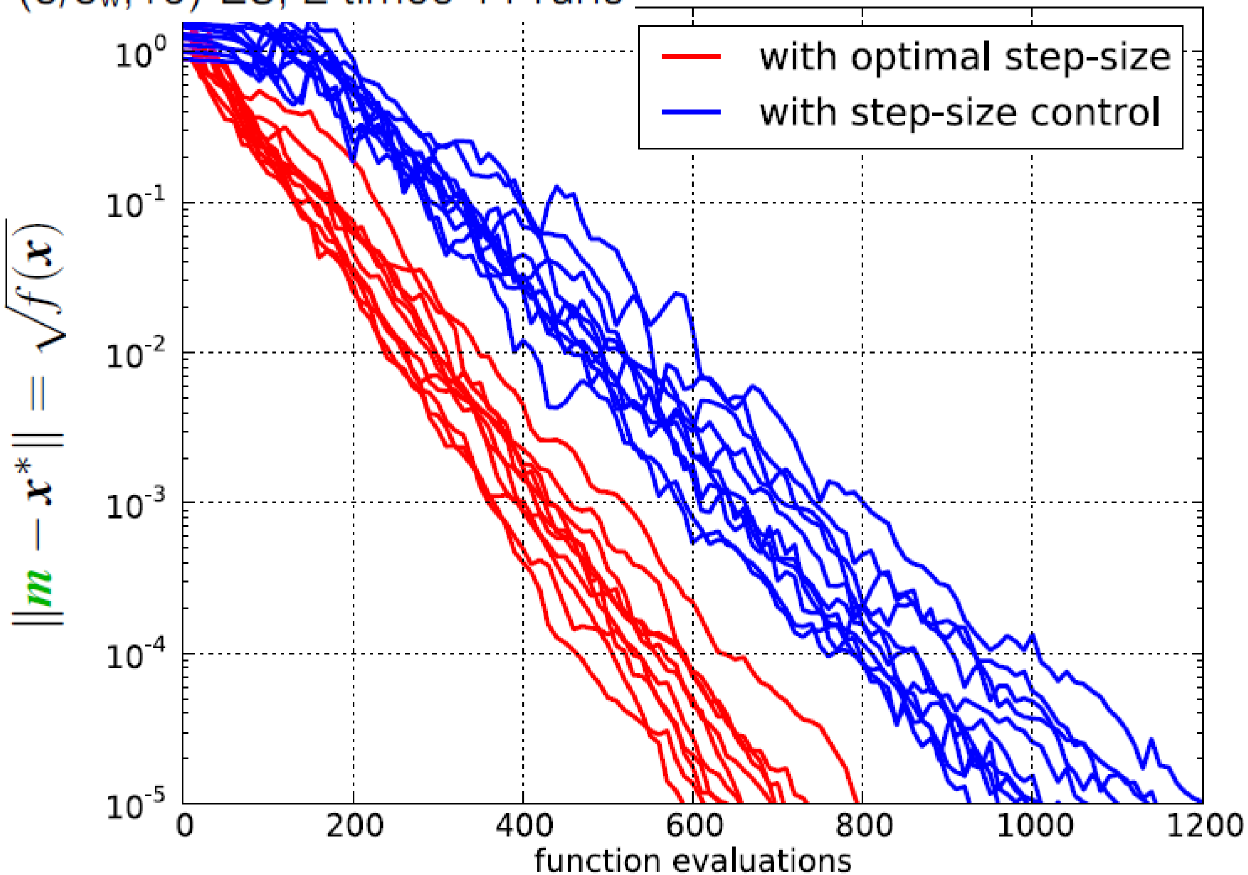
for $n = 10$ and
$\boldsymbol{x}^0 \in [-0.2, 0.8]^n$

with optimal step-size σ

from [Hansen, p. 47]

Why Step-Size Control?

