Evolution strategies
Presented by:
Faramarz Safi (Ph.D.)
Faculty of Computer Engineering
**Islamic Azad University, Najafabad Branch**

Chapter 4

Evolution Strategies

- Similar to Genetic Algorithms
  - find a (near-)optimal solution to a problem within a search space (all possible solutions)
- Developed by Ingo Rechenberg Germany in 1970's, independently from genetic algorithms
- Often used for empirical experiments, numerical optimisation
- Attributed features:
  - fast
  - good optimizer for real-valued optimisation
- Special:
  - self-adaptation of (mutation) parameters standard
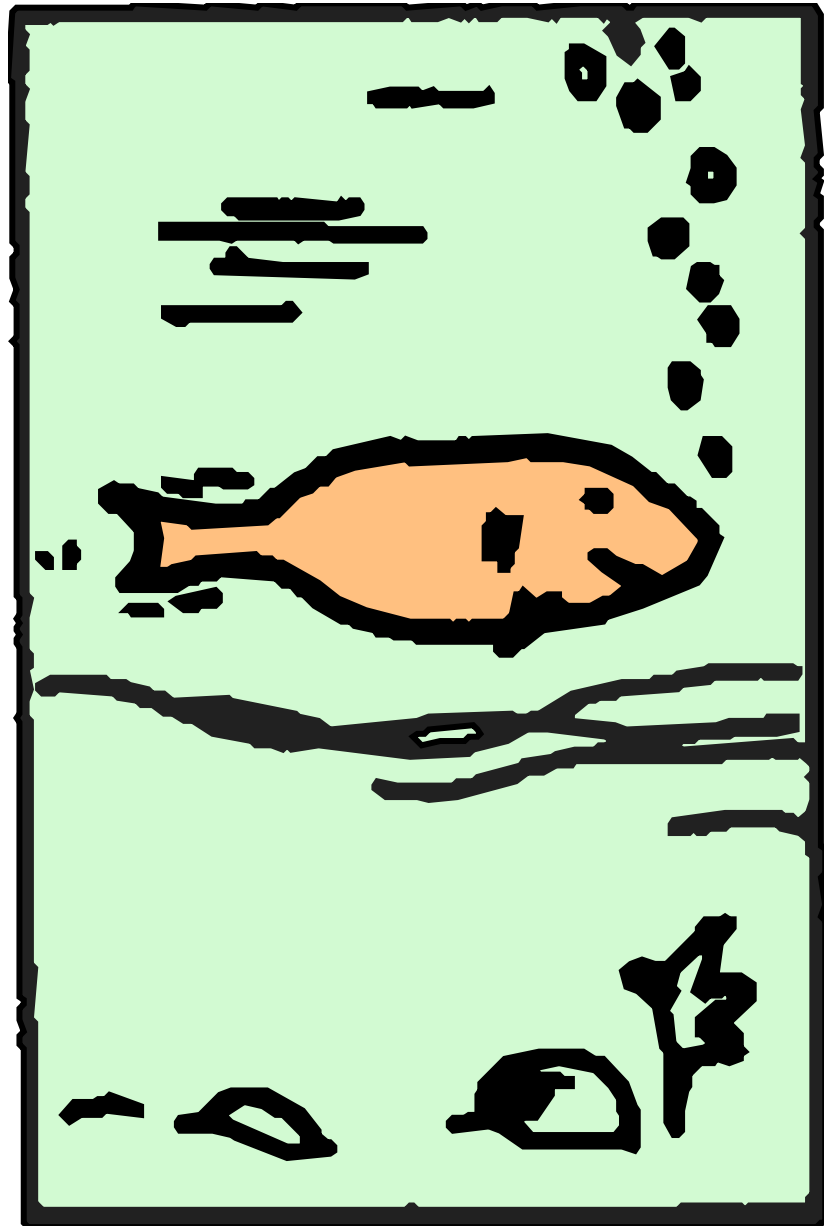- Based on principal of strong causality: Small changes have small effects
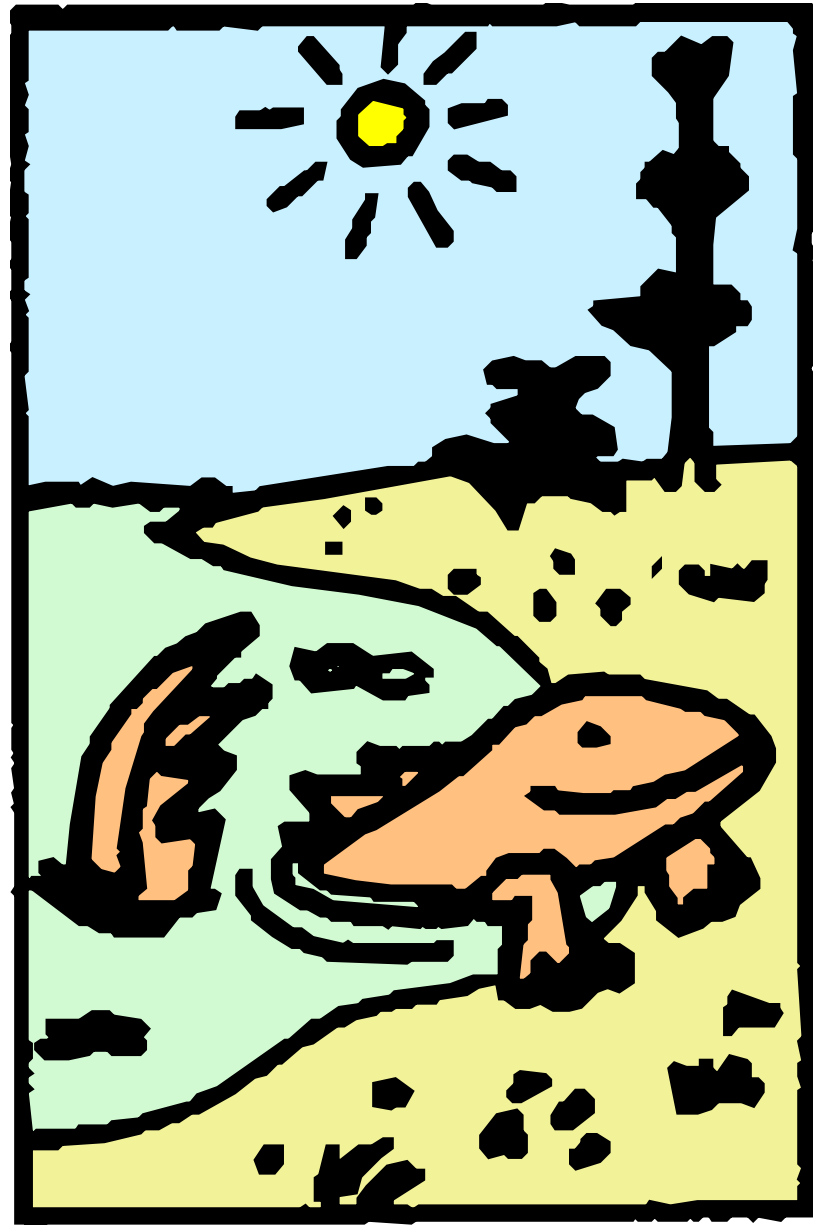
1

Protoplasm lump in the primordial ocean
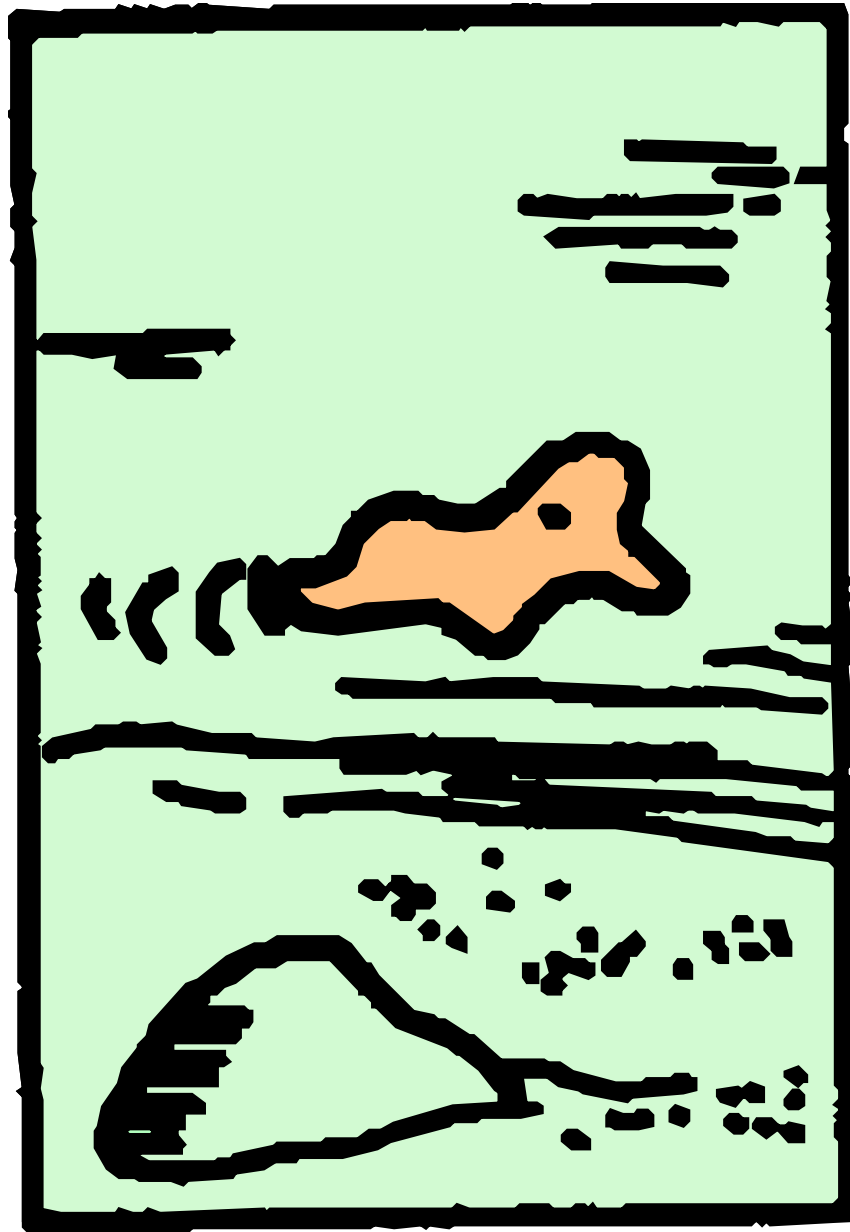
2

From this the fish developed

3

Life peeks out of the water and spreads over the country

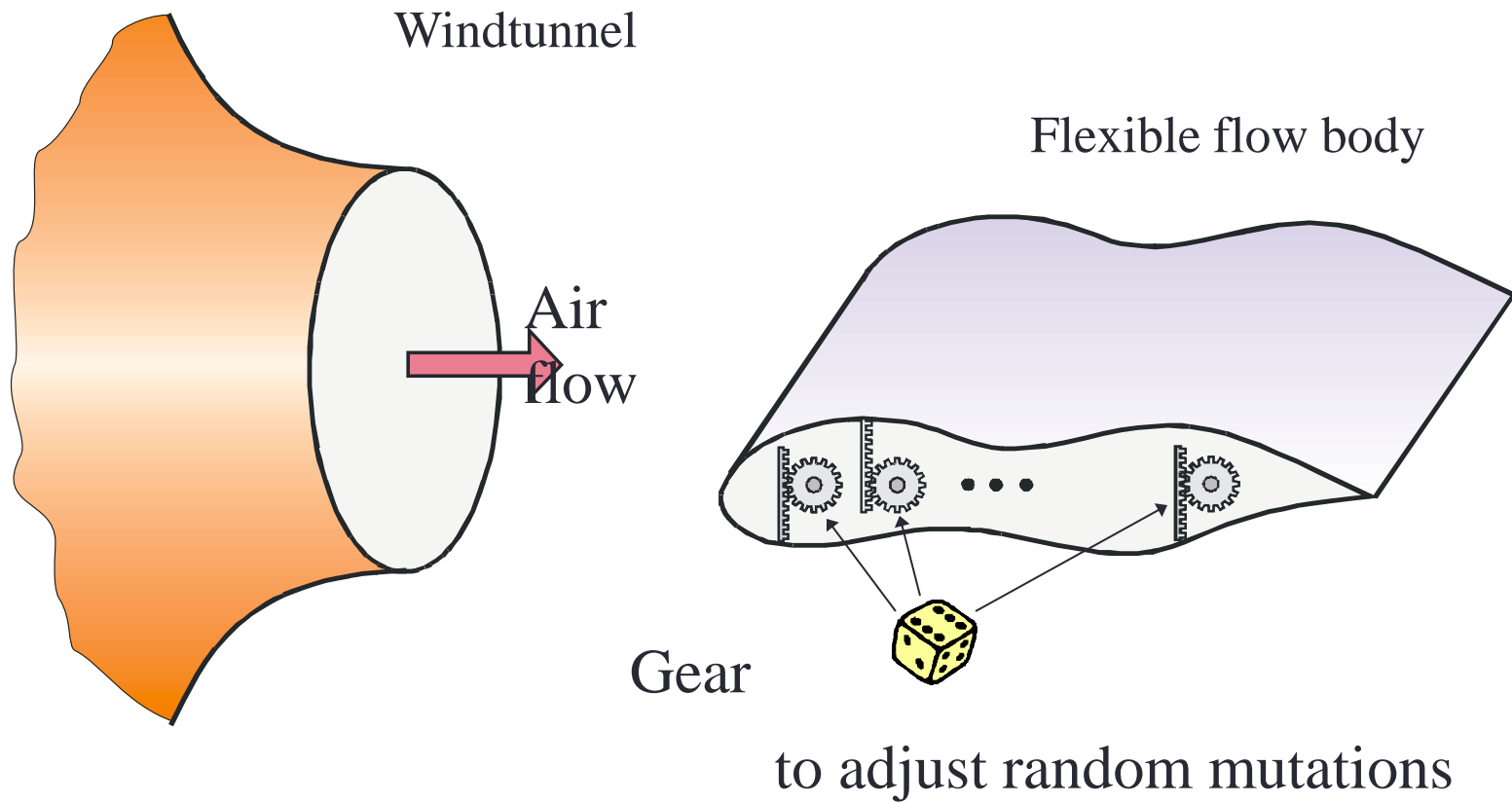**What Evolution does**

4

Our ancestors climb the treetops

5

Finally we admire ourselves in the mirror

History of the Evolution Strategy

Windtunnel

Flexible flow body

Air flow

Gear

to adjust random mutations
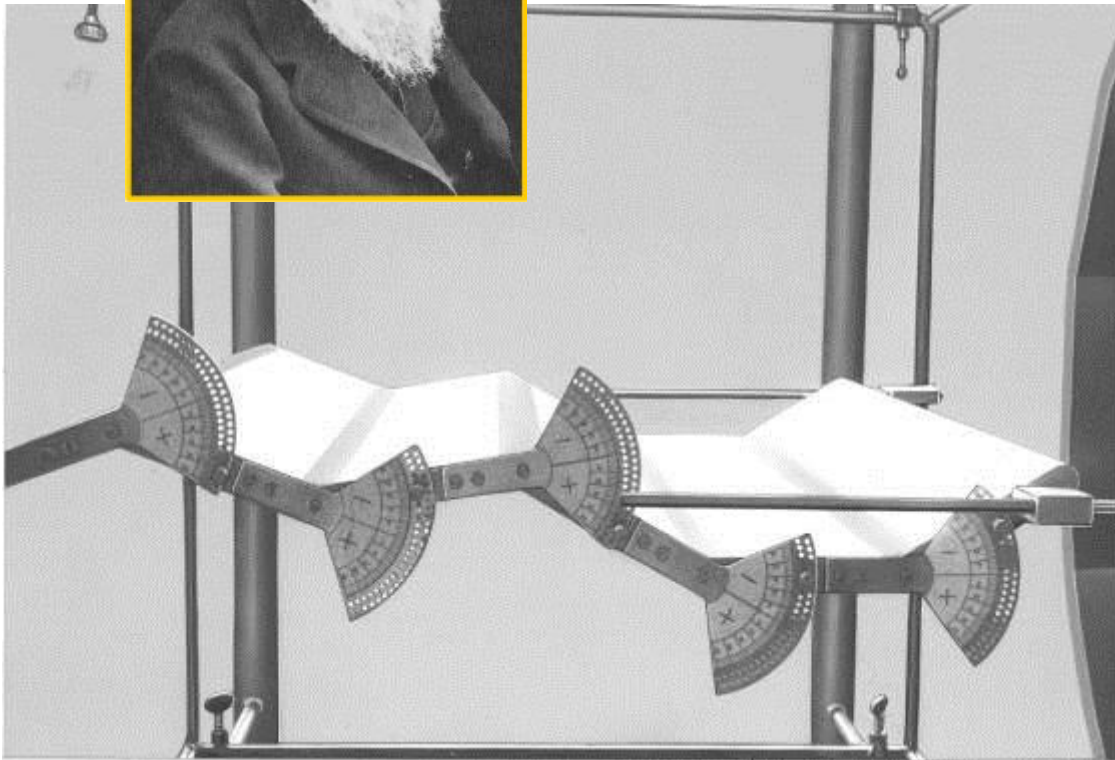
History of the Evolution Strategy

