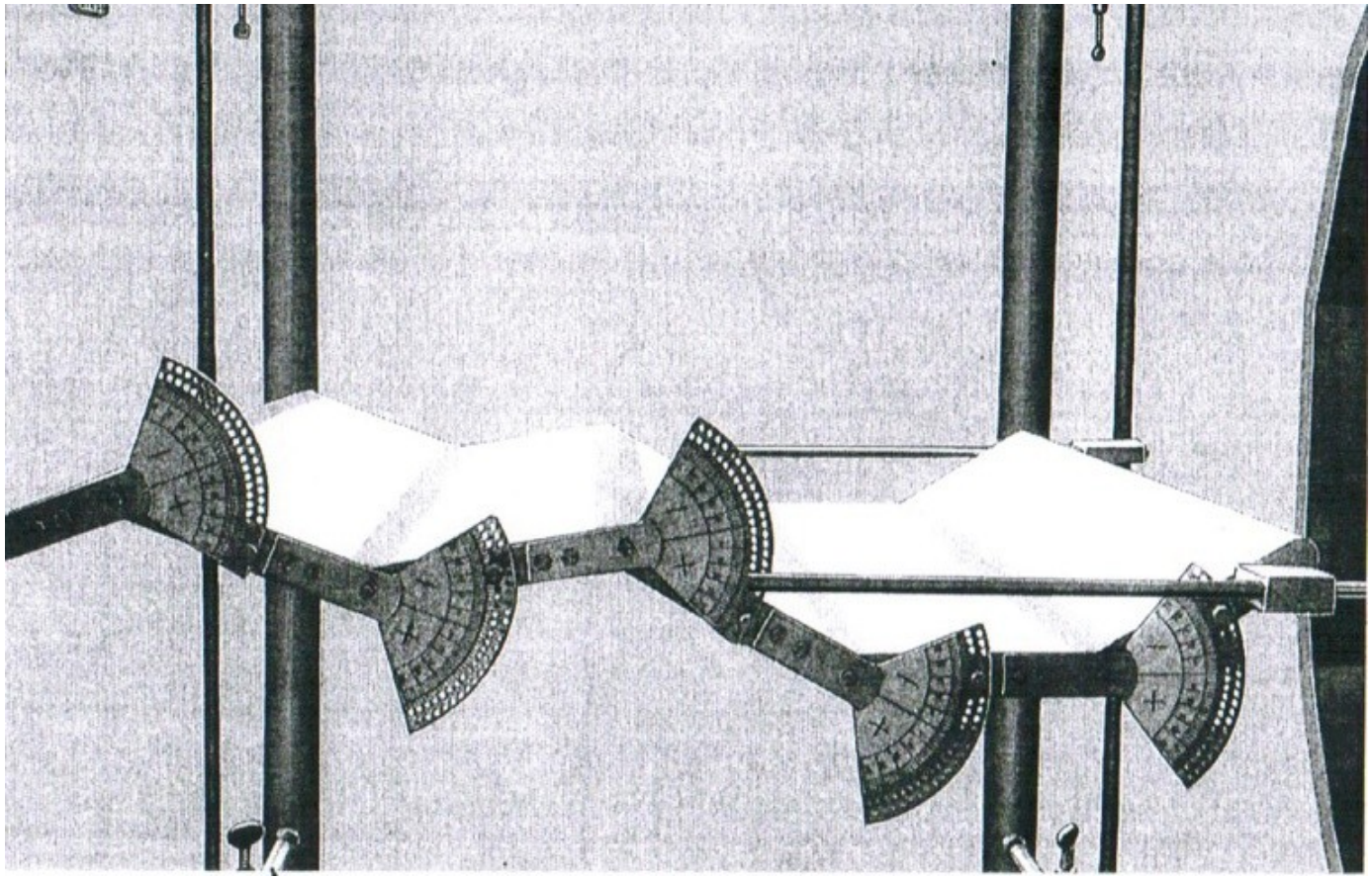
Evolution Strategies

Overview

- The early experiments
- Basic introduction
  - Representation, strategy parameters, mutation, recombination, selection
- *Self-adaptation of strategy parameters*
- *The (1+1)-evolution strategy*
- *Convergence velocity theory*
- *Application examples*

THE EARLY EXPERIMENTS

Early Experiments I: Flow Plate



- A plate with 5 controllable angle brackets
- Measurable air flow drag (by a pitot tube)

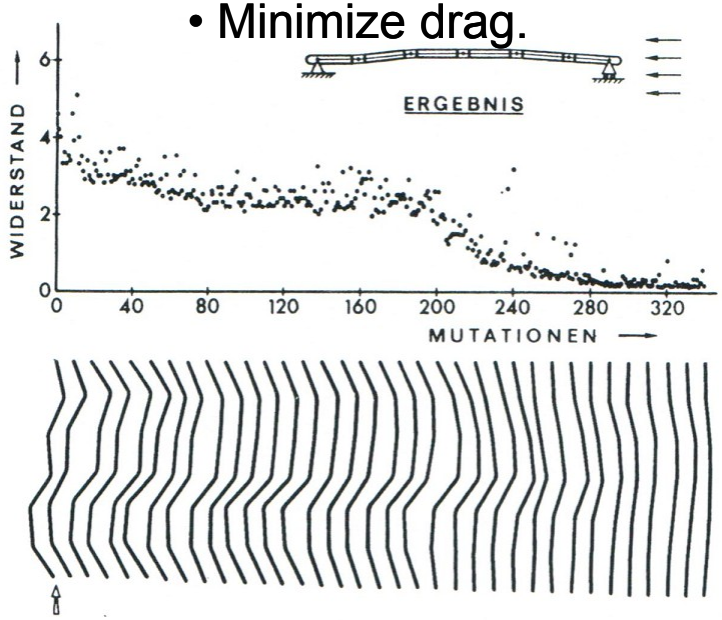
Figure from: I. Rechenberg, Evolutionsstrategie ´73, frommann-holzboog, Stuttgart 1973

Early Experiments I: Flow Plate

Experiment 1:
- Left / right supporting point at same y-coordinate.
- Horizontal flow.
- Minimize drag.

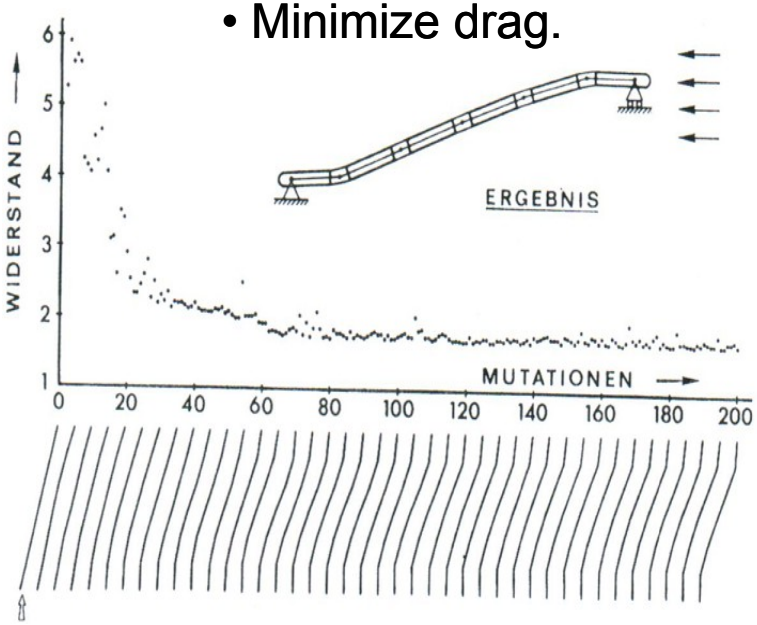
Experiment 2:
- Left supporting point 25% lower than right one.
- Horizontal flow.
- Minimize drag.

Plate drag



Number of mutations and selected plate shapes

Start	-30	-40	40	-30	40
End	0	4	0	6	-6

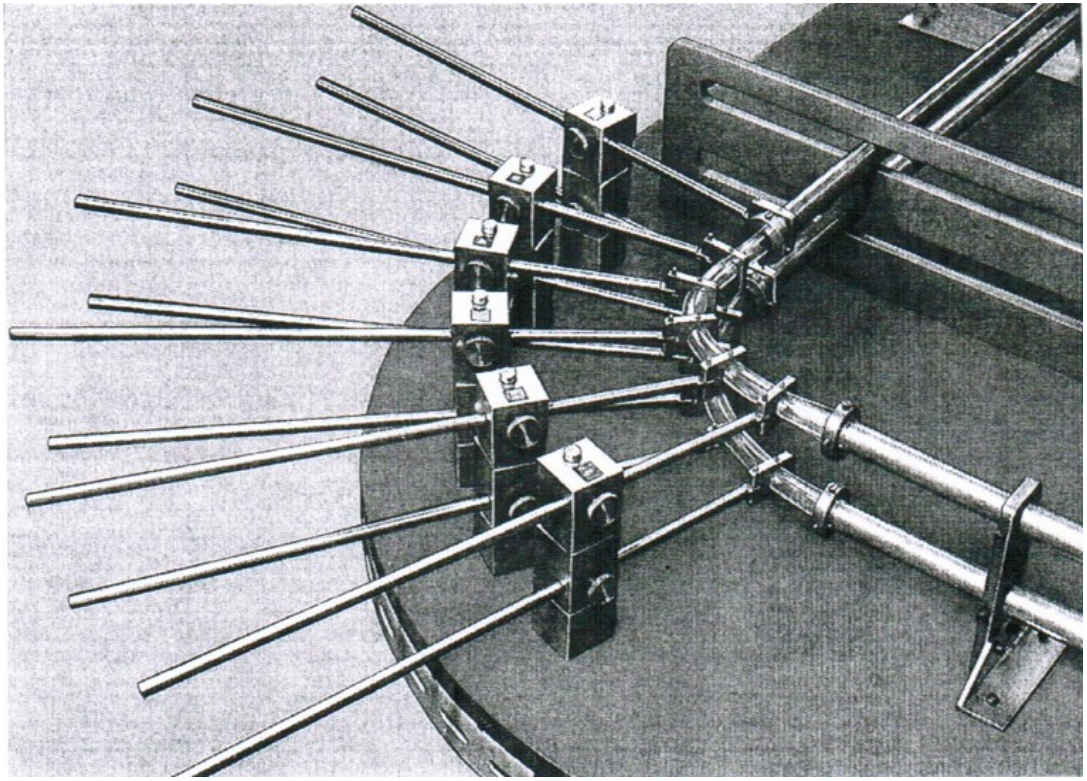


Number of mutations and selected plate shapes

Start	0	0	0	0	0
End	16	6	2	0	-18

Figures from: I. Rechenberg, Evolutionsstrategie ´73, frommann-holzboog, Stuttgart 1973