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



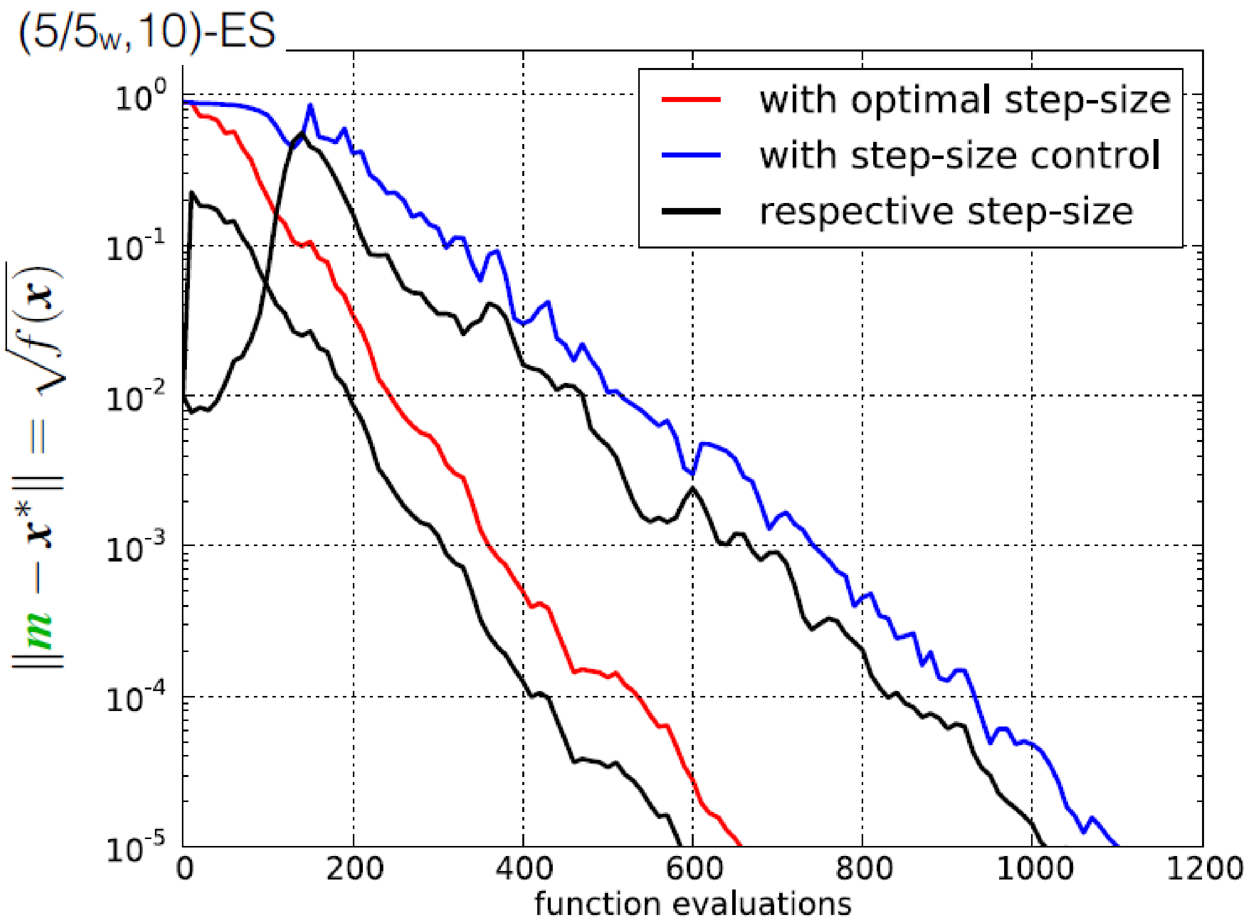
$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$$

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?