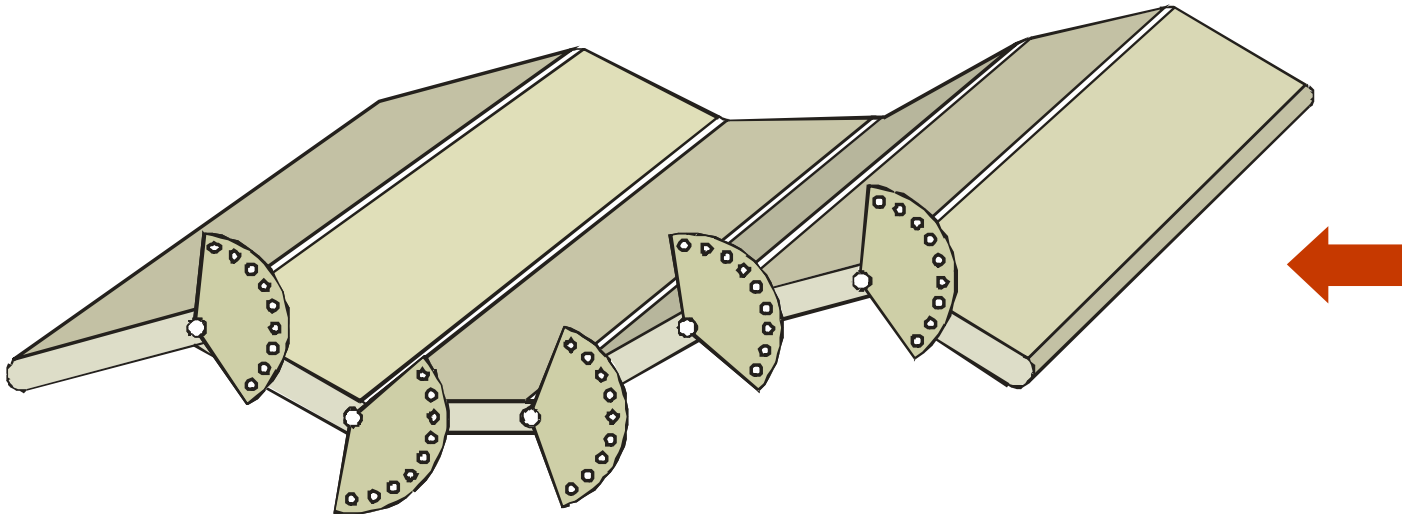DARWIN in the windtunnel

The kink plate for the key experiment with the Evolution Strategy

History of the Evolution Strategy

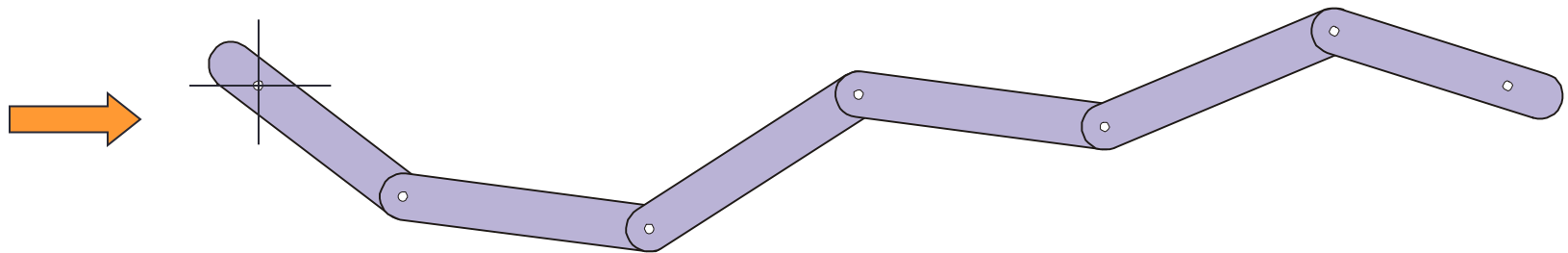**Number of possible adjustments**

$$51^5 = 345\ 025\ 251$$

History of the Evolution Strategy

The first 12 mutations

History of the Evolution Strategy

1

History of the Evolution Strategy
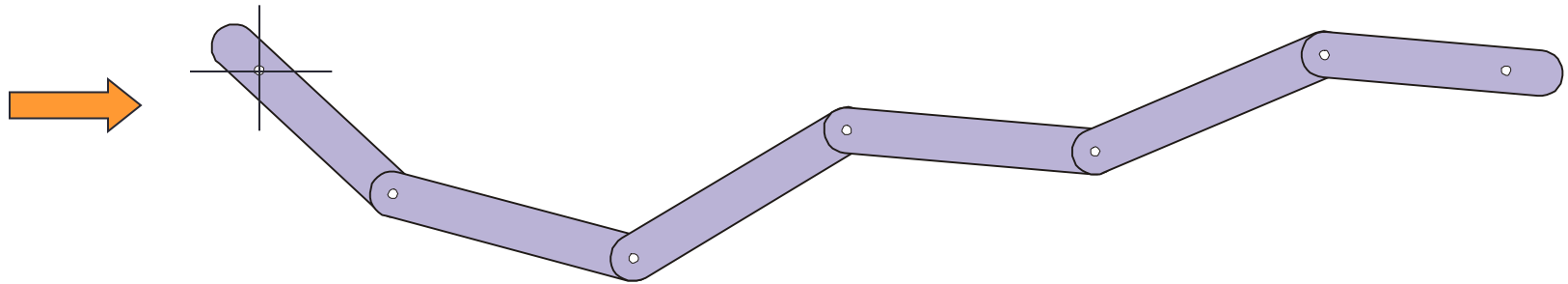
2

3

History of the Evolution Strategy
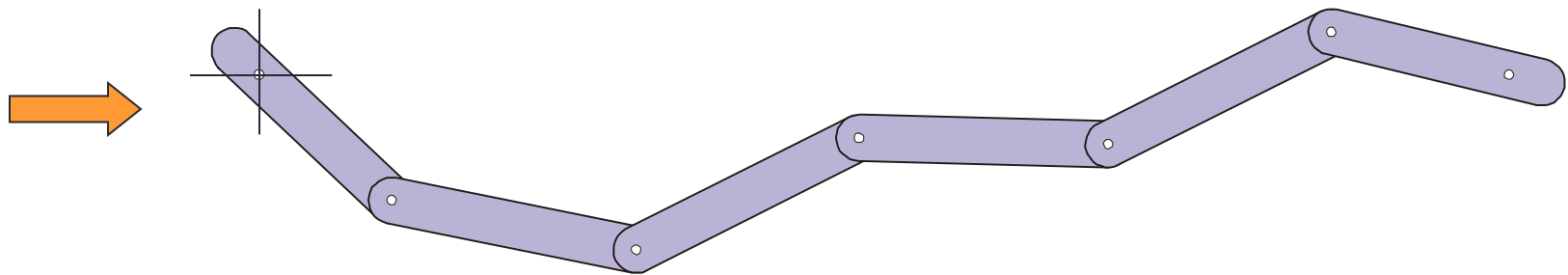
4
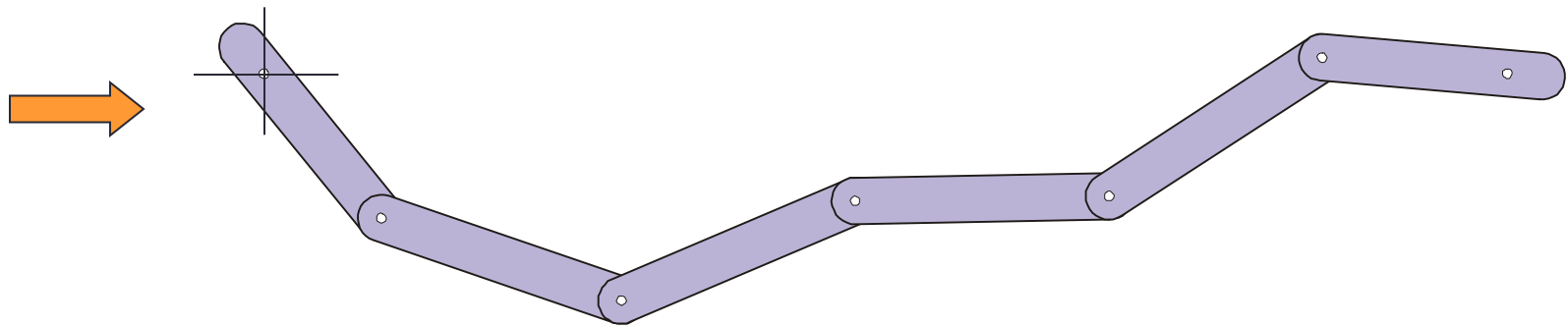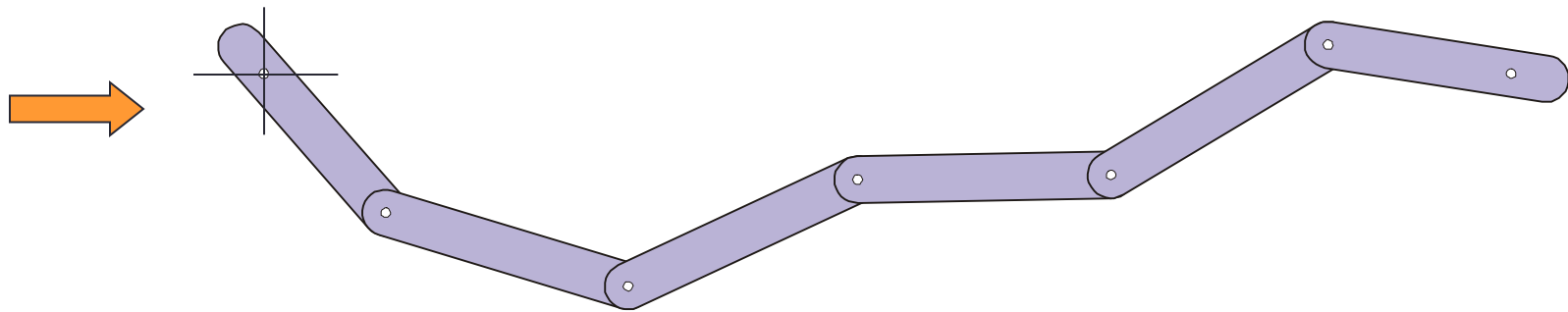
5