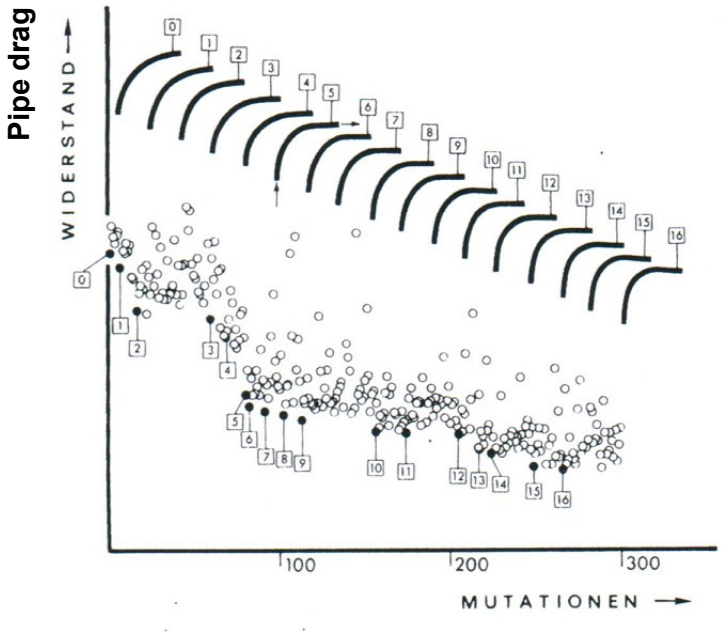
Early Experiments II: Bended Pipe



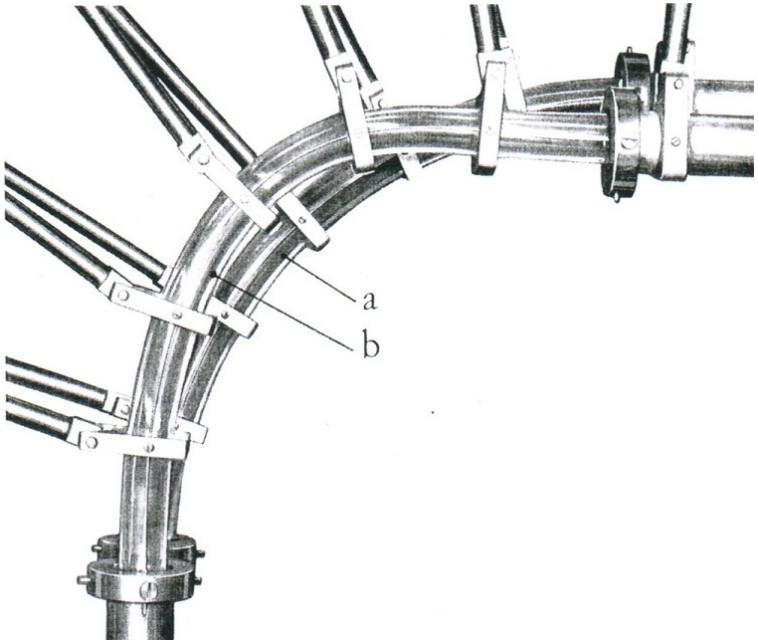
- A flexible pipe with 6 controllable bending devices
  - Minimize bend losses of liquid flow
    - Measure drag by pitot tube

Figure from: I. Rechenberg, Evolutionsstrategie ´73, frommann-holzboog, Stuttgart 1973

Early Experiments II: Bended Pipe



Number of mutations and selected pipe shapes



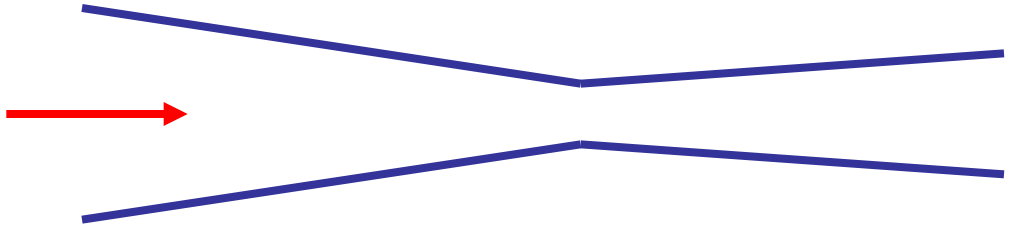
Initial (a) and optimized (b) pipe shape

- Bend loss of final form reduced by 10%
- Including drag a total reduction of 2%

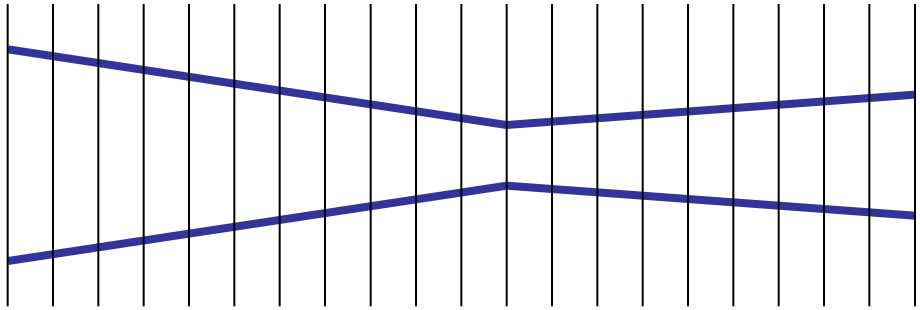
Figure from: I. Rechenberg, Evolutionsstrategie ´73, frommann-holzboog, Stuttgart 1973

Early Experiment III: Nozzle

- What can be done if physics, (bio-) chemistry, … of process unkown?
- No model or simulation program available!
- Idea: Optimize with the real object
- “Hardware in the loop”
- Example: Supersonic nozzle, turbulent flow, physical model not available.



Experimental Setup: Nozzle

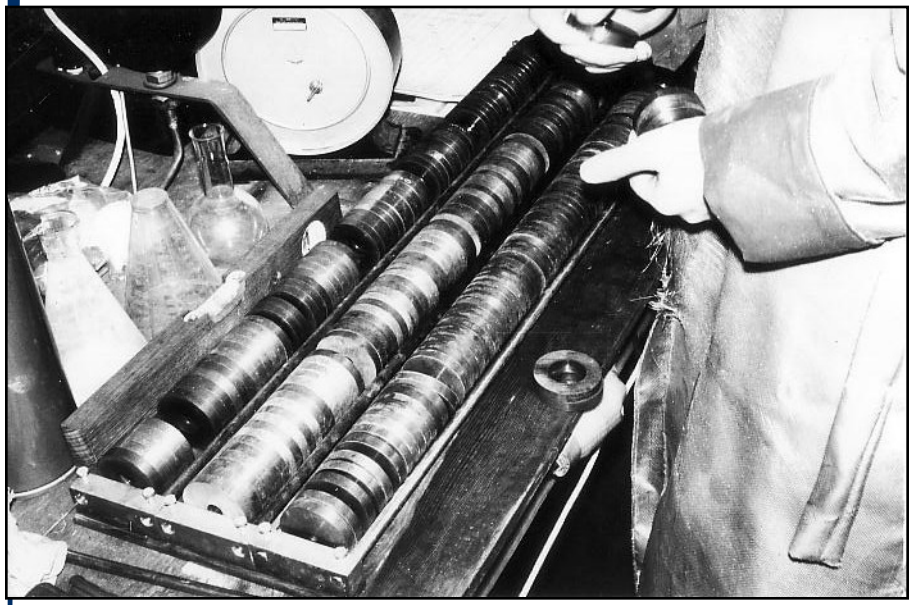


- Production of differently formed conic nozzle parts (pierced plates).
- Form of nozzle part is value of decision variable.

choosing conic nozzle parts (by EA)
clamping of conic nozzle parts (manually)
steam under high pressure passed into nozzle
degree of efficiency is **measured!**