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

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

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

from [Hansen, p. 49]

Adapting the Step-Size

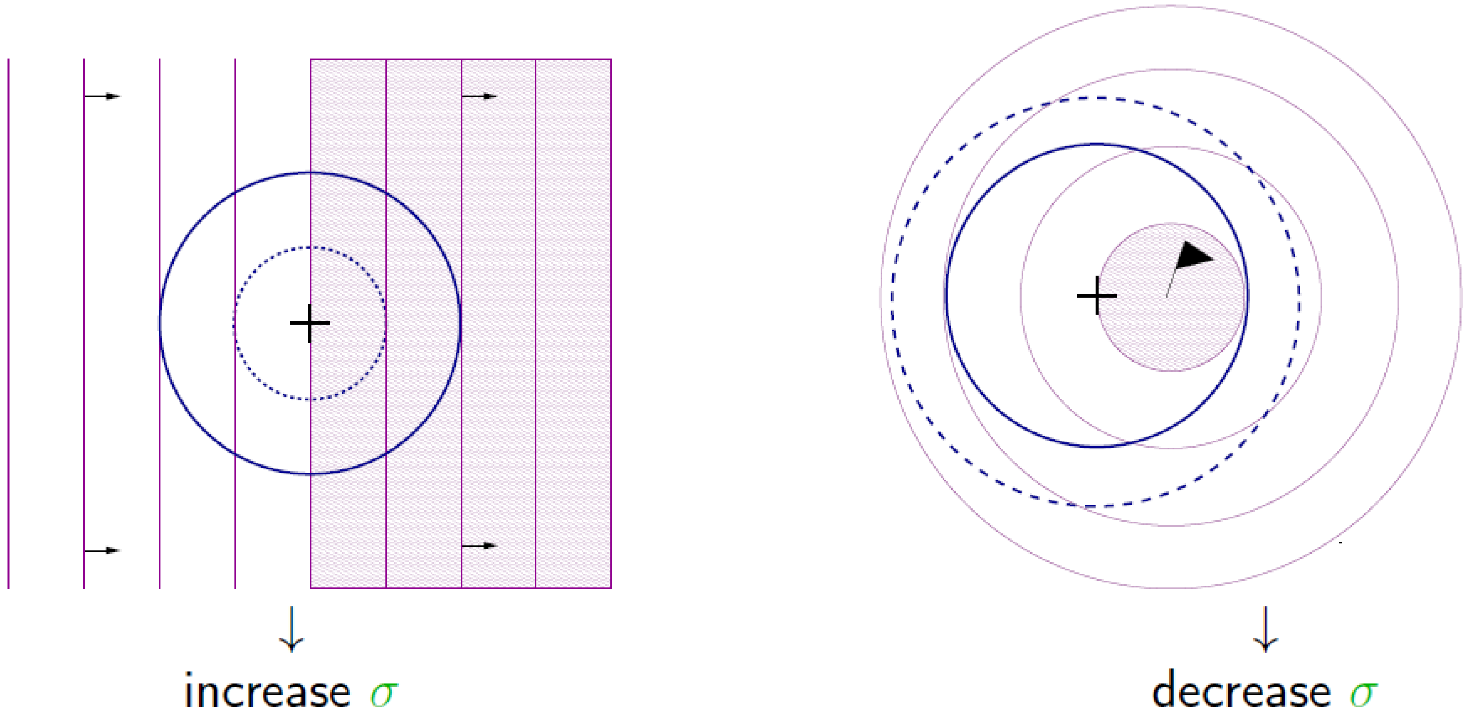
Question:

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

Most common:

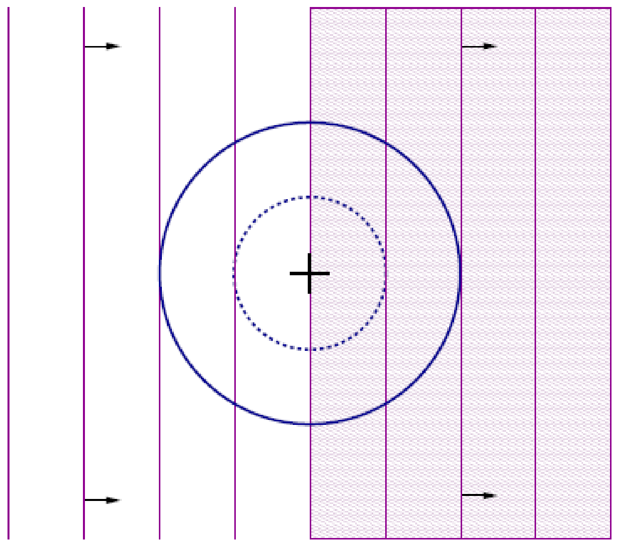
- 1/5 success rule
- Cumulative Step-Size Adaptation (CSA, as in standard CMA-ES)
- others possible (Two-Point Adaptation, self-adaptive step-size, ...)