6

7

8

9
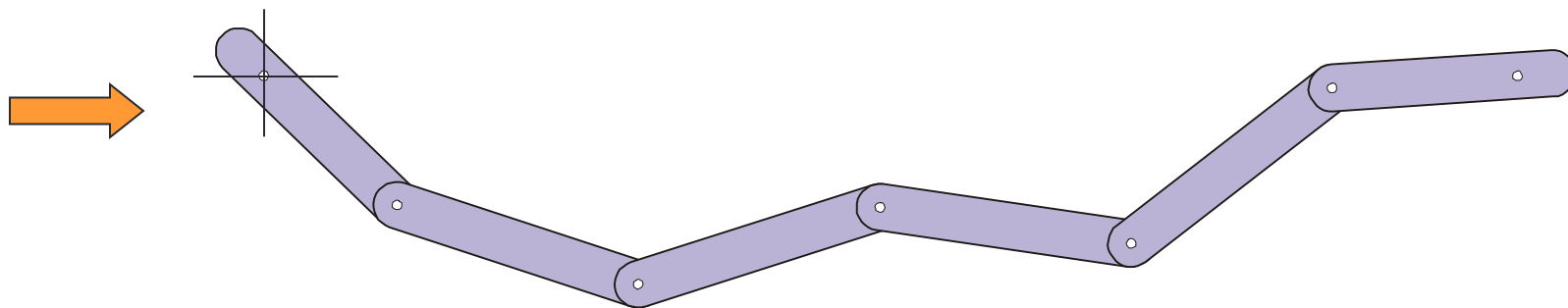
10

History of the Evolution Strategy
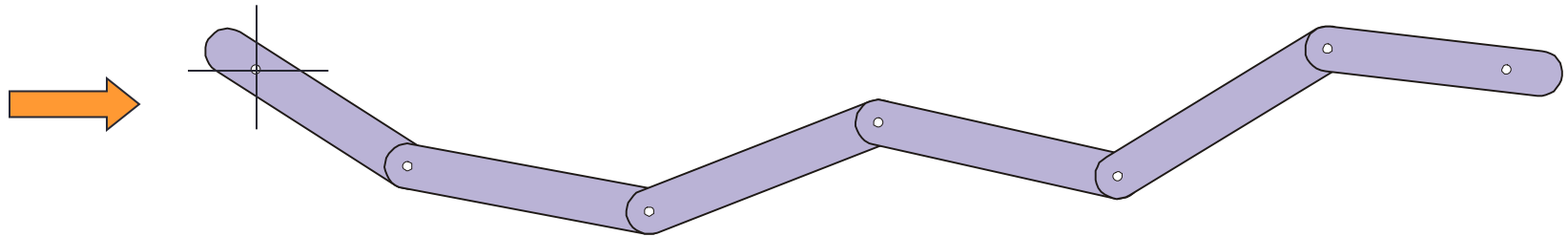
11

History of the Evolution Strategy

12

History of the Evolution Strategy

The *experimentum crucis* – Drag minimization of the kink plate
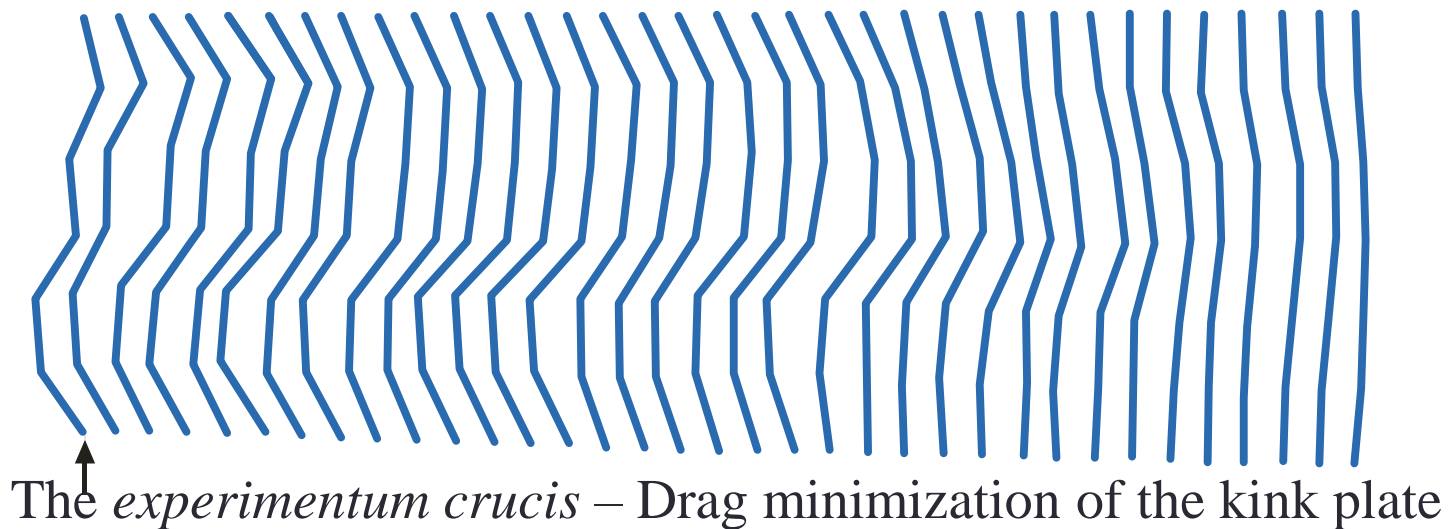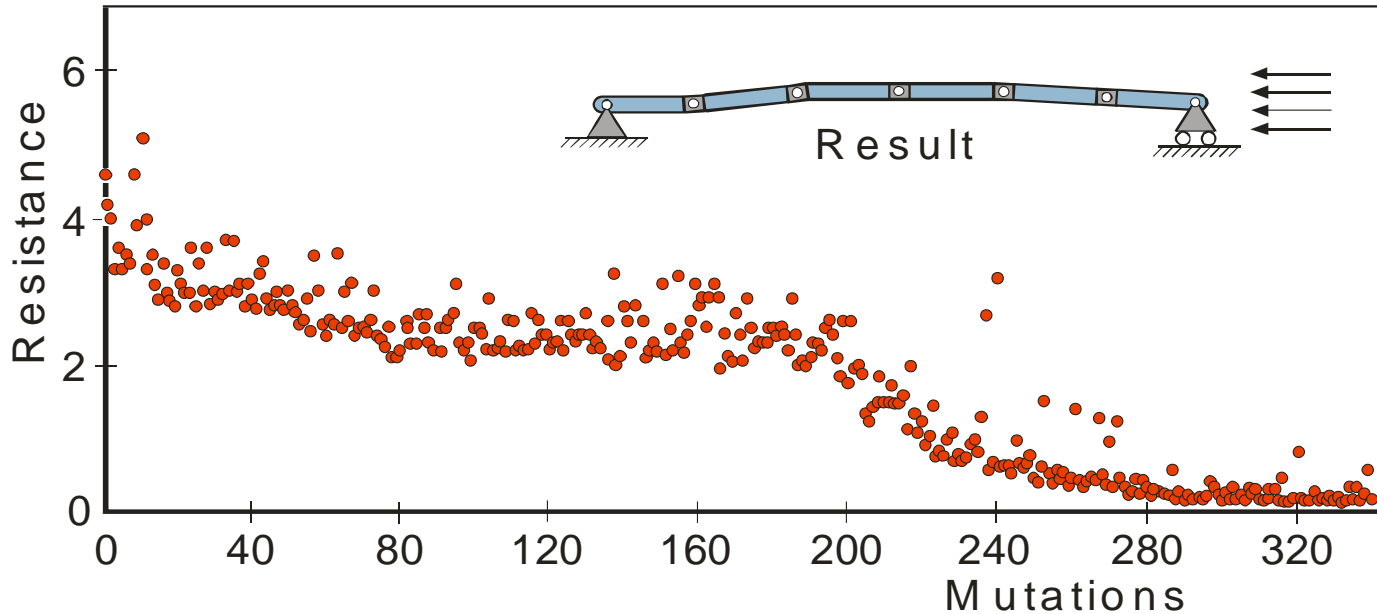
History of the Evolution Strategy

Six manually adjustable shafts determine the form of a 90°pipe bend

Evolution of a 90° pipe bend

Start

Result
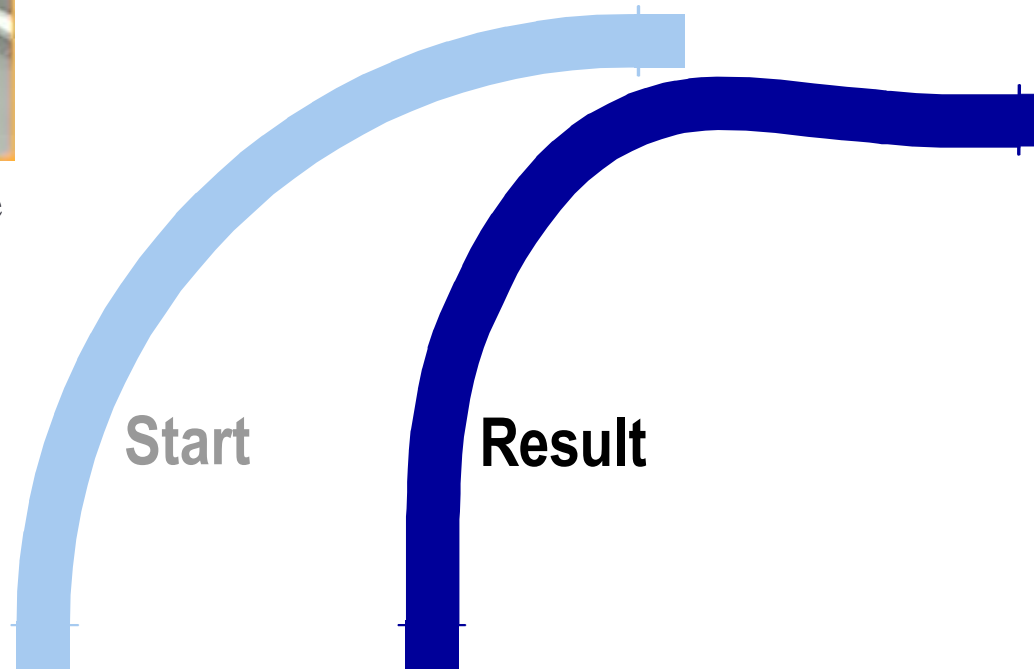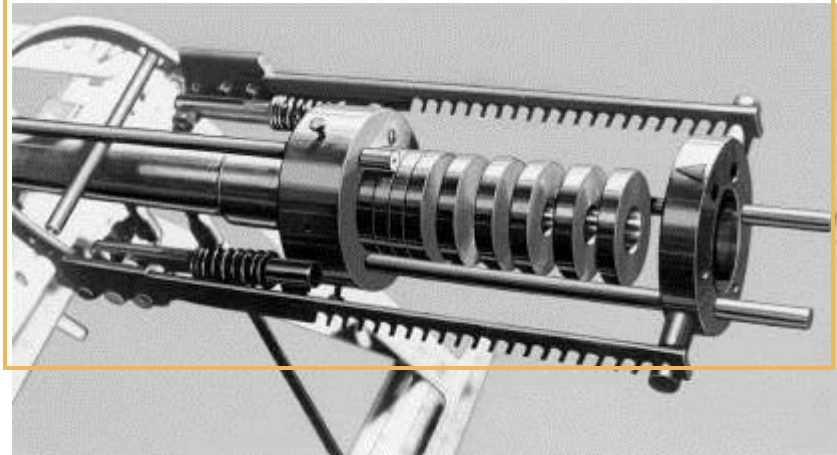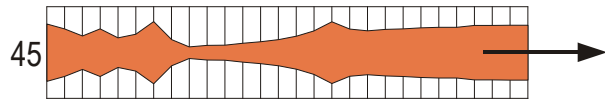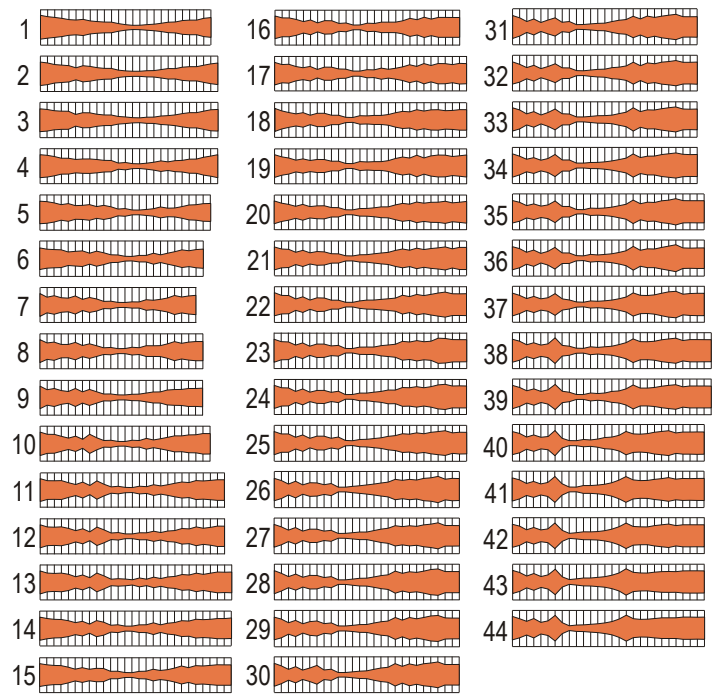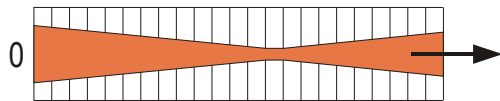
Evolution of a two
phase flow nozzle

(Hans-Paul
Schwefel)

Mimicry in biological evolution

Bad tasting

A blue jay eats
a monarch

But it does't taste

Because of nausea
the feathers struggle

Out with the poison

And the teaching
isn't forgotten

Subjective selection in nature

Mimicry in biological evolution

Bad tasting

Good tasting

Continuation of the evolution: From of a square to
a Mercedes-Star through subjective selection

Actual figure
after:

Generation 1

Generation 20

Generation 40

Generation 60

Generation 80

Generation 100

Generation 200
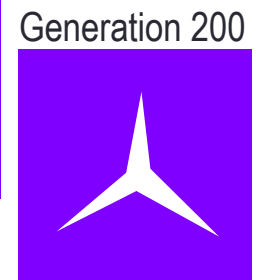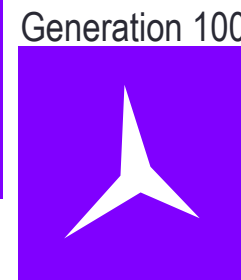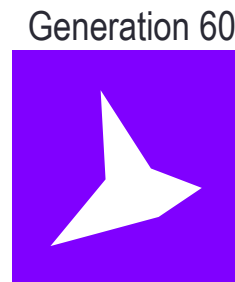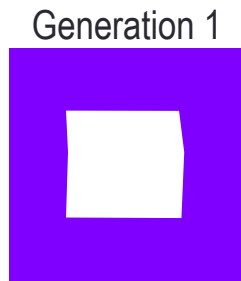
Subjective Selection

Coffee-composition using the Evolution Stratey

Mix of the offspring

Target coffee
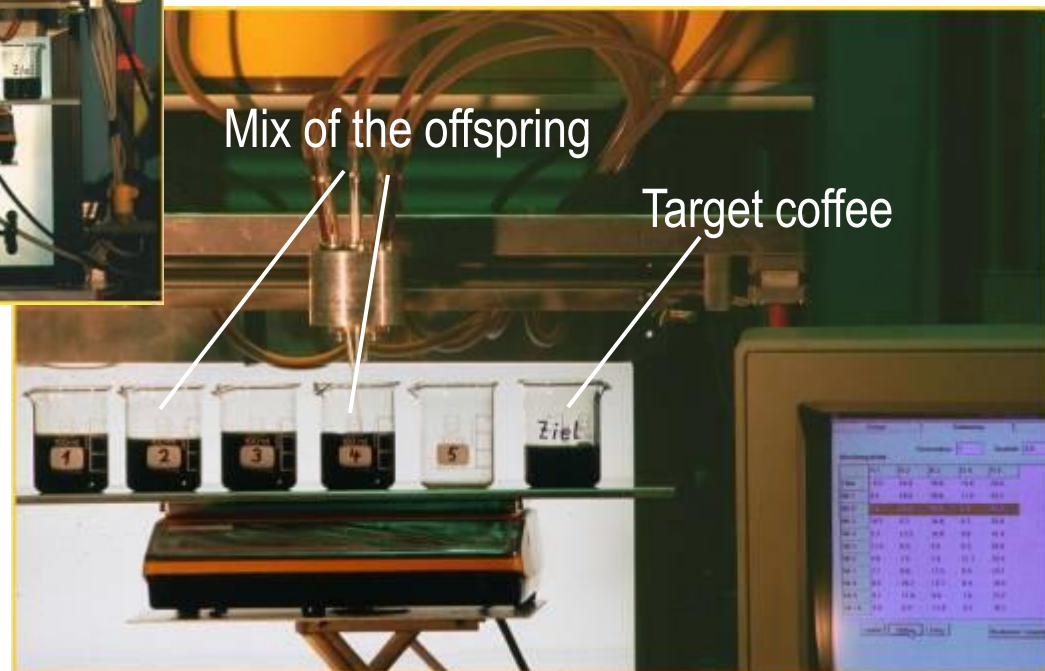
Parent

25%	Columbia
40%	Sumatra
13%	Java
5%	Bahia
17%	Jamaica

Offspring 1

20%	C...
34%	S...
23%	Ja...
5%	B...
18%	Ja...

Offspring 2

23%	C...
37%	Su...
12%	Ja...
10%	B...
18%	Ja...

Offspring 3

25%	Columbia
32%	Su...
15%	Ja...
8%	B...
20%	Ja...

Offspring 4

30%	Columbia
38%	Su...
8%	Ja...
2%	B...
22%	Ja...

Offspring 5

33%	Columbia
38%	Sumatra
9%	Java
8%	Bahia
12%	Jamaica

Subjective evaluation

M. Herdy
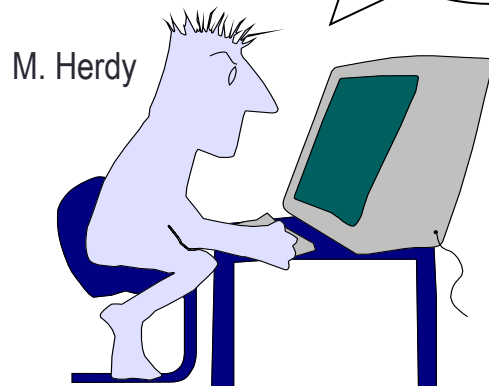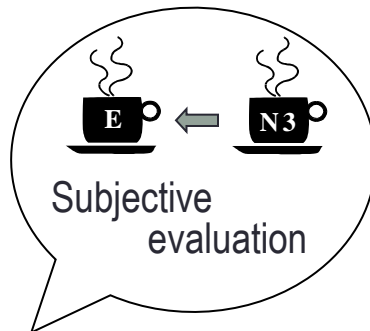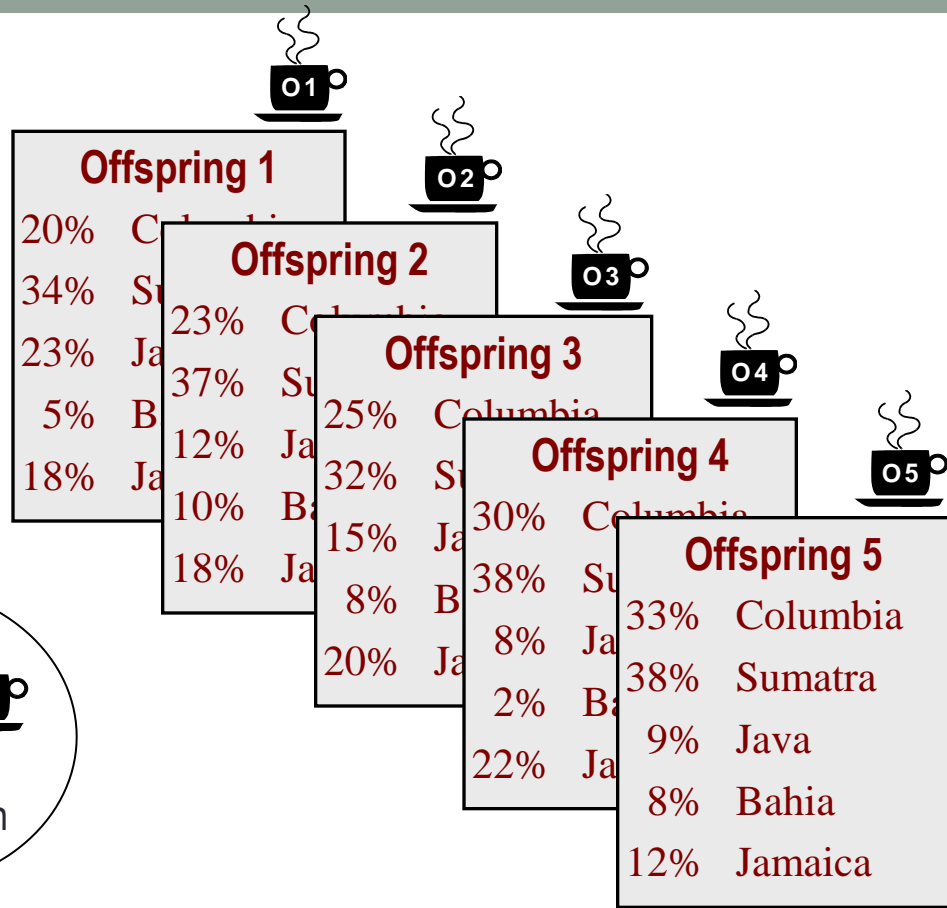
Evolution-strategic development of a coffee blend
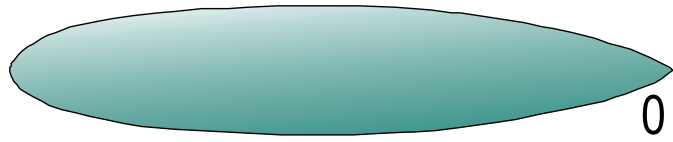
0
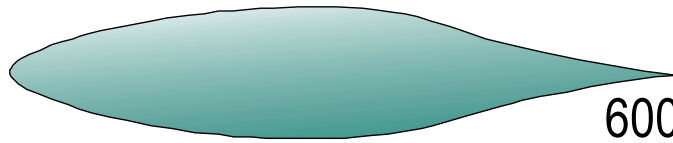
600
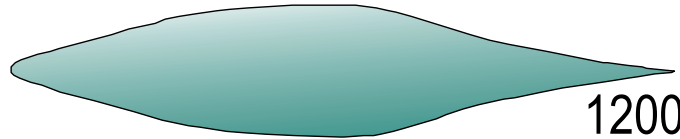
1200
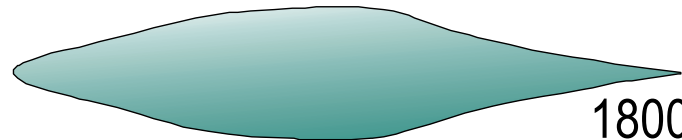
1800
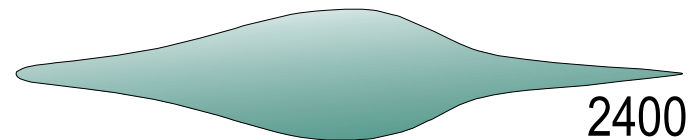
2400

Evolution of a hull spindle with least resistance
(Cross-section an length of the body are given)
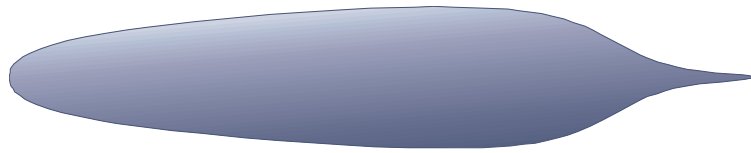
$$\text{Re} = 5 \cdot 10^6$$
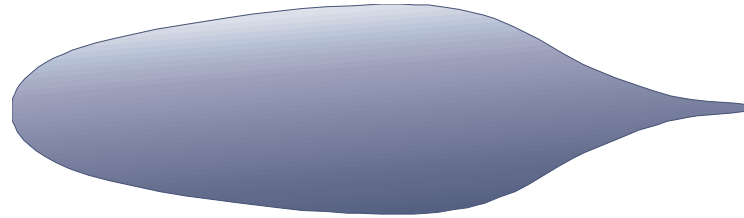
$$\text{Re} = 1 \cdot 10^7$$
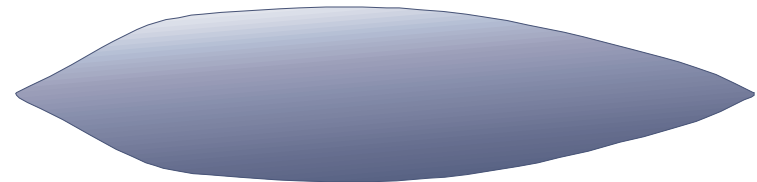
$$\text{Re} = 1 \cdot 10^8$$

Airship bodies found by Evolutionary Computation (EC)

(T. Lutz, Stuttgart)

I. Rechenberg   HFI 1965

ausgelesene
Mutationen

Schema der Evolutionsstrategie

Idea for a mechanical evolution experiment
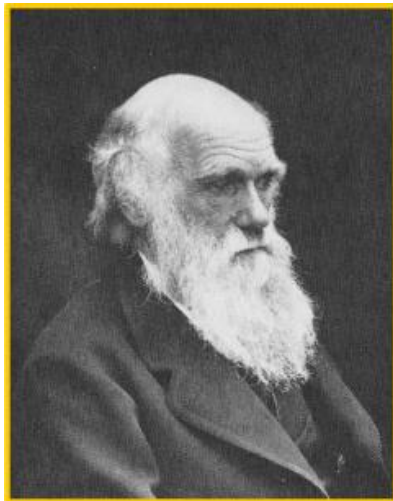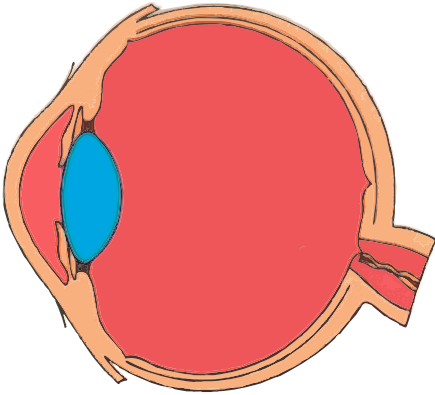
To suppose that the eye, with all its inimitable contrivances for adjusting the focus to different distances, for admitting different amounts of light, and for the correction of spherical and chromatic abberation, **could have been formed by natural selection, seems, I freely confess, absurd in the highest possible degree.**

From Charles Darwin: *»The origin of species«*

Evolution of an eye lens

Computer simulated evolution of a covergent lens

Flexible glass body

d_k

F

q_k
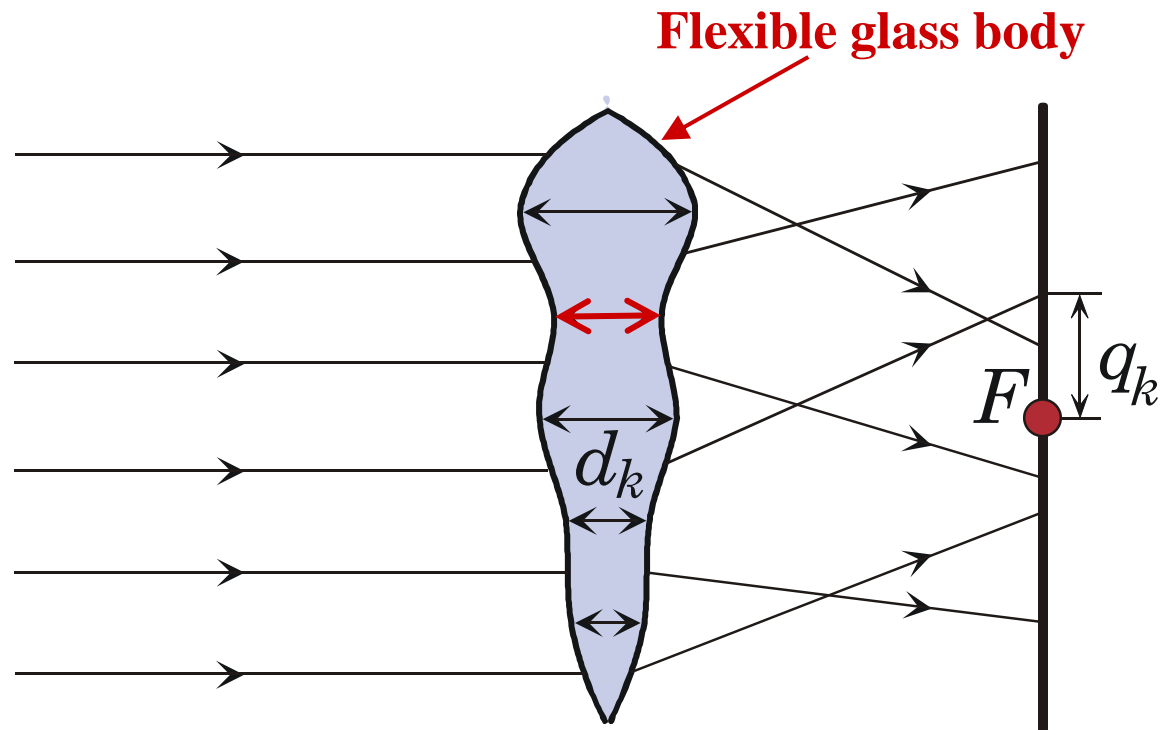
$$\sum q_k^2 \rightarrow \text{Minimum}$$
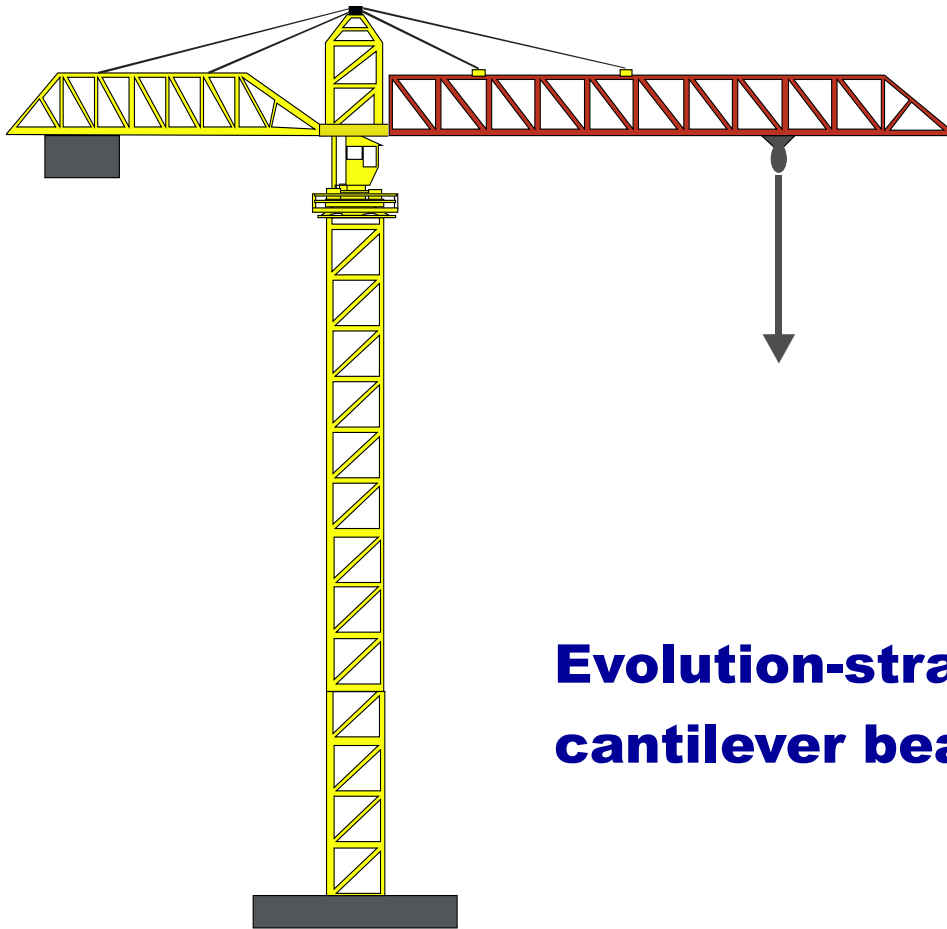
Evolution-strategic optimization of a cantilever beam with minimum weight
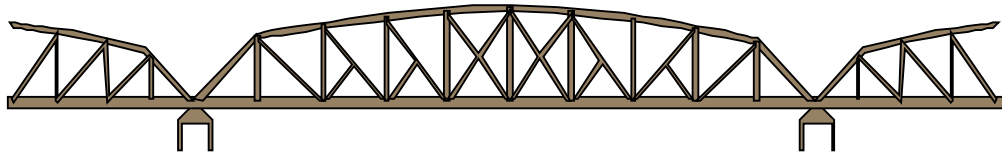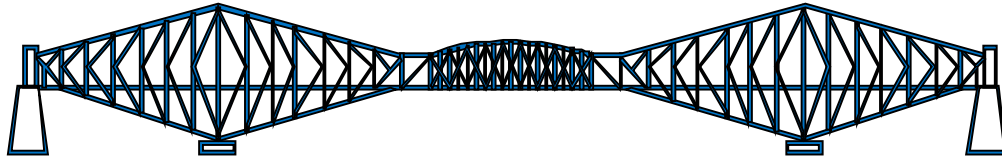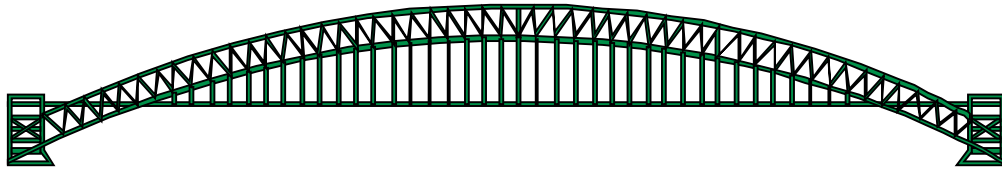
**Evolution-strategic optimization of
a truss bridge with minimum weight**
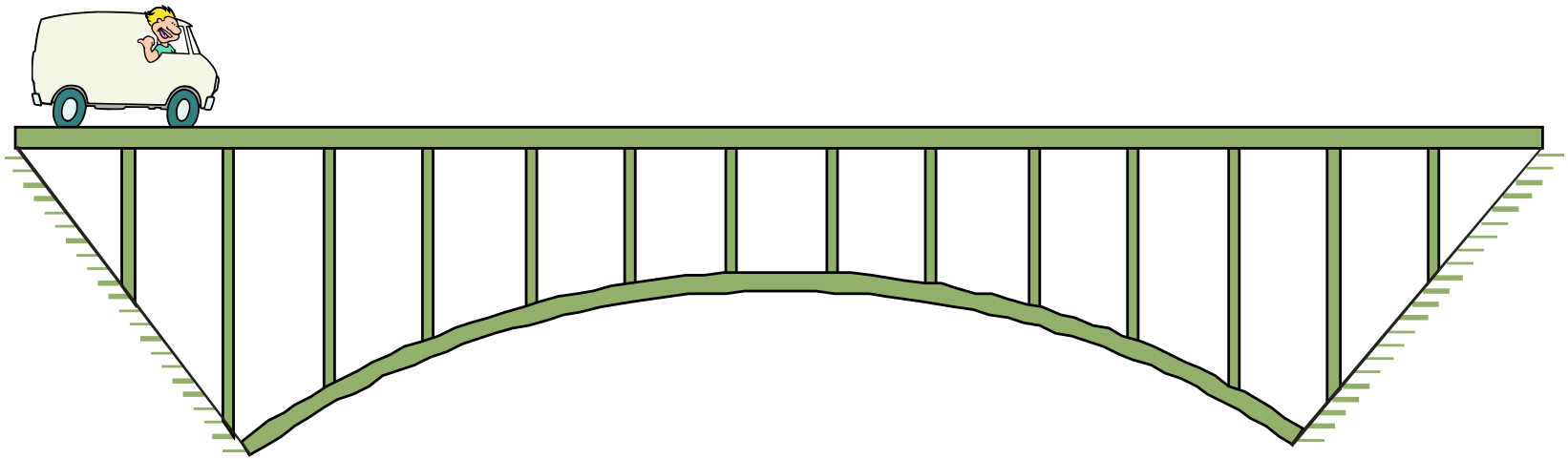
Dynamic optimization of a truss bridge
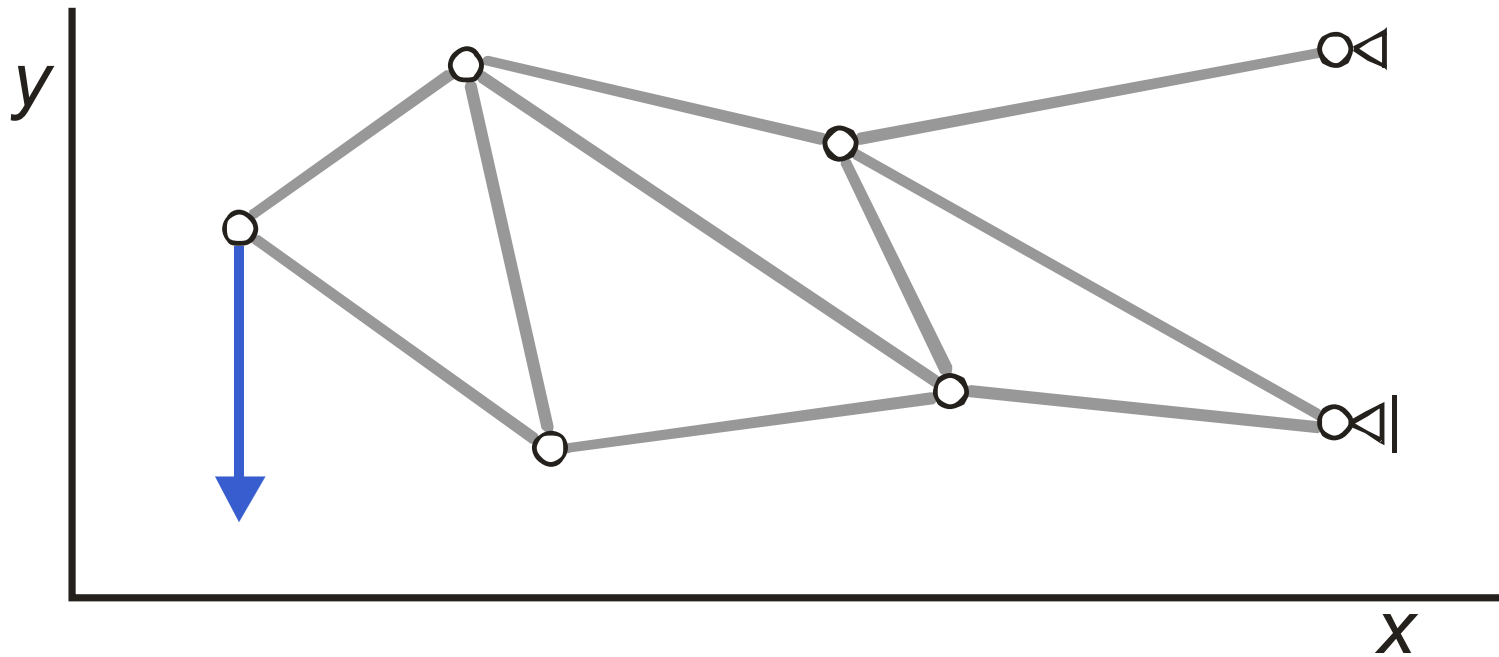
Weight $\Rightarrow$ Minimum
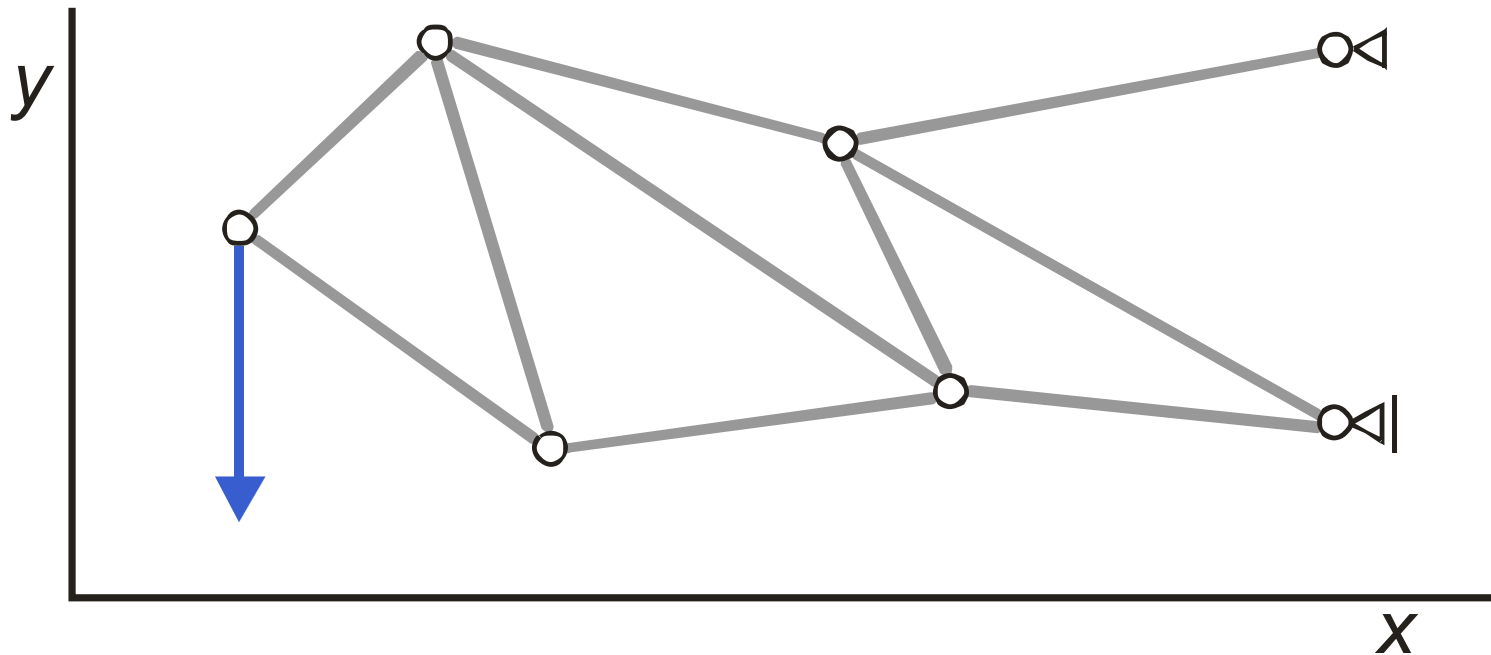
Weight $\Rightarrow$ Minimum
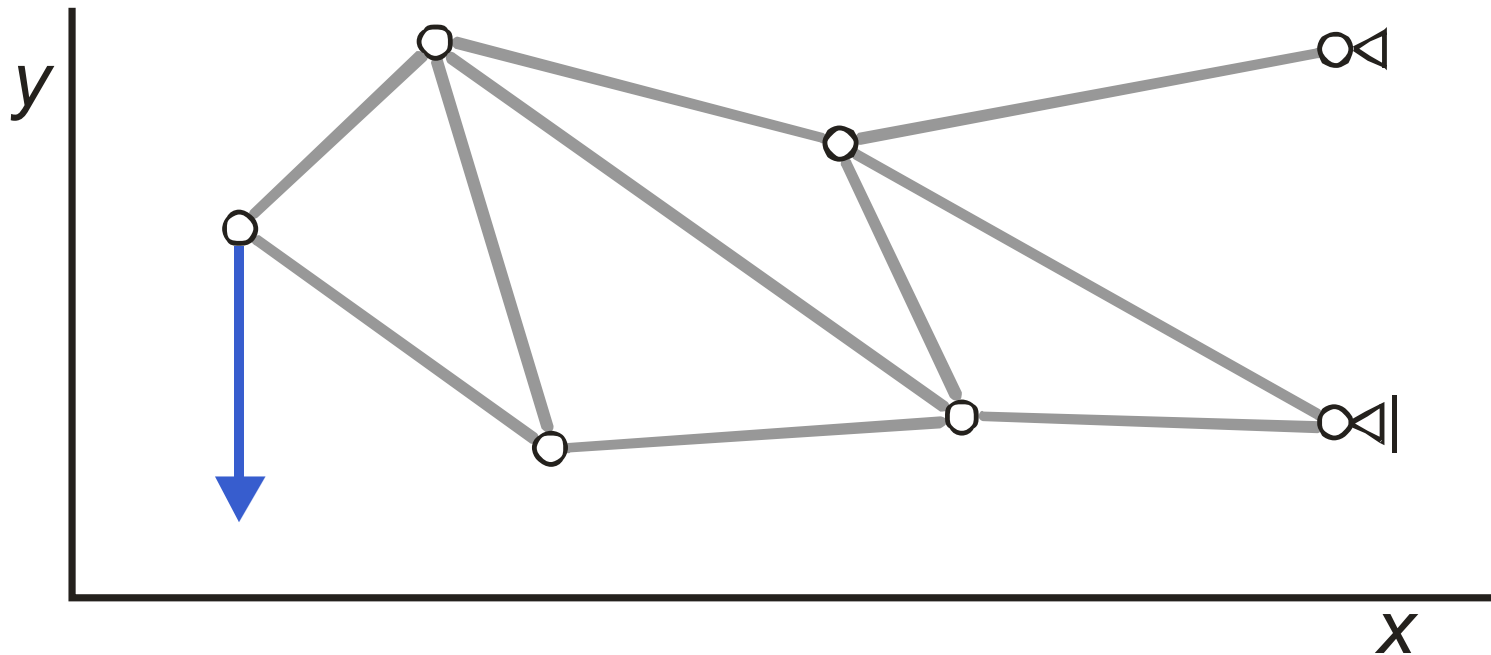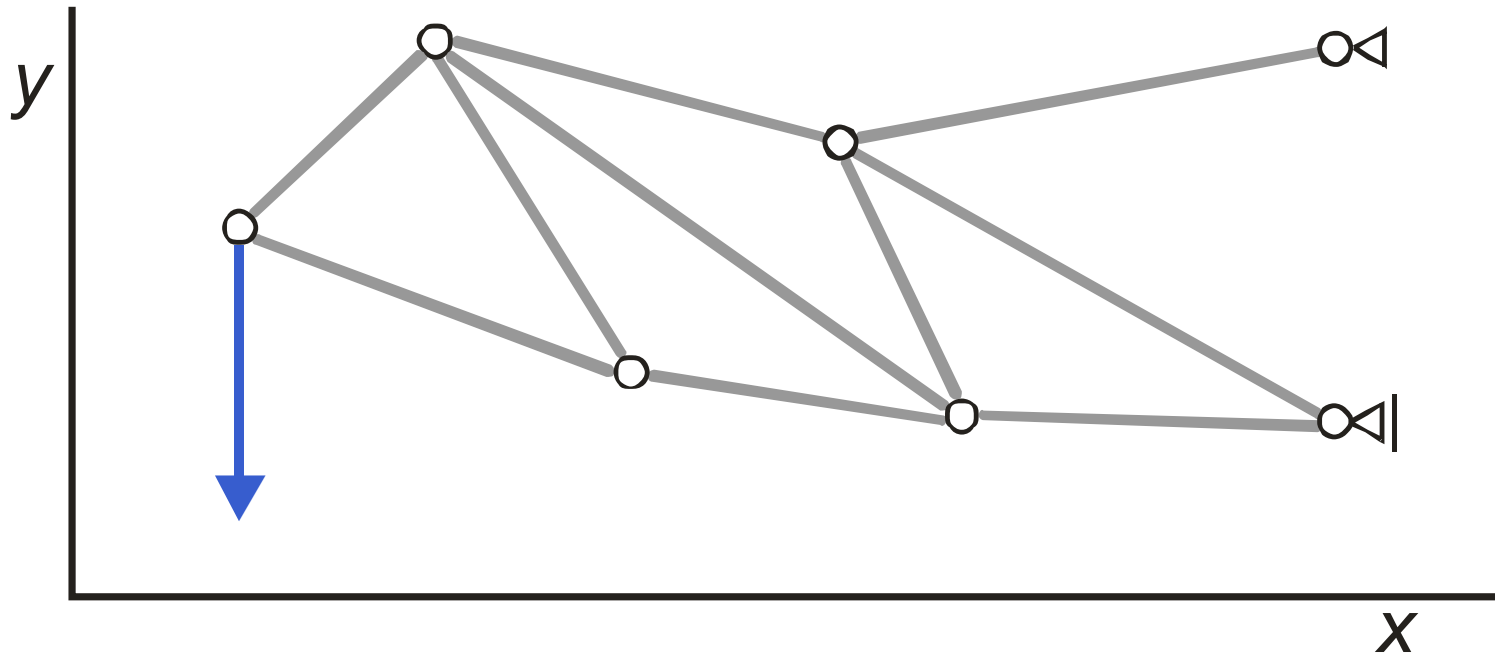
Weight $\Rightarrow$ Minimum

Weight $\Rightarrow$ Minimum

History of the Evolution Strategy

Multi-winglets at the
wing tip of a black vulture
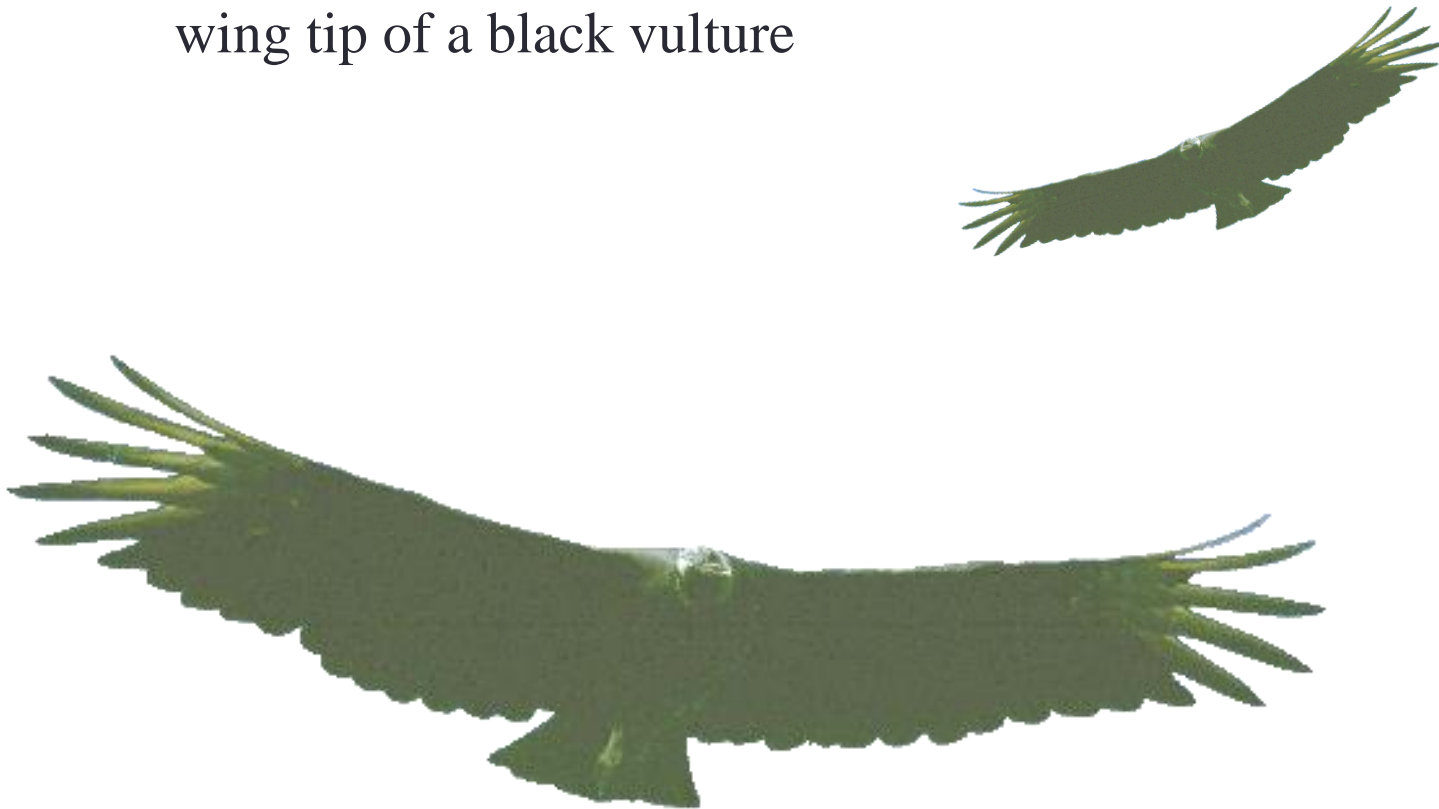
History of the Evolution Strategy

Evolution of a spread wing
in the windtunnel
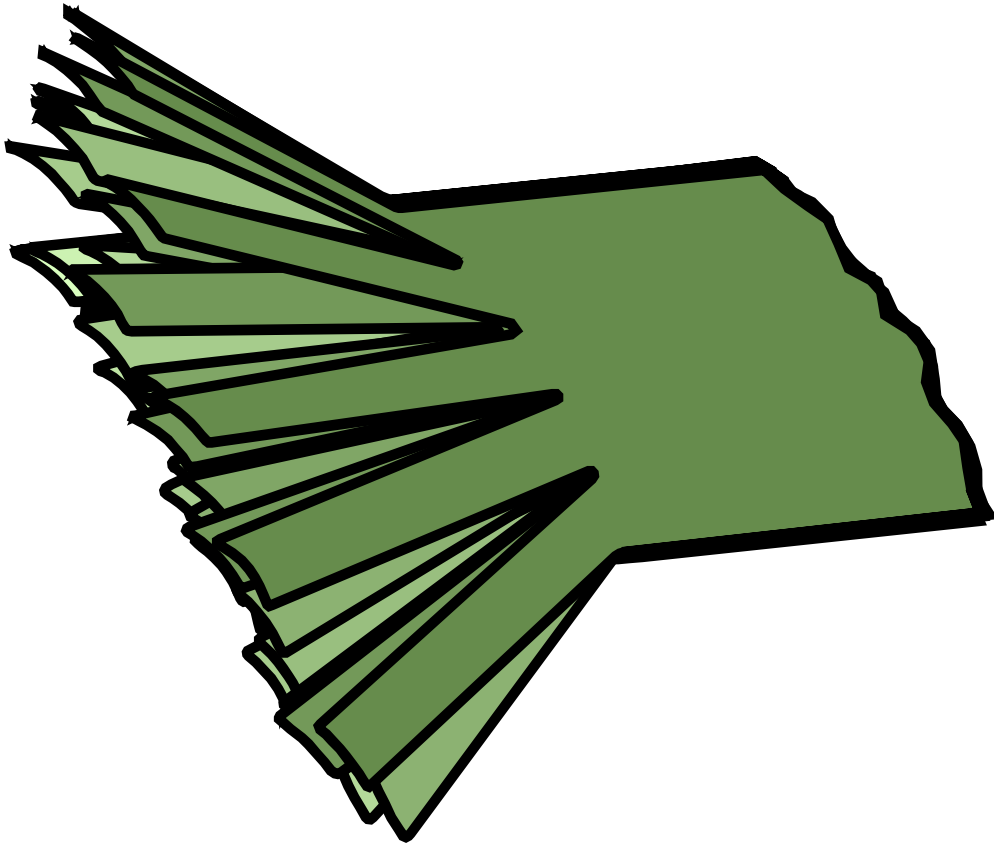
History of the Evolution Strategy

Photo: Michael Stache

Multiwinglets at a glider designed with the Evolution Strategy
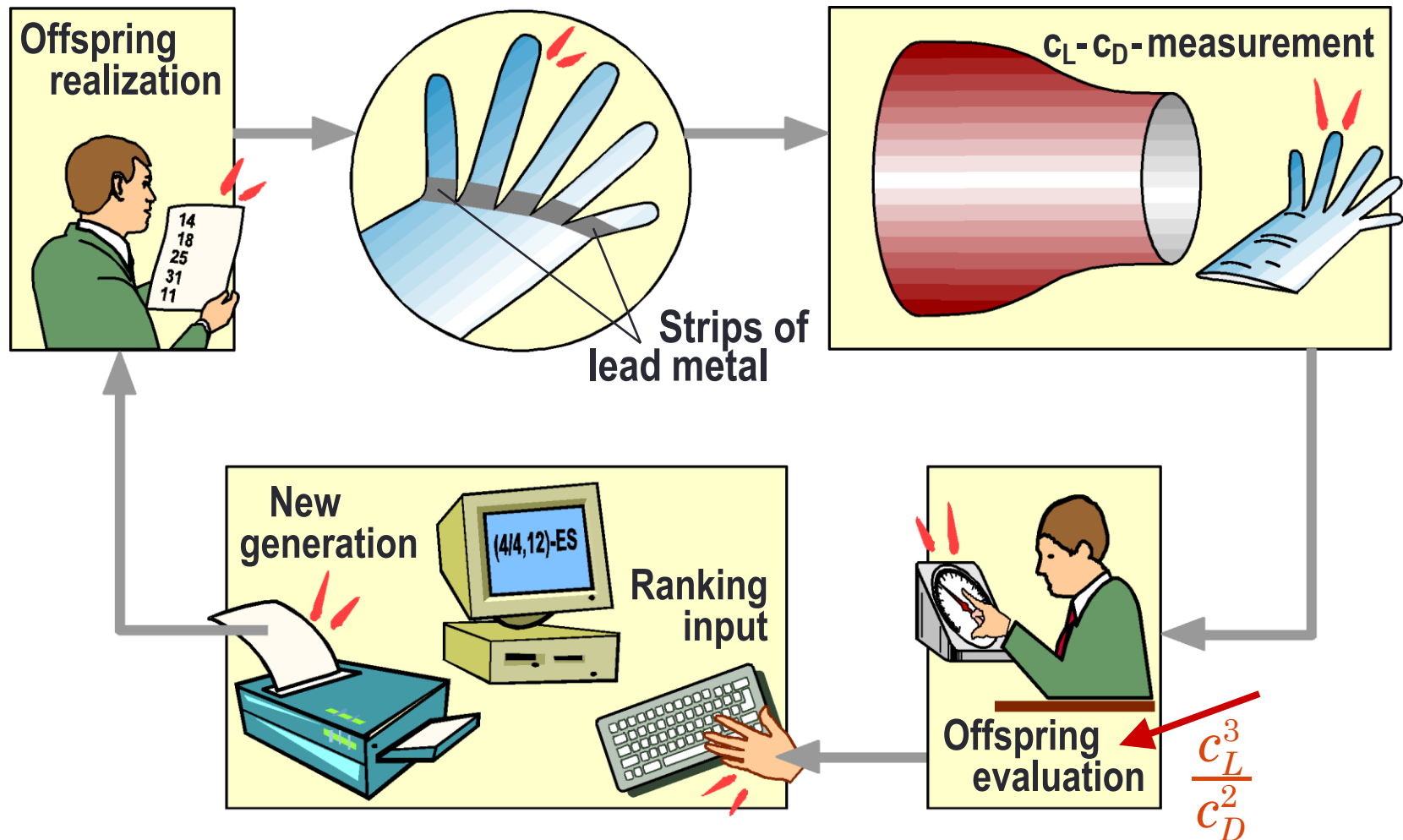
Post-Evolution-Experiment

The art of evolutionary experimentation in a wind tunnel

Generation
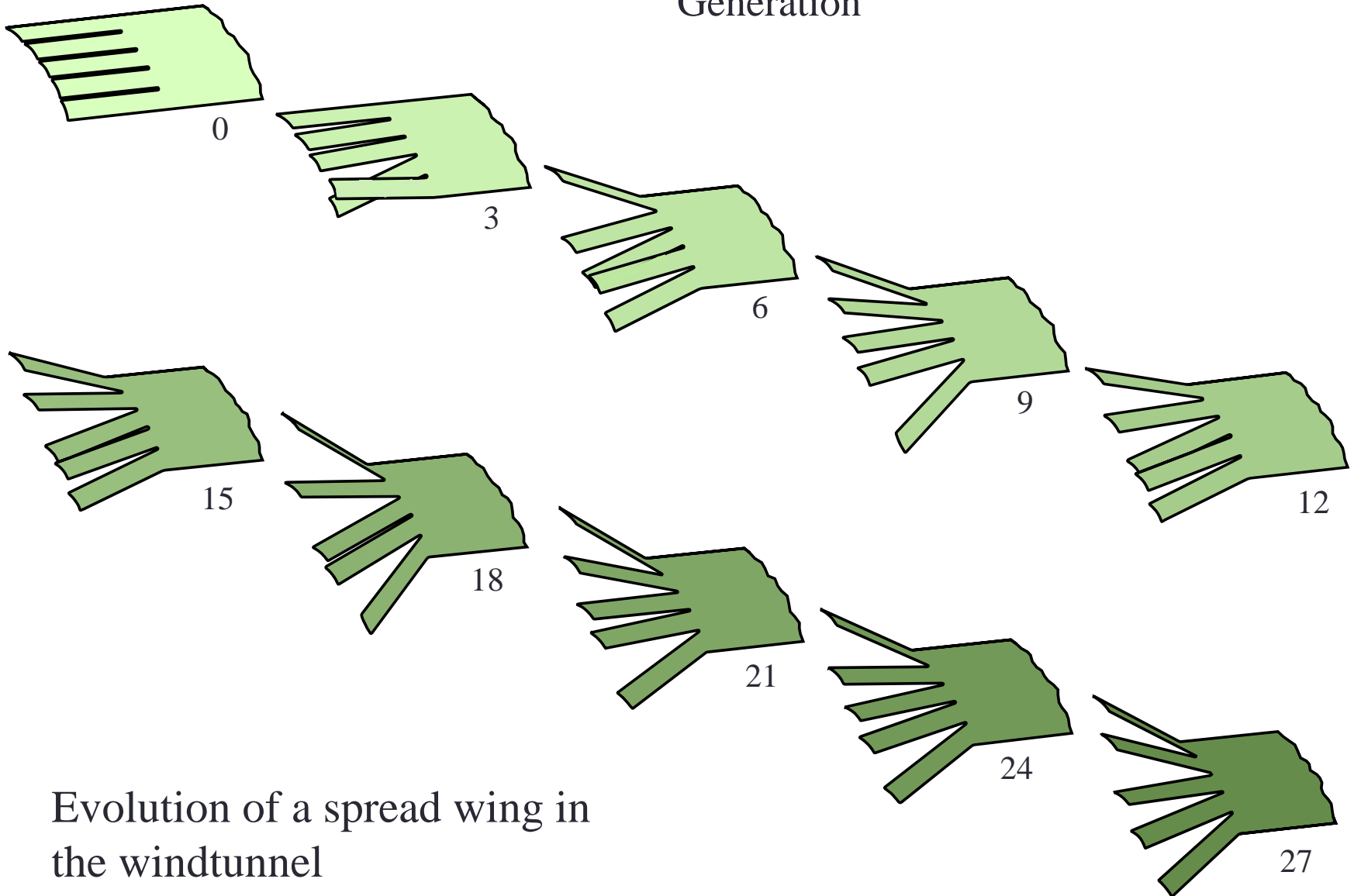
Evolution of a spread wing in
the windtunnel
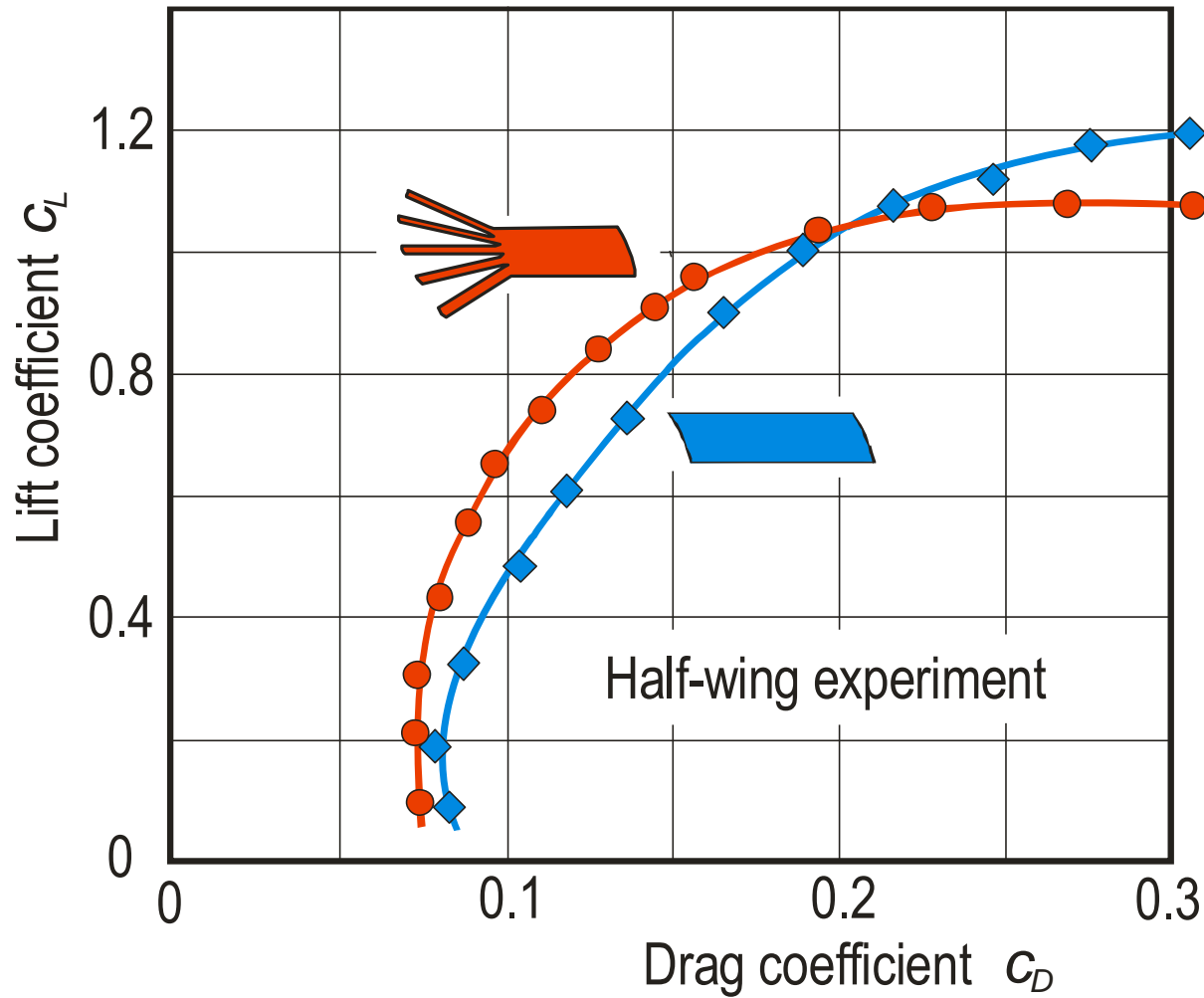
Spread wing versus normal wing

The difference between mathematical optimization and optimization in the real physical world
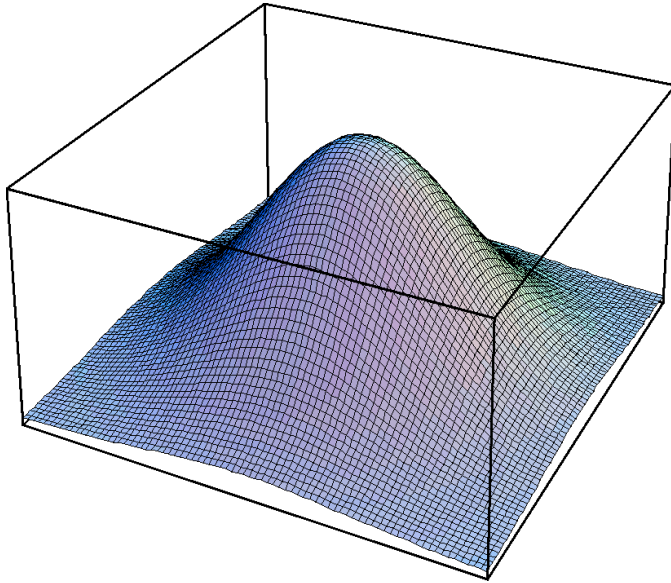
Ideal function in the mathematical world

Rugged hill in the experimental world
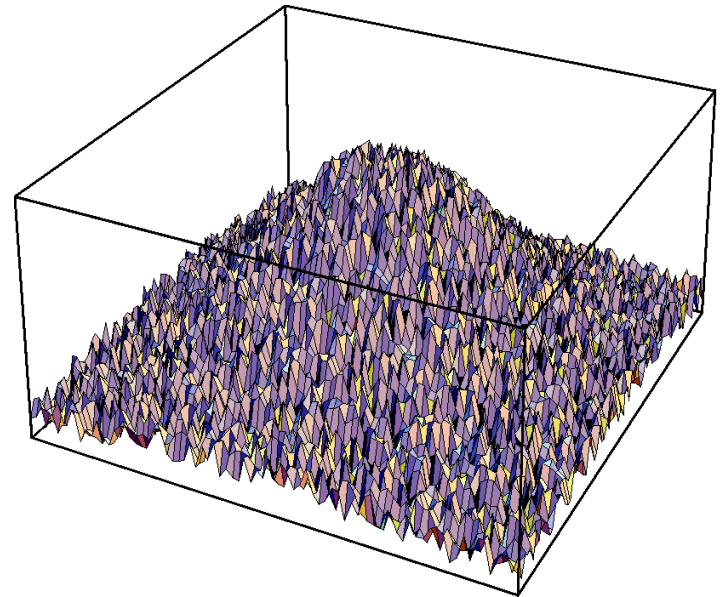
ES quick overview

In this chapter we introduce evolution strategies (ES), another member of the evolutionary algorithm family. We also use these algorithms to illustrate a very useful feature in evolutionary computing: self-adaptation of strategy parameters.

In general, self-adaptivity means that some parameters of the EA are varied during a run in a specific manner: the parameters are included in the chromosomes and coevolve with the solutions.

This feature is inherent in modern evolution strategies. That is, since the procedure was detailed in 1977 [340] most ESs have been self-adaptive, and over the last ten years other EAs have increasingly adopted self-adaptivity.

ES quick overview

- Developed: Germany in the 1970's
- Early names: I. Rechenberg, H.-P. Schwefel
- Typically applied to:
  - numerical optimisation
- Attributed features:
  - fast
  - good optimizer for real-valued optimisation
  - relatively much theory
- Special:
  - self-adaptation of (mutation) parameters standard

ES technical summary tableau

Representation	Real-valued vectors
Recombination	Discrete or intermediary
Mutation	Gaussian perturbation
Parent selection	Uniform random
Survivor selection	(μ, λ) or $(\mu + \lambda)$
Specialty	Self-adaptation of mutation step sizes

Introductory example

Here we describe the basic algorithm, termed the two-membered evolution strategy, for the abstract problem of minimizing an n-dimensional function $R^n \rightarrow R$.

- Task: minimalize $f : R^n \rightarrow R$

- Algorithm: "two-membered ES" using
  - Vectors from R^n directly as chromosomes
  - Population size 1
  - Only mutation creating one child
  - Greedy selection

Introductory example: pseudcode

```
BEGIN
  set t = 0;
  Create an initial point ⟨x₁ᵗ,...,xₙᵗ⟩ ∈ IRⁿ;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    draw zᵢ from a normal distr. for all i ∈ {1,...,n} independently;
    yᵢᵗ = xᵢᵗ + zᵢ for all i ∈ {1,...,n};
    IF (f(x̄ᵗ) ≤ f(ȳᵗ)) THEN
        x̄ᵗ⁺¹ = x̄ᵗ;
    ELSE
        x̄ᵗ⁺¹ = ȳᵗ;
    FI
    set t = t + 1;
  OD
END
```

$$\langle x_1^t,\ldots,x_n^t\rangle \in \mathbb{R}^n$$

$$y_i^t = x_i^t + z_i \quad \text{for all } i \in \{1,\ldots,n\}$$

$$\bar{x}^{t+1} = \bar{x}^t$$

$$\bar{x}^{t+1} = \bar{y}^t$$

$$X_{i,t+1} = X_{i,t} + N(0,\sigma)$$
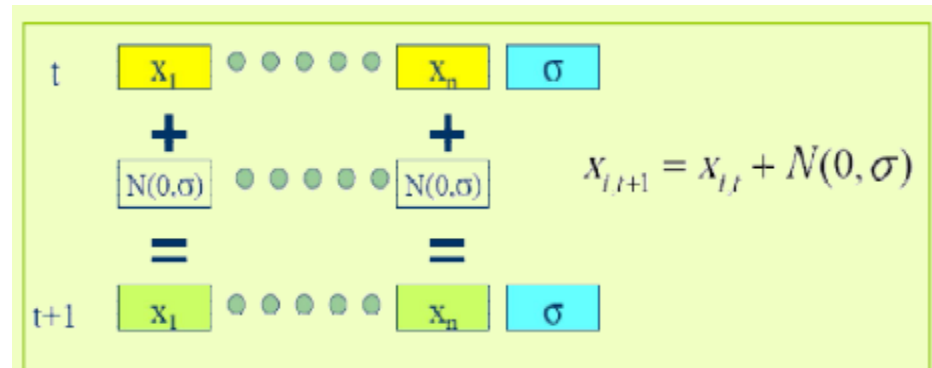
Fig. 4.1. Outline of simple two-membered evolution strategy

Concepts: Standard deviation, Normal distribution

- Consider $X = \langle x_1, x_2, \ldots, x_n \rangle$ *n*-dimensional random variable.

- The mean $M(X) = (x_1 + x_2 + \ldots + x_n)/n$.

- The **square of *standard deviation*** (also called ***variance***):

$$\sigma^2 = M(X - M(X))^2 = \Sigma(x_k - M(X))^2 / n$$

- **Normal distribution**:

$$N(M, \delta) = \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{(x-M)}{2\delta^2}}$$

The distribution with $M = 0$ and $\sigma^2 = 1$ is called the **standard normal**.

Illustration of normal distribution
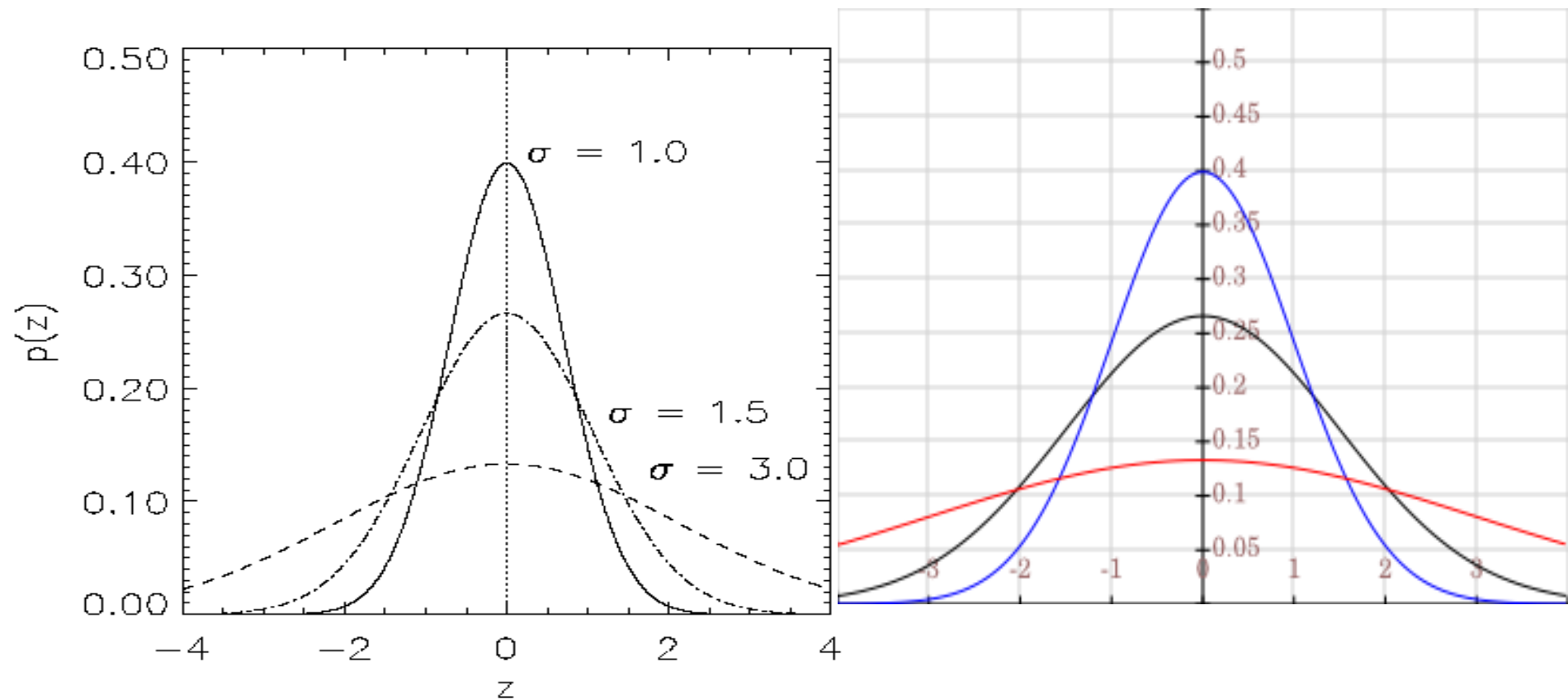
Introductory example: mutation mechanism

Given a current solution x^t in the form of a vector of length n, a new candidate x^{t+1} is created by adding a random number z_i for $i \in \{1,..., n\}$ to each of the n components.

A Gaussian, or normal, distribution is used with zero mean and standard deviation σ for drawing the random numbers.

This distribution is symmetric about zero and has the feature that the probability of drawing a random number with any given magnitude is a rapidly decreasing function of the standard deviation σ.

Thus the σ value is a parameter of the algorithm that determines the extent to which given values x_i are perturbed by the mutation operator.

Introductory example: mutation mechanism

For this reason σ is often called the **mutation step size**. Theoretical studies motivated an on-line adjustment of step sizes by the famous **1/5 success rule of Rechenberg**.

This rule states that the ratio of successful mutations (those in which the child is fitter than the parent) to all mutations should be 1/5.

Hence if the ratio is greater than 1/5 the step size should be increased to make a wider search of the space, and if the ratio is less than 1/5 then it should be decreased to concentrate the search more around the current solution.

The rule is executed at periodic intervals, for instance, after k iterations each σ is reset by

- $\sigma = \sigma / c$ if $p_s > 1/5$ (Foot of big hill ➔ increase σ)
- $\sigma = \sigma \cdot c$ if $p_s < 1/5$ (Near the top of the hill ➔ decrease σ)
- $\sigma = \sigma$ if $p_s = 1/5$

where p_s is the relative frequency of successful mutations measured over a number of trials, and the parameter c is in the range $0.817 < c < 1$. As is apparent, using this mechanism the step sizes change based on feedback from the search process.

Introductory example: mutation mechanism-summary

- z values drawn from normal distribution $N(\xi,\sigma)$
  - mean ξ is set to 0
  - variation σ is called mutation step size
- σ is varied on the fly by the "1/5 success rule":
- This rule resets σ after every k iterations by
  - $\sigma = \sigma / c$ if $p_s > 1/5$ (Foot of big hill ➔ increase σ)
  - $\sigma = \sigma \cdot c$ if $p_s < 1/5$ (Near the top of the hill ➔ decrease σ)
  - $\sigma = \sigma$ if $p_s = 1/5$
- where p_s is the % of successful mutations, $0.8 \le c < 1$

Introductory example: characteristics

This example illuminates some essential characteristics of evolution strategies:

1. Evolution strategies are typically used for continuous parameter optimization.

2. There is a strong emphasis on mutation for creating offspring.

3. Mutation is implemented by adding some random noise drawn from a Gaussian distribution.

4. Mutation parameters are changed during a run of the algorithm.

Another historical example: the jet nozzle experiment

Task: to optimize the shape of a jet nozzle
Approach: random mutations to shape + selection
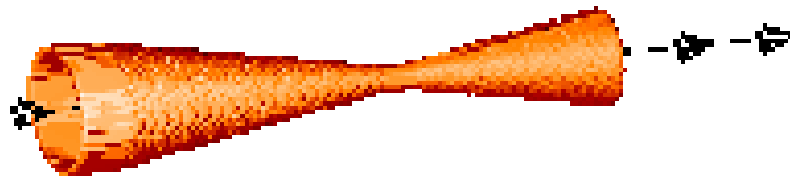
Initial shape
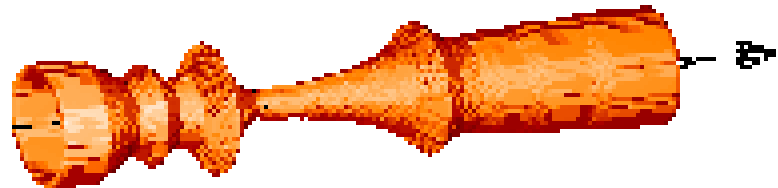
Final shape

Representation

- Nowadays evolution strategies almost always use self-adaptation **(rather than the 1/5 success rule adaptation)**, the vector $x = \langle x_1,\ldots,x_n \rangle$ forms only part of a typical ES genotype.

- Individuals contain some strategy parameters, in particular, parameters of the mutation operator. Details of mutation are treated in the next section; here we only discuss the structure of individuals, and specify the meaning of the special genes there.

- Strategy parameters can be divided into two sets, the σ values and the α values. The σ values represent the mutation step sizes, and their number n_σ is usually either *1* or *n*.

- For any reasonable self-adaptation mechanism at least one σ must be present. The α values, which represent interactions between the step sizes used for different variables, are not always used. In the most general case their number $n_\alpha = (n - n_\alpha/2)(n_\alpha - 1)$.

- Putting this all together, we obtain $\langle x_1,\ldots,x_n, \sigma_1,\ldots,\sigma_n, \alpha_1,\ldots, \alpha_k \rangle$ as the general form of individuals in ES.

Representation-summary

- Chromosomes consist of three parts:
  - Object variables: $x_1,\ldots,x_n$
  - Strategy parameters:
    - Mutation step sizes: $\sigma_1,\ldots,\sigma_{n_\sigma}$
    - Rotation angles: $\alpha_1,\ldots, \alpha_{n_\alpha}$

- Not every component is always present

- Full size: $\langle\, x_1,\ldots,x_n, \sigma_1,\ldots,\sigma_n, \alpha_1,\ldots, \alpha_k \,\rangle$

- where k = n(n-1)/2 (no. of i,j pairs)

65

Mutation Operators

n_σ	n_α	Structure of individuals	Remark
1	0	$\langle x_1, \ldots, x_n . \sigma \rangle$	Standard mutation
n	0	$\langle x_1, \ldots, x_n . \sigma_1, \ldots, \sigma_n \rangle$	Standard mutations
n	$n \cdot (n-1)/2$	$\langle x_1, \ldots, x_n . \sigma_1, \ldots, \sigma_n, \alpha_1, \ldots, \alpha_{n \cdot (n-1)/2} \rangle$	Correlated mutations

Table 4.2. Some possible settings of n_σ and n_α for different mutation operators

Mutation

- Main mechanism: changing value by adding random noise drawn from normal distribution
- $x'_i = x_i + N(0,\sigma)$
- Key idea:
  - σ is part of the chromosome $\langle x_1,\dots,x_n, \sigma \rangle$
  - σ is also mutated into σ' (see later how)
- Thus: mutation step size σ is coevolving with the solution x

Mutate σ first

- Net mutation effect: $\langle$ x, σ $\rangle$ $\rightarrow$ $\langle$ x', σ' $\rangle$
- Order is important:
  - first σ $\rightarrow$ σ' (see later how)
  - then x $\rightarrow$ x' = x + N(0,σ')
- Rationale: new $\langle$ x' ,σ' $\rangle$ is evaluated twice
  - Primary: x' is good if f(x') is good
  - Secondary: σ' is good if the x' it created is good
- Reversing mutation order this would not work

Mutation case 1: Uncorrelated mutation with one σ

- Chromosomes: $\langle x_1, \ldots, x_n, \sigma \rangle$
- $\sigma' = \sigma \cdot \exp(\tau \cdot N(0,1))$
- $x'_i = x_i + \sigma' \cdot N(0,1)$
- Typically the "learning rate" $\tau \propto 1/n^{\frac{1}{2}}$
- And we have a boundary rule $\sigma' < \varepsilon_0 \Rightarrow \sigma' = \varepsilon_0$

Mutants with equal likelihood

Fig. 4.2. Mutation with $n = 2$, $n_\delta = 1$, $n_\alpha = 0$.

Part of a fitness landscape with a conical shape is shown. The **black dot** indicates an individual. Points where the offspring can be placed with a given probability form a circle. The probability of moving along the y-axis (little effect on fitness) is the same as that of moving along the x-axis (large effect on fitness)

Mutation case 2: Uncorrelated mutation with n σ's

- Chromosomes: $\langle x_1,\ldots,x_n, \sigma_1,\ldots, \sigma_n \rangle$
- $\sigma'_i = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1))$
- $x'_i = x_i + \sigma'_i \cdot N_i(0,1)$
- Two learning rate parameters:
  - τ' overall learning rate
  - τ coordinate wise learning rate
- $\tau \propto 1/(2\,n)^{\frac{1}{2}}$ and $\tau \propto 1/(2\,n^{\frac{1}{2}})^{\frac{1}{2}}$
- And $\sigma_i' < \varepsilon_0 \Rightarrow \sigma_i' = \varepsilon_0$

Mutants with equal likelihood

Local maximum
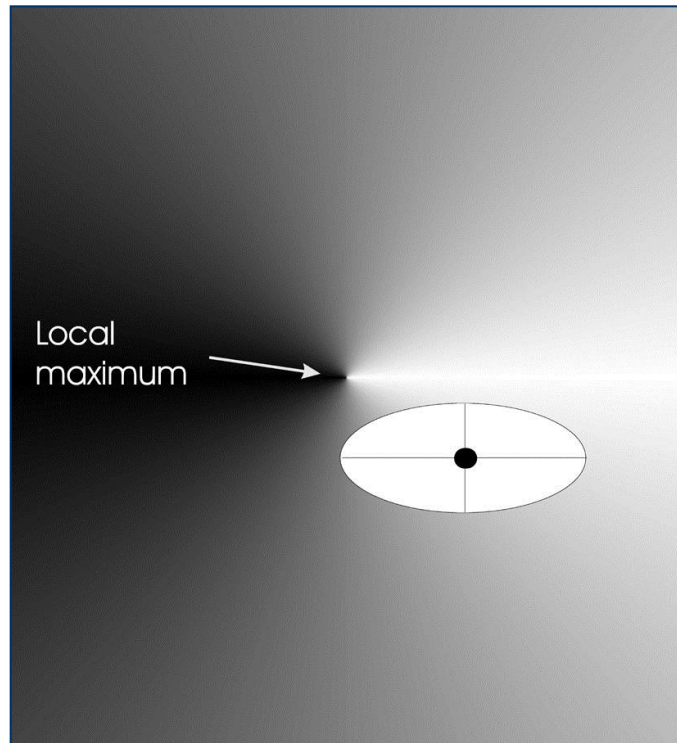
Fig. 4.3. Mutation with $n = 2$, $n_\delta = 2$, $n_\alpha = 0$.

Part of a fitness landscape with a conical shape is shown. The black dot indicates an individual. Points where the offspring can be placed with a given probability form an ellipse.

The probability of moving along the x-axis (large effect on fitness) is larger than that of moving along the y-axis (little effect on fitness).

Mutation case 3: Correlated mutations

- Chromosomes: $\langle x_1,\ldots,x_n, \sigma_1,\ldots, \sigma_n ,\alpha_1,\ldots, \alpha_k \rangle$
- where $k = n \cdot (n-1)/2$
- and the covariance matrix C is defined as:
  - $c_{ii} = \sigma_i^2$
  - $c_{ij} = 0$ if i and j are not correlated
  - $c_{ij} = \frac{1}{2} \cdot (\sigma_i^2 - \sigma_j^2) \cdot \tan(2 \alpha_{ij})$ if i and j are correlated
- Note the numbering / indices of the α's

Correlated mutations cont'd

The mutation mechanism is then:

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

- $\alpha'_j = \alpha_j + \beta \cdot N(0,1)$

- **$x' = x + N(0,C')$**
  - x stands for the vector $\langle x_1,\ldots,x_n \rangle$
  - C' is the covariance matrix C after mutation of the α values

- $\tau \propto 1/(2 n)^{\frac{1}{2}}$ and $\tau \propto 1/(2 n^{\frac{1}{2}})^{\frac{1}{2}}$ and $\beta \approx 5°$

- $\sigma_i' < \varepsilon_0 \Rightarrow \sigma_i' = \varepsilon_0$ and

- $| \alpha'_j | > \pi \Rightarrow \alpha'_j = \alpha'_j - 2\pi \, \text{sign}(\alpha'_j)$

Mutants with equal likelihood

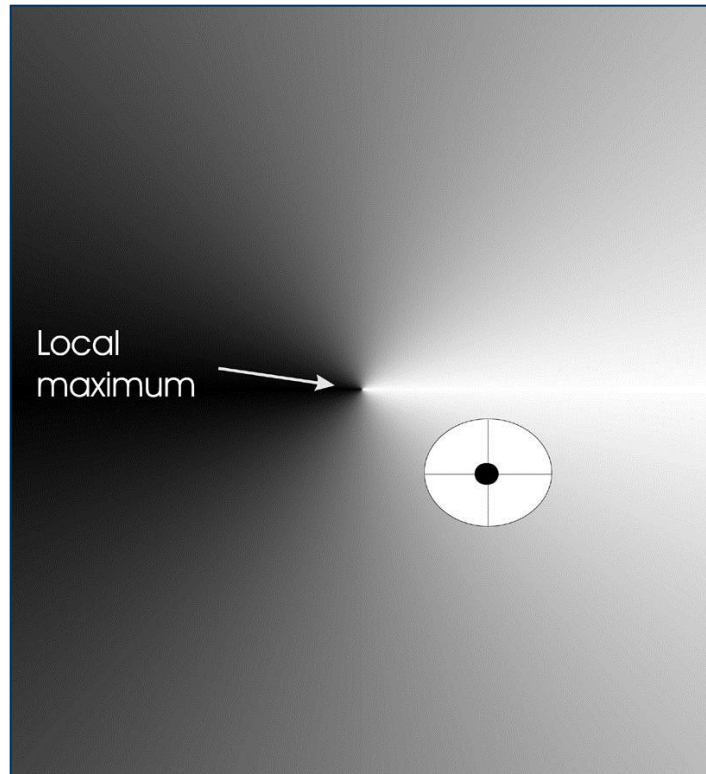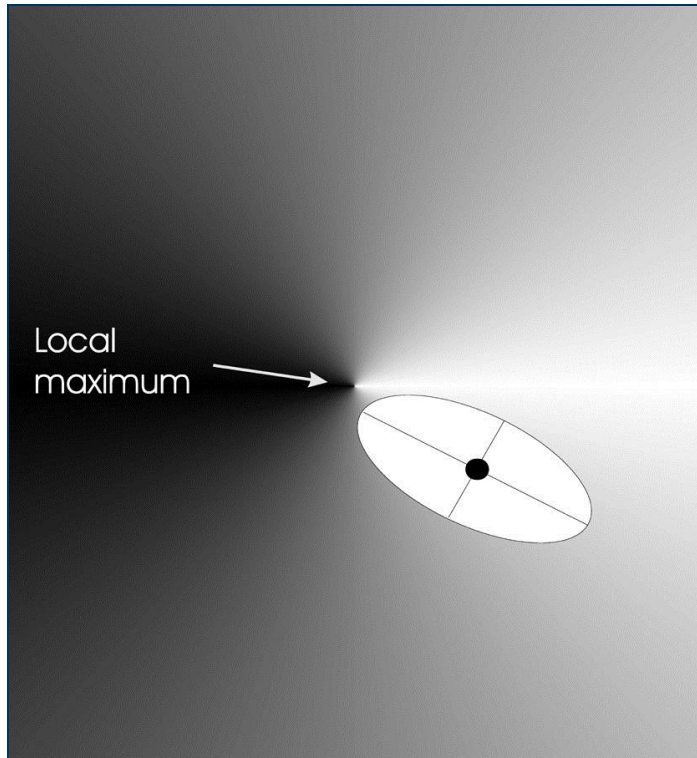Fig. 4.4. Correlated mutation: $n = 2$, $n_\delta = 2$, $n_\alpha = 1$.

Part of a fitness landscape with a conical shape is shown. The black dot indicates an individual. Points where the offspring can be placed with a given probability form a rotated ellipse. The probability of generating a move in the direction of the steepest ascent (largest effect on fitness) is now larger than that for other directions.

75

Recombination methods Discrete/Intermediary vs. Global/Local

	Two fixed parents	Two parents selected for each i
$z_i = (x_i + y_i)/2$	Local intermediary	Global intermediary
z_i is x_i or y_i chosen randomly	Local discrete	Global discrete

Recombination

- The basic recombination scheme in evolution strategies involves two parents that create one child.

- To obtain λ offspring recombination is performed λ times.

- There are two recombination variants distinguished by the manner of recombining parent alleles. Using **discrete recombination** one of the parent alleles is randomly chosen with equal chance for either parents. In **intermediate recombination** the values of the parent alleles are averaged.

Recombination

An extension of this scheme allows the use of **more than two recombinants**, because the two parents x and y are drawn randomly for each position $i \in \{1.....n\}$ in the offspring anew.

These drawings take the whole population of μ individuals into consideration, and the result is a recombination operator with possibly more than two individuals contributing to the offspring.

The exact number of parents, however, cannot be defined in advance. This multi-parent variant is called **global recombination**. To make terminology unambiguous, the original variant is called **local recombination**.

Evolution strategies typically use global recombination. Interestingly, different **recombination is used for the object variable part (discrete is recommended)** and **the strategy parameters part (intermediary is recommended)**.

This scheme preserves diversity within the phenotype (solution) space, allowing the trial of very different combinations of values, whilst the averaging effect of intermediate recombination assures a more cautious adaptation