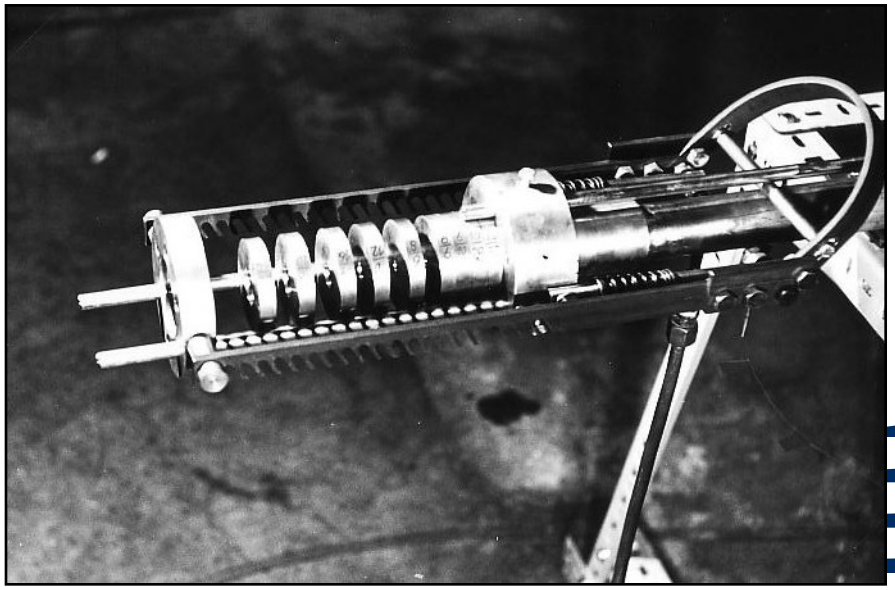
„simulator replacement“

Nozzle Experiment (I)



collection of conical nozzle parts

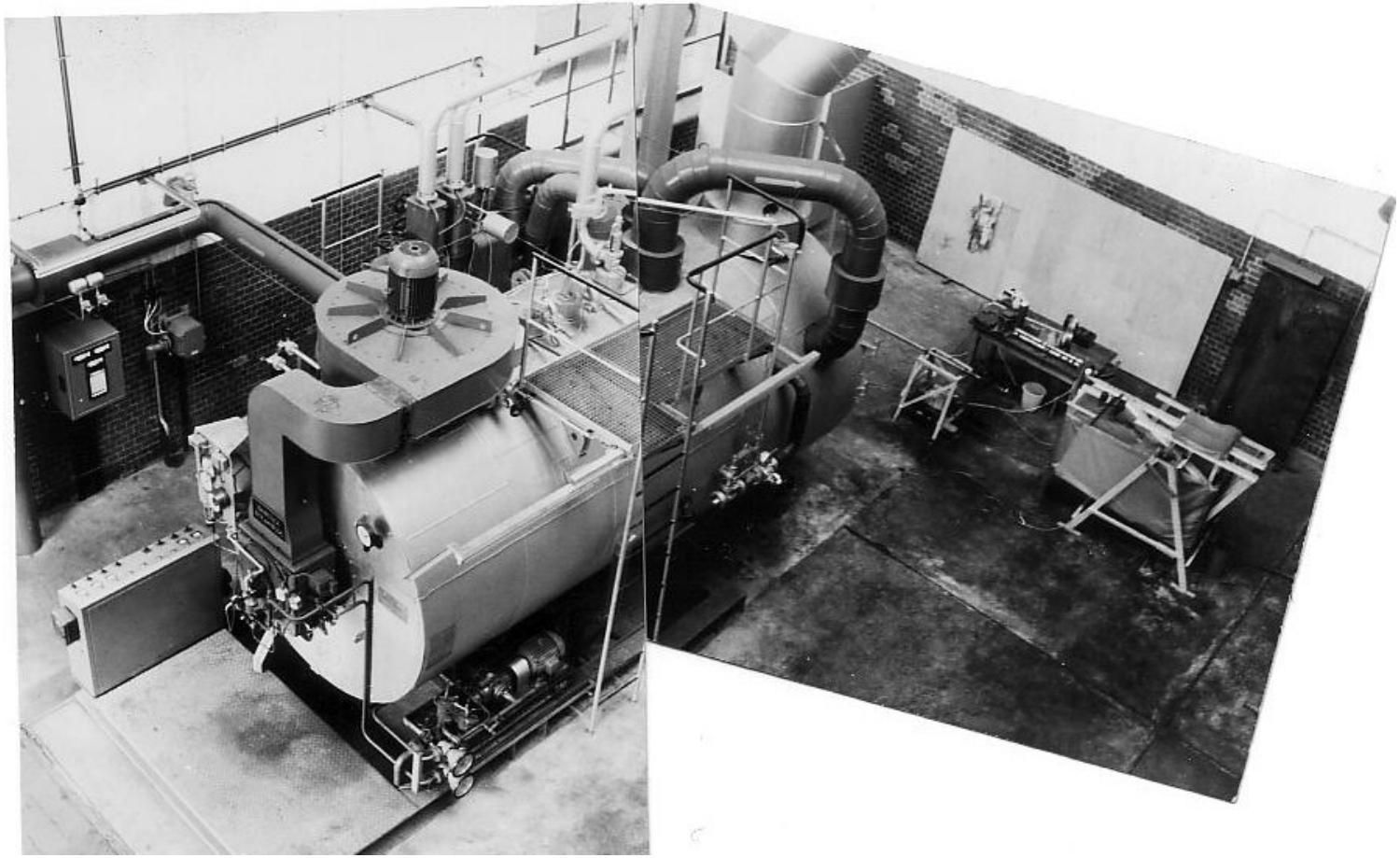
device for clamping nozzle parts



Nozzle Experiment (II)

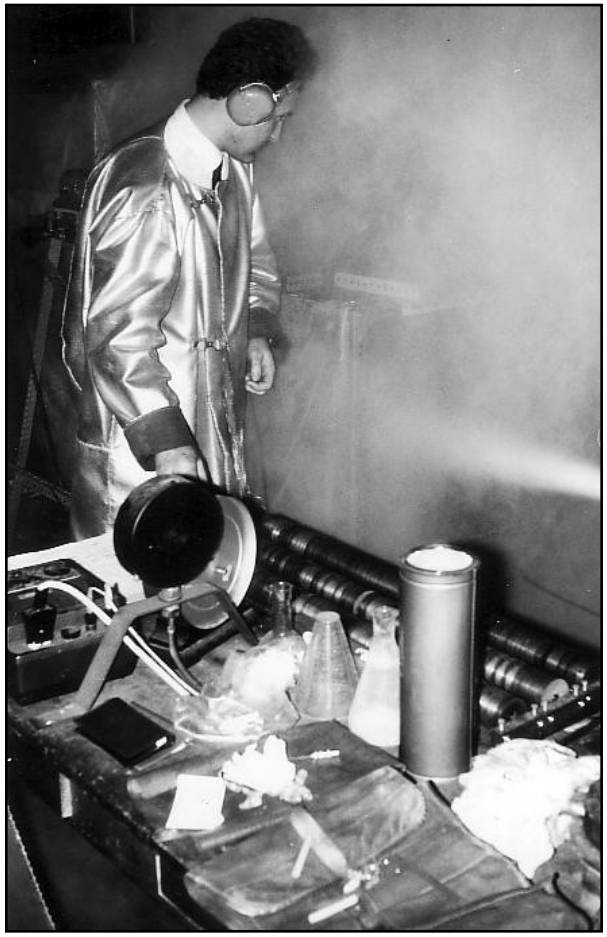
Hans-Paul Schwefel
while changing nozzle parts

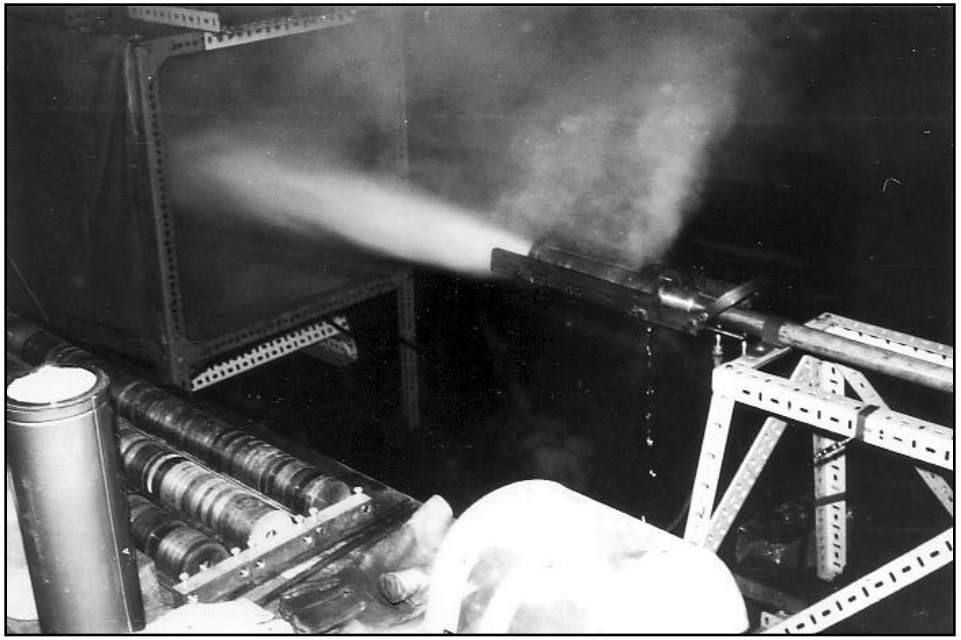
Nozzle Experiment (III)



steam plant / experimental setup

Nozzle Experiment (IV)





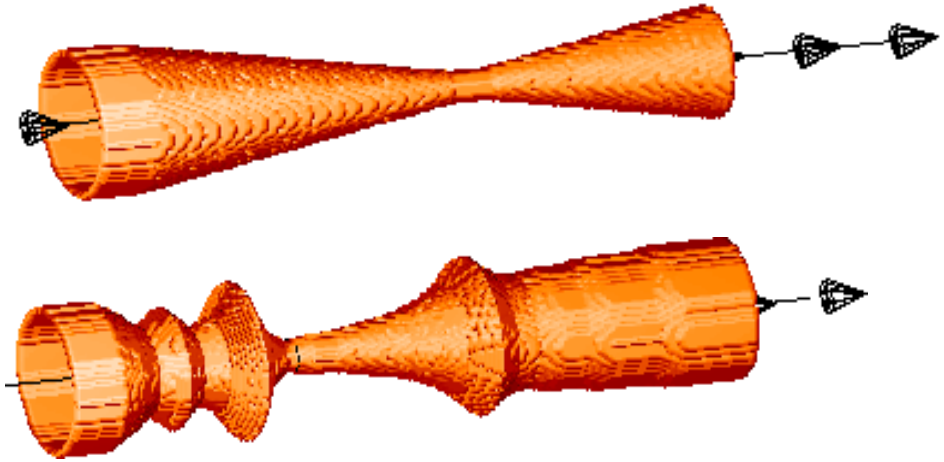
the nozzle in operation …

… while measuring degree of efficiency

Nozzle Results (I)

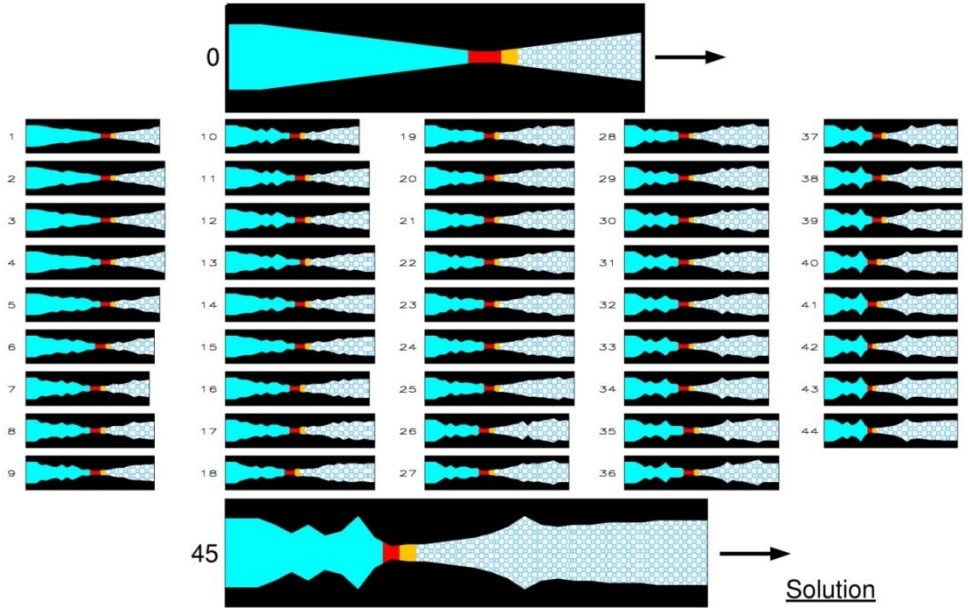
- Illustrative Example: Optimize Efficiency
  - Initial:

  - Evolution:



- 32% Improvement in Efficiency !

Nozzle Results (II)



- 250 experiments were made.
- 45 improvements found.
- Discrete ring segments, variable-dimensional optimisation
- Gene duplication and deletion as additional operators.

J. Klockgether and H.-P. Schwefel, “Two-phase nozzle and hollow core jet experiments,” in Proceedings of the 11th Symposium on Engineering Aspects of Magneto-Hydrodynamics, Caltech, Pasadena, California, USA, 1970.

BASIC INTRODUCTION

Evolution Strategy – Basics

- Mostly real-valued search space IR^n
  - also mixed-integer, discrete spaces
- Emphasis on mutation
  - n-dimensional normal distribution
  - expectation zero
- Different recombination operators
- Deterministic selection
  - (μ, λ)-selection: Deterioration possible
  - $(\mu+\lambda)$-selection: Only accepts improvements
- $\lambda >> \mu$, i.e.: Creation of offspring surplus
- Self-adaptation of strategy parameters.

Evolution Strategy:

Algorithms
Representation

Representation of search points

- Simple ES with 1/5 success rule:
  - Exogenous adaptation of step size σ
  - Mutation: N(0, σ)

$$\vec{a} = (x_1, \ldots, x_n)$$

- Self-adaptive ES with single step size:
  - One σ controls mutation for all x_i
  - Mutation: N(0, σ)

$$\vec{a} = ((x_1, \ldots, x_n), \sigma)$$

Representation of search points

- Self-adaptive ES with individual step sizes:
  - One individual σ_i per x_i
  - Mutation: $N_i(0, \sigma_i)$

$$\vec{a} = ((x_1,...,x_n),(\sigma_1,...,\sigma_n))$$

- Self-adaptive ES with correlated mutation:
  - Individual step sizes
  - One correlation angle per coordinate pair
  - Mutation according to covariance matrix: N(**0**, **C**)

$$\vec{a} = ((x_1,...,x_n),(\sigma_1,...,\sigma_n),(\alpha_1,...,\alpha_{n(n-1)/2}))$$

How many strategy parameters?

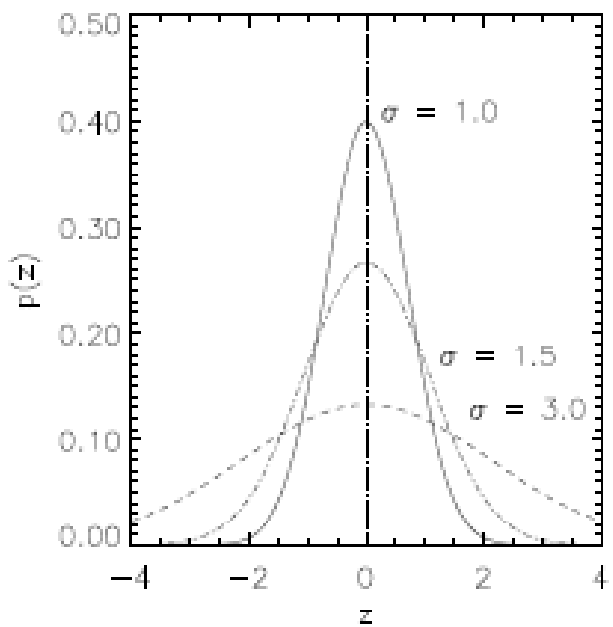
n_σ	n_α	**Remark**
1	0	Standard mutation, one global stepsize
n	0	Standard mutation, individual step sizes
n	$n(n-1)/2$	Correlated mutations
$1 \le n_\sigma \le n$	$\left(n - \frac{n_\sigma}{2}\right)(n_\sigma - 1)$	General case (correlated mutations)

Evolution Strategy:

Algorithms
Mutation

Normal distribution

- Mutation makes use of normally distributed variations
- Probability density function: $$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),$$
- Expectation typically zero (μ=0)
- Standard deviation σ needs to be adapted



Operators: Mutation – one σ

- Self-adaptive ES with one step size:
  - One σ controls mutation for all x_i
  - Mutation: N(0, σ)

$$\vec{a} = ((x_1, \ldots, x_n), \sigma)$$

$$\vec{a}' = ((x_1', \ldots, x_n'), \sigma')$$

$$\sigma' = \sigma \cdot \exp(\tau_0 \cdot N(0,1))$$

$$x_i' = x_i + \sigma' \cdot N_i(0,1)$$

Individual before mutation

Individual after mutation

1.: Mutation of step sizes

2.: Mutation of objective variables

Here the new σ‘ is used!

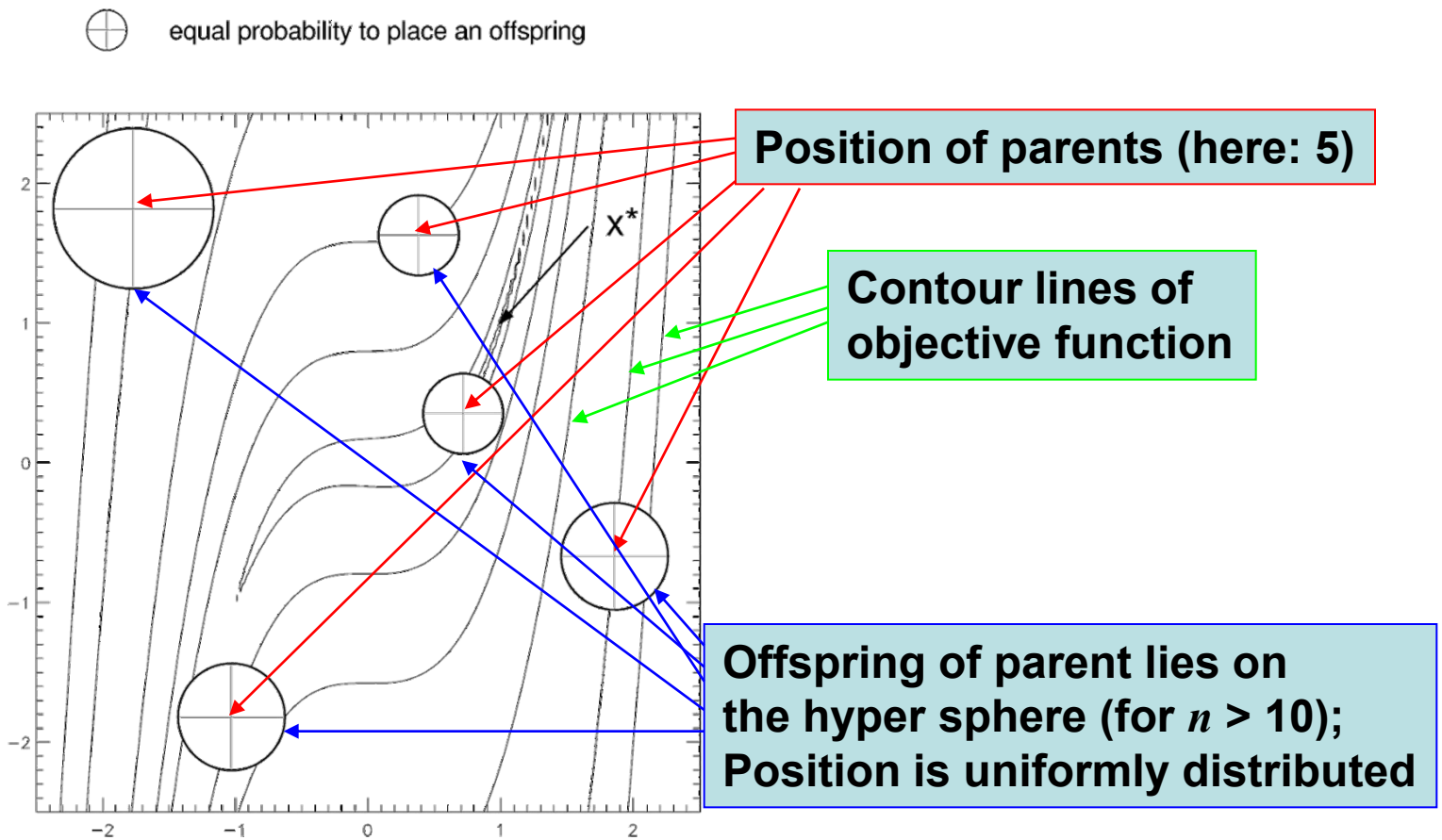
Operators: Mutation – one σ

- Thereby τ_0 is the so-called learning rate
  - Affects the speed of the σ-Adaptation
  - τ_0 bigger: faster but more imprecise
  - τ_0 smaller: slower but more precise
  - How to choose τ_0?
  - According to recommendation of Schwefel*:

$$\tau_0 = \frac{1}{\sqrt{n}}$$

*H.-P. Schwefel: Evolution and Optimum Seeking, Wiley, NY, 1995.

Operators: Mutation – one σ



Pros and Cons: One σ

- Advantages:
  - Simple adaptation mechanism
  - Self-adaptation usually fast and precise
- Disadvantages:
  - Bad adaptation in case of complicated contour lines
  - Bad adaptation in case of very differently scaled object variables
    - $-100 < x_i < 100$ and e.g. $-1 < x_j < 1$

Operators: Mutation – individual σ_i

- Self-adaptive ES with individual step sizes:
  - One σ_i per x_i
  - Mutation: $N_i(0, \sigma_i)$

$$\vec{a} = ((x_1,\ldots,x_n),(\sigma_1,\ldots,\sigma_n))$$

$$\vec{a}' = ((x_1',\ldots,x_n'),(\sigma_1',\ldots,\sigma_n'))$$

$$\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1))$$

$$x_i' = x_i + \sigma_i' \cdot N_i(0,1)$$

Individual before Mutation

Individual after Mutation

1.: Mutation of individual step sizes

2.: Mutation of object variables

The new individual σ_i' are used here!

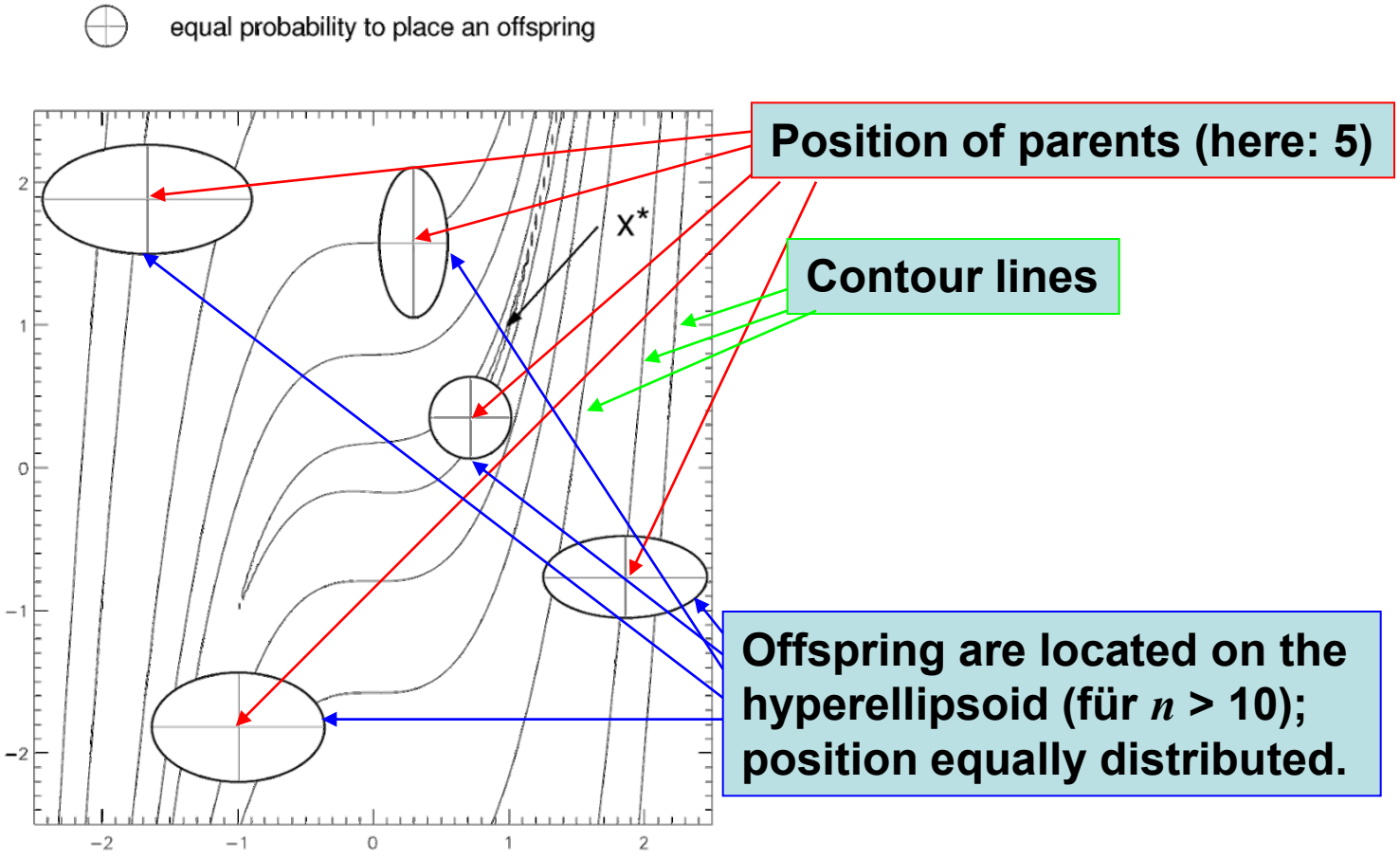
Operators: Mutation – individual σ_i

- τ, τ' are learning rates, again
  - τ': Global learning rate
  - N(0,1): Only one realisation
  - τ: local learning rate
  - $N_i(0,1)$: n realisations
  - Suggested by Schwefel*:

$$\tau' = \frac{1}{\sqrt{2n}} \qquad \tau = \frac{1}{\sqrt{2\sqrt{n}}}$$

*H.-P. Schwefel: Evolution and Optimum Seeking, Wiley, NY, 1995.

Operators: Mutation – individual σ_i



Pros and Cons: Individual σ_i

- Advantages:
  - Individual scaling of object variables
  - Increased global convergence reliability
- Disadvantages:
  - Slower convergence due to increased learning effort
  - No rotation of coordinate system possible
    - Required for badly conditioned objective function

Operators: Correlated Mutations

- Self-adaptive ES with correlated mutations:
  - Individual step sizes
  - One rotation angle for each pair of coordinates
  - Mutation according to covariance matrix: N(**0**, **C**)

$$\vec{a} = ((x_1,\ldots,x_n),(\sigma_1,\ldots,\sigma_n),(\alpha_1,\ldots,\alpha_{n(n-1)/2}))$$

Individual before mutation

$$\vec{a}' = ((x_1',\ldots,x_n'),(\sigma_1',\ldots,\sigma_n'),(\alpha_1',\ldots,\alpha_{n(n-1)/2}'))$$

Individual after mutation

$$\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1))$$

1.: Mutation of Individual step sizes

$$\alpha_j' = \alpha_j + \beta \cdot N_j(0,1)$$

2.: Mutation of rotation angles

$$x_i' = x_i + \vec{N}(\vec{0}, C')$$

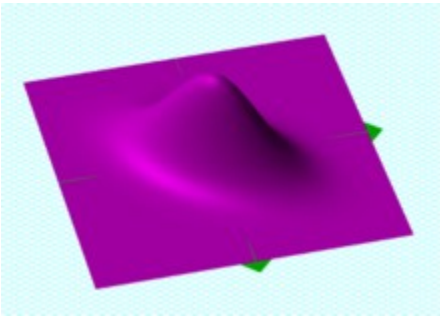
3.: Mutation of object variables

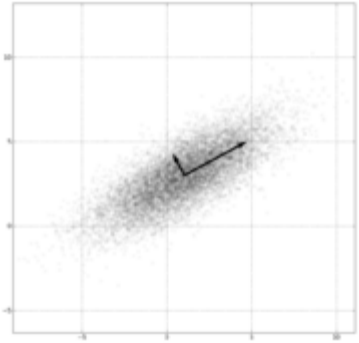
New convariance matrix C‘ used here!

Multivariate Normal Distribution

- C is the covariance matrix

$$f_X(\vec{x}) = \frac{1}{(2\pi)^{n/2}|C|^{1/2}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})' C^{-1} (\vec{x}-\vec{\mu})\right)$$



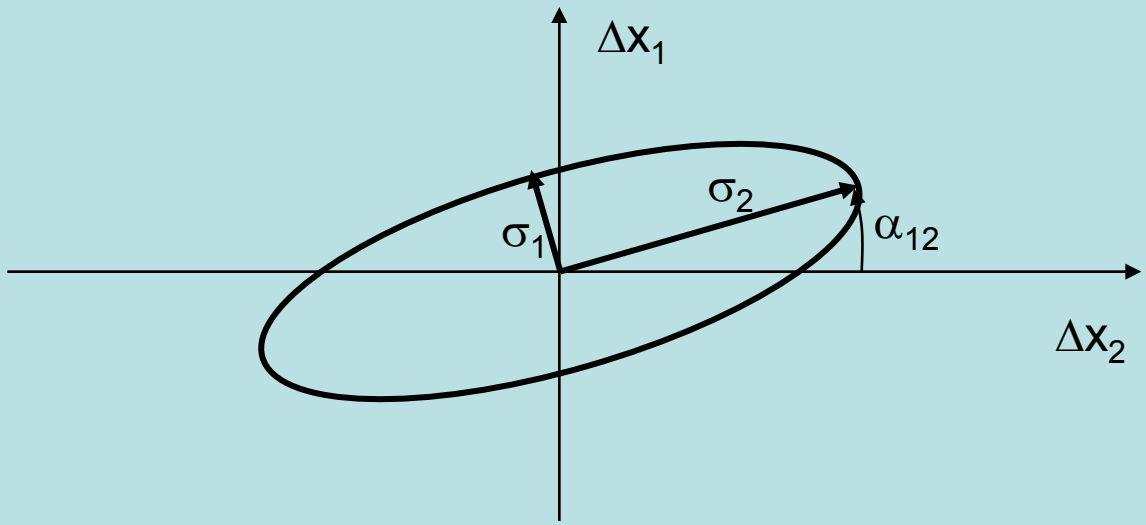


$$C_{ij} = \mathrm{cov}(X_i, X_j) = E\left[(X_i - \mu_i)(X_j - \mu_j)\right]$$

Operators: Correlated Mutations

- Interpretation of rotation angles α_{ij}
- Mapping onto convariances according to

$$c_{ij(i \neq j)} = \frac{1}{2}(\sigma_i^2 - \sigma_j^2)\tan(2\alpha_{ij})$$

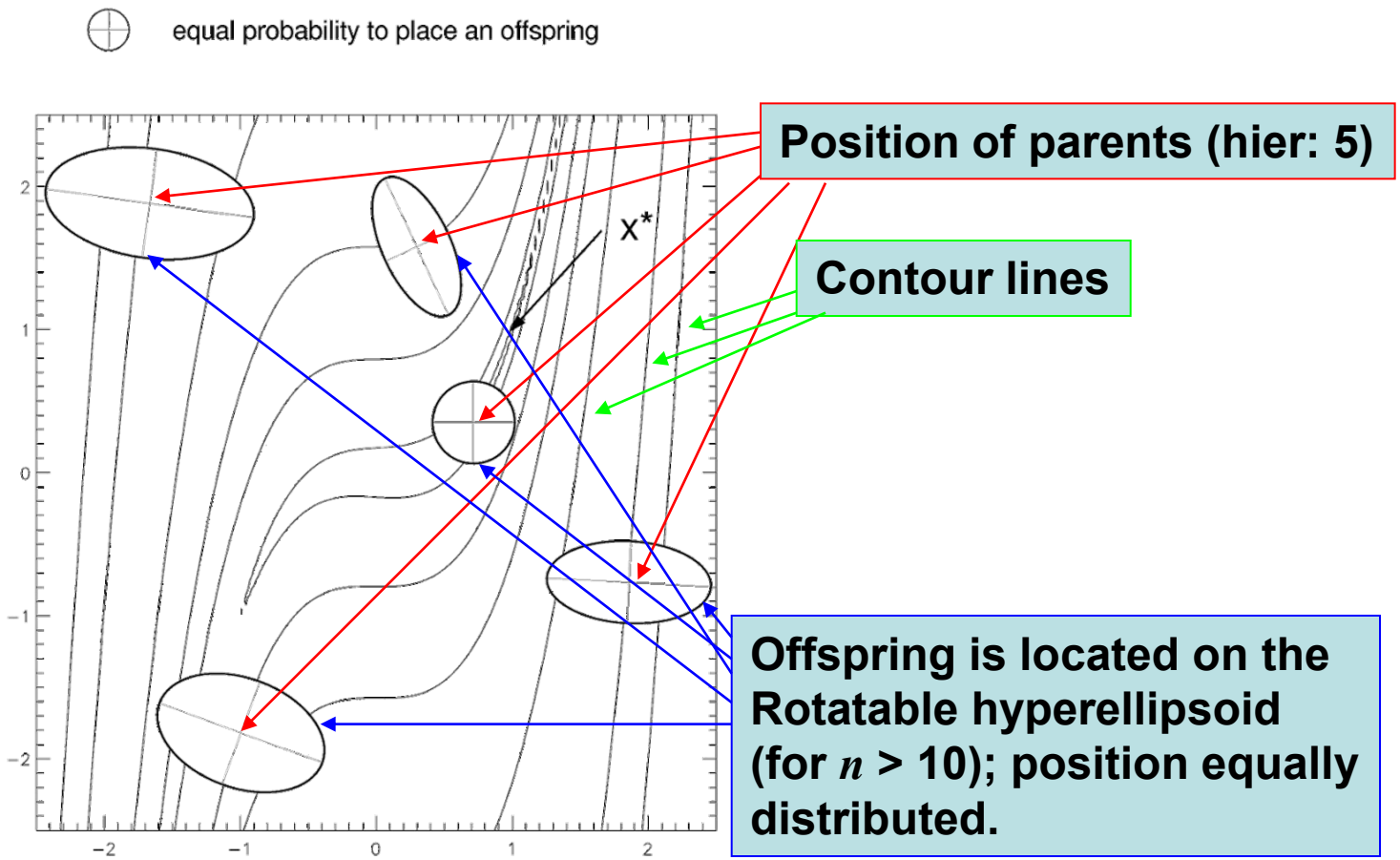


Operators: Correlated Mutation

- τ, τ', β are again learning rates
  - τ, τ' as before
  - β = 0,0873 (corresponds to 5 degree)
  - Out of boundary correction:

$$\left|\alpha'_j\right| > \pi \Rightarrow \alpha'_j \leftarrow \alpha'_j - 2\pi \cdot sign(\alpha'_j)$$

Correlated Mutations for ES



Operators: Correlated Mutations

- How to create $\vec{N}(\vec{0}, C?)$
  - Multiplication of uncorrelated mutation vector with *n(n-1)/2* rotational matrices

$$\vec{\sigma}_c = \prod_{i=1}^{n-1} \prod_{j=i+1}^{n} R(\alpha_{ij}) \cdot \vec{\sigma}_u$$

  - Generates only feasible (positiv definite) correlation matrices

Operators: Correlated Mutations

- Structure of rotation matrix

$$R(\alpha_{ij}) = \begin{pmatrix} 1 & & & & & & & & \\ & 1 & & & & & & 0 & \\ & & \cos(\alpha_{ij}) & & & & -\sin(\alpha_{ij}) & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & \sin(\alpha_{ij}) & & & & \cos(\alpha_{ij}) & & \\ & 0 & & & & & & 1 & \\ & & & & & & & & 1 \end{pmatrix}$$

Operators: Correlated Mutations

- Implementation of correlated mutations

```
nq := n(n-1)/2;
for i:=1 to n do
    σu[i] := σ[i] * N_i(0,1);
for k:=1 to n-1 do
    n1 := n-k;
    n2 := n;
    for i:=1 to k do
        d1     := σu[n1]; d2:= σu[n2];
        σu[n2] := d1*sin(α[nq])+ d2*cos(α[nq]);
        σu[n1] := d1*cos(α[nq])- d2*sin(α[nq]);
        n2     := n2-1;
        nq     := nq-1;
    od
od
```

Generation of the uncorrelated mutation vector

Rotations

Pros and Cons: Correlated Mutations

- Advantages:
  - Individual scaling of object variables
  - Rotation of coordinate system possible
  - Increased global convergence reliability
- Disadvantages:
  - Much slower convergence
  - Effort for mutations scales quadratically
  - Self-adaptation very inefficient

Operators: Mutation – Addendum

- Generating N(0,1)-distributed random numbers?

$$u = 2 \cdot U[0,1) - 1$$

$$v = 2 \cdot U[0,1) - 1$$

$$w = u^2 + v^2$$

$$\left.\begin{aligned} x_1 &= u \cdot \sqrt{\frac{-2\log(w)}{w}} \\ x_2 &= v \cdot \sqrt{\frac{-2\log(w)}{w}} \end{aligned}\right\} \text{ If } w > 1$$

$$x_1, x_2 \sim N(0,1)$$

The idea behind mutations?

- Biological model: Repair enzymes, mutator genes
- No deterministic control: Strategy parameters evolve
- Indirect link between fitness and useful strategy parameter settings
- $\vec{\sigma}, \vec{\alpha}$ are conceivable as an internal model of the local topology

Evolution Strategy:

Algorithms
Recombination

Operators: Recombination

- Only for $\mu > 1$
- Directly after Selektion
- Iteratively generates λ offspring:

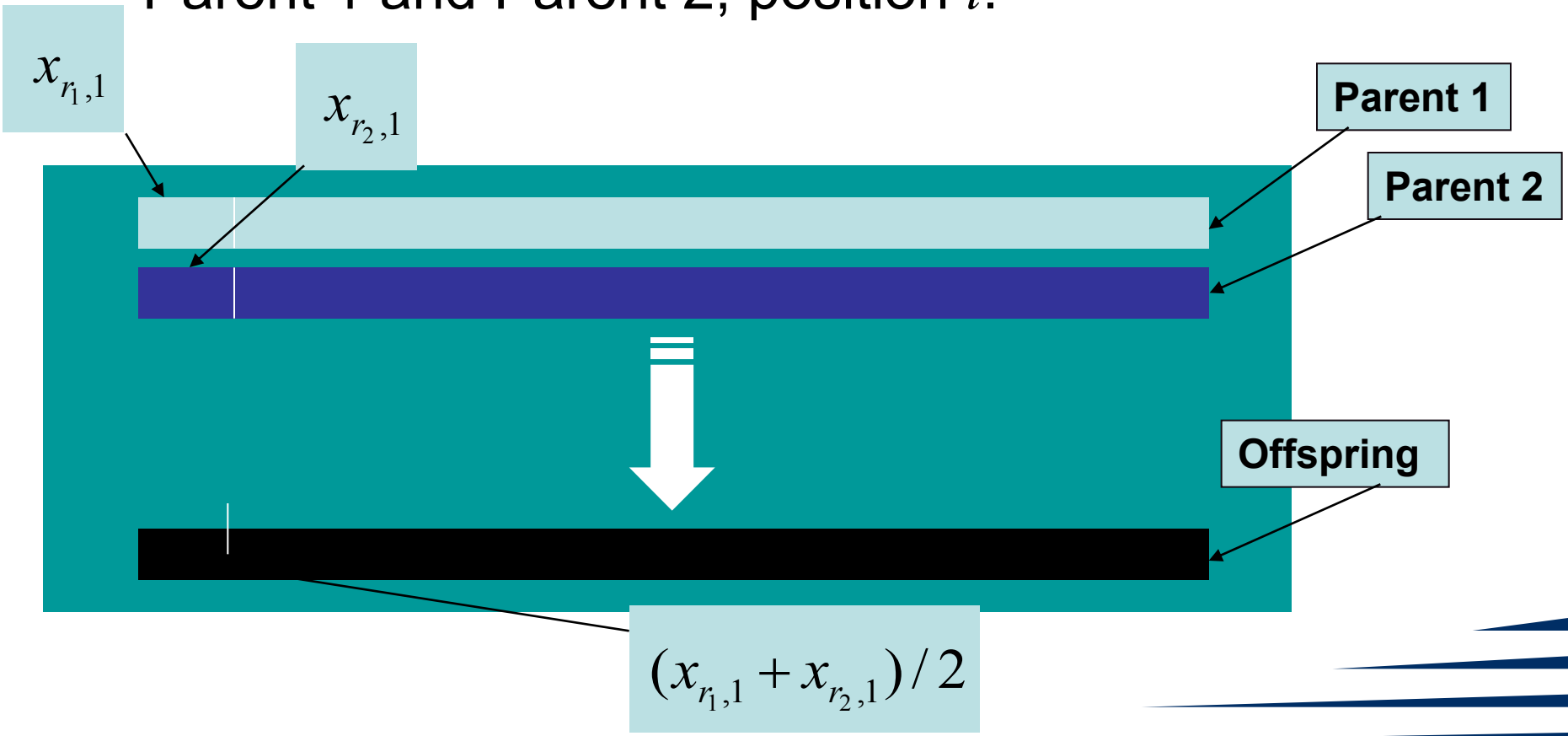
```
for i:=1 to λ do
   choose recombinant r1 uniformly at random
        from parent_population;
   choose recombinant r2 <> r1 uniformly at random
        from parent population;
   offspring := recombine(r1,r2);
   add offspring to offspring_population;
od
```

Operators: Recombination

- How does recombination work?
- Discrete recombination:
  - Variable at position i will be copied at random (uniformly distr.) from parent 1 or parent 2, position i.

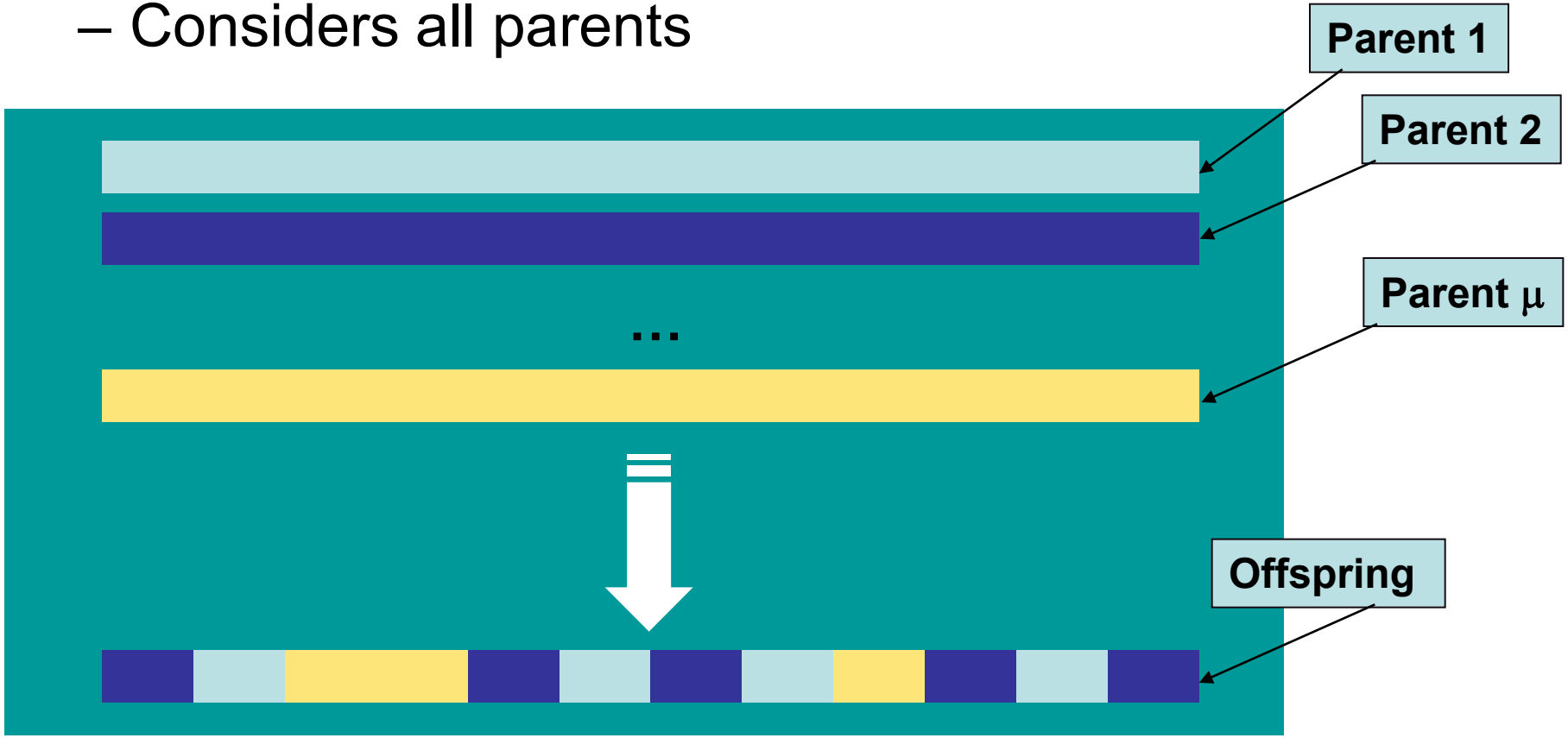
Operators: Recombination

- Intermediate recombination:
  - Variable at position i is arithmetic mean of Parent 1 and Parent 2, position i.



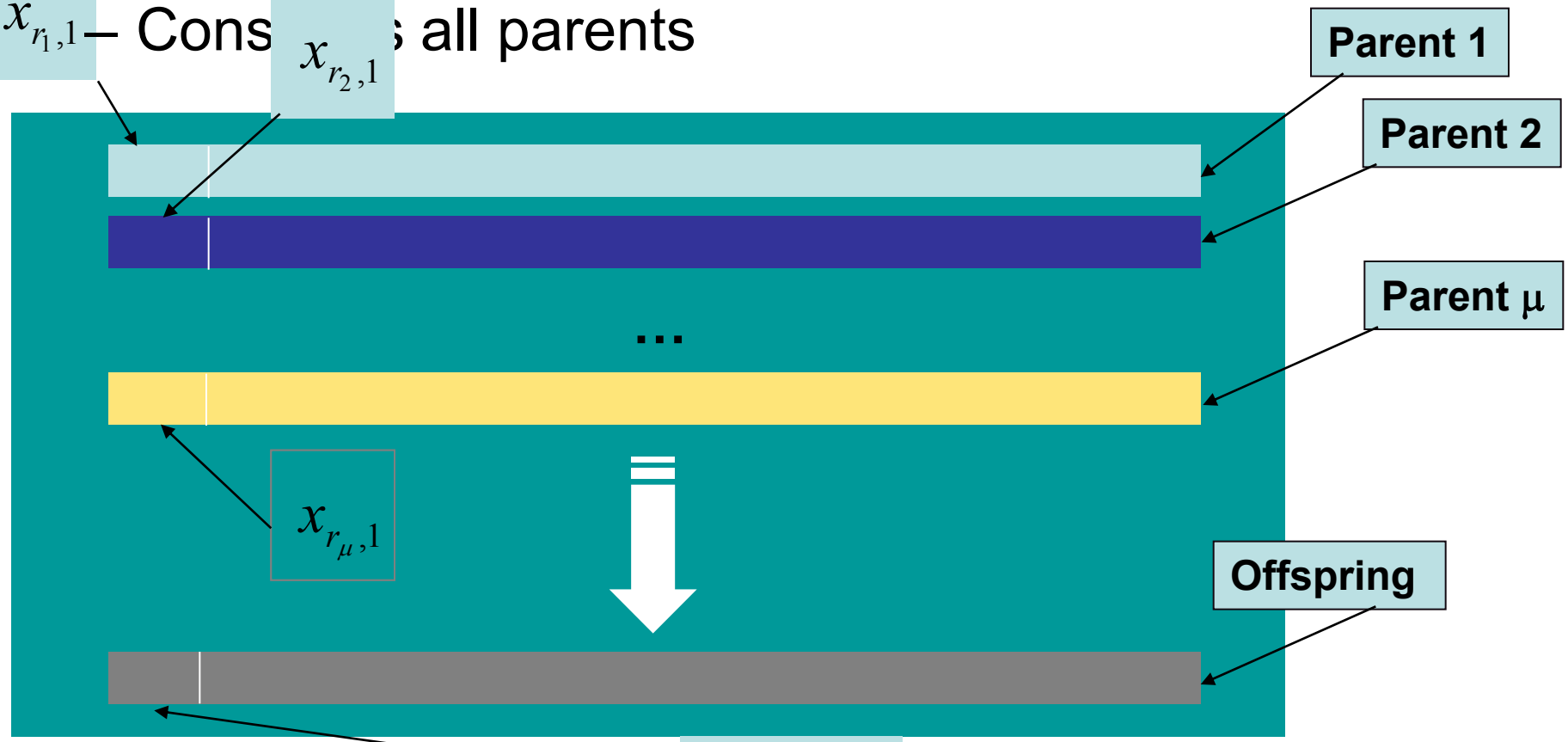
Operators: Recombination

- Global discrete recombination:
  - Considers all parents



Operators: Recombination

- Global intermediary recombination:
  - Cons s all parents



$$\frac{1}{\mu}\sum_{i=1}^{\mu} x_{r_i,1}$$

Evolution Strategy

Algorithms
Selection

Operators: (μ+λ)-Selection

= Actual solution candidate

= New solution candidate

- (μ+λ)-Selection means:
  - μ parents produce λ offspring by
    - (Recombination +)
    - Mutation
  - These μ+λ individuals will be considered together
  - The μ best out of μ+λ will be selected („survive“)
    - Deterministic selection
  - This method guarantees monotony
    - Deteriorations will never be accepted

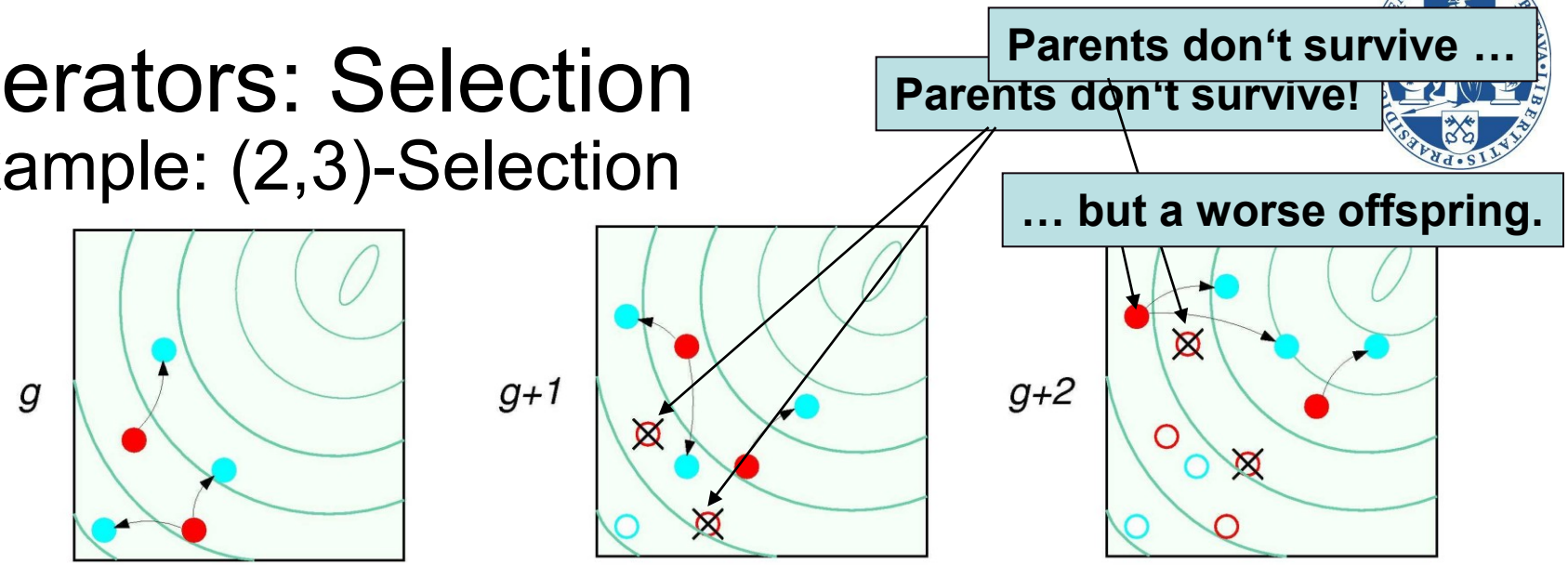
Recombination may be left out
Mutation always exists!

Operators: (μ,λ)-Selection

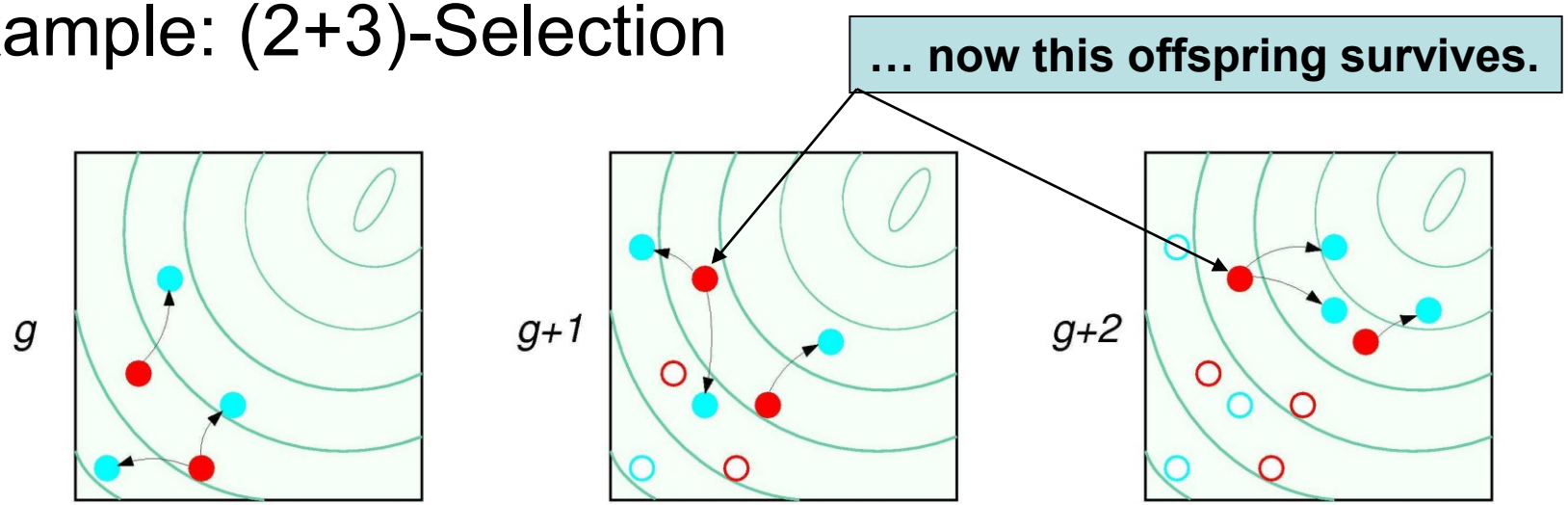
- (μ,λ)-Selection means:
  - μ parents produce $\lambda >> \mu$ offspring by
    - (Recombination +)
    - Mutation
  - λ offspring will be considered alone
  - The μ best out of λ offspring will be selected
    - Deterministic selection
  - The method doesn't guarantee monotony
    - Deteriorations are possible
    - The best objective function value in generation $t+1$ may be worse than the best in generation t.

Operators: Selection

- Example: (2,3)-Selection



- Example: (2+3)-Selection



Operators: Selection

Exception!

- Possible occurrences of selection
  - (1+1)-ES: One parent, one offspring, 1/5-Rule
  - (1,λ)-ES: One Parent, λ offspring
    - Example: (1,10)-Strategy
    - One step size / n self-adaptive step sizes
    - Mutative step size control
    - Derandomized strategy
  - (μ,λ)-ES: $\mu > 1$ parents, $\lambda > \mu$ offspring
    - Example: (2,15)-Strategy
    - Includes recombination
    - Can overcome local optima
  - (μ+λ)-strategies: elitist strategies

Selective pressure

- The „selective pressure“ of this method is very high.
- How to define this?
- Takeover time τ^*:
  - The number of generations until repeated application of selection completely fills the population with copies of the initially best individual.
- For (μ,λ)-selection: $$\tau^* = \frac{\ln \lambda}{\ln(\lambda / \mu)}$$
- For a standard (15,100)-ES: $\tau^* \approx 2$
- Proportional selection in genetic algorithms: $\tau^* \approx \lambda \ln \lambda = 460$

➔ There is a huge difference!

Other Aspects

- Initialization:
  - x_i, α_i typically at random
  - $\sigma_i \approx \delta x_i / \sqrt{n}$ with δx_i being a rough measure of distance to the optimum
  - Or: Feasible range $[x_{i,max}; x_{i,min}]/6$
- Termination:
  - Typically after a given number of generations
  - Or function evaluations
  - Or some population diversity measure
  - Or some improvement measure

Evolution Strategy:

Reproduction Cycle

Generational ES Model

$t := 0$;

initialize $P(t)$;

evaluate $P(t)$;

while not terminate do

$P'(t)$:= recombine($P(t)$);

$P''(t)$:= mutate($P'(t)$);

$P(t+1)$:= select($P''(t) \cup P(t)$); // or: $P(t+1)$:= select($P''(t)$);

$t := t+1$;

od

return(best individual found);

Remarks

- Recombine: Recombination is applied to all individuals!
- P‘(t) has size $\lambda > \mu$, P(t) size μ.
- Mutate: Normally distributed variations, applied to all individuals.
- Select: $(\mu+\lambda)$ or (μ,λ).