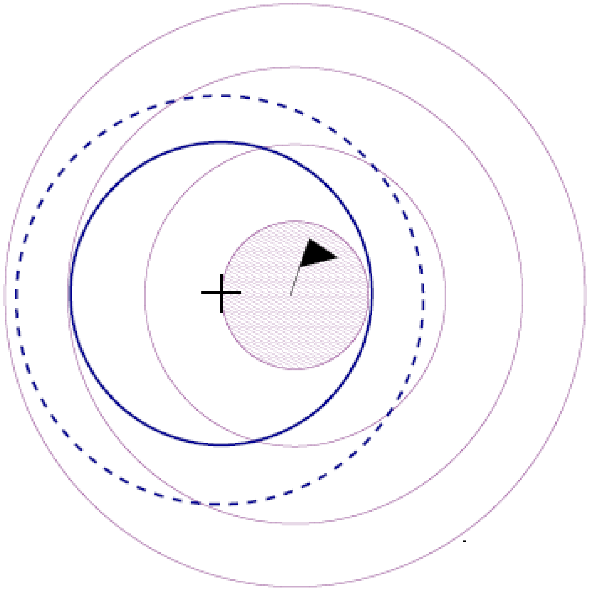
One-fifth success rule



from [Auger, p. 32]

One-fifth success rule





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

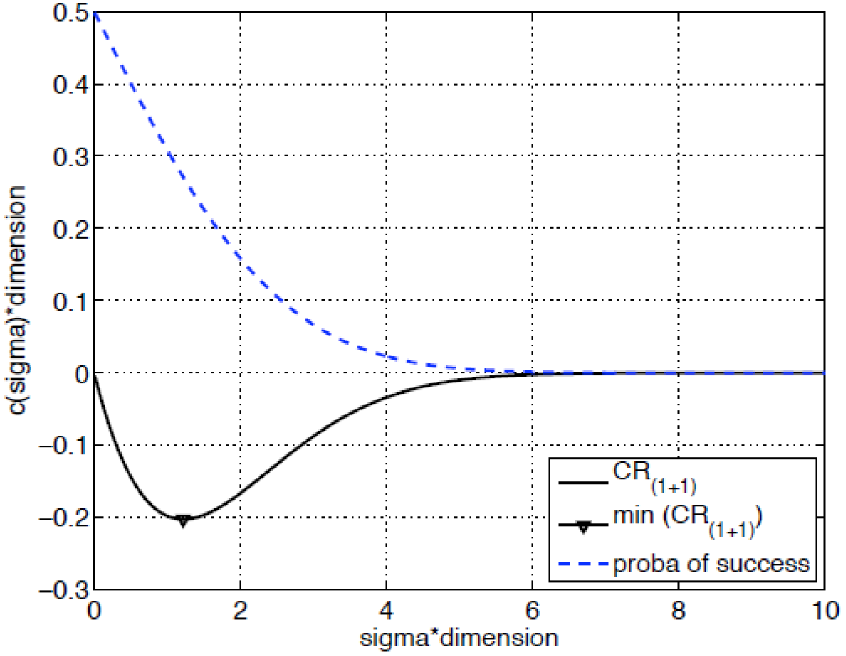
$p_s = 0, \sigma \leftarrow \sigma / \exp(1/3)^{1/4}$

from [Auger, p. 34]

One-Fifth Success Rule

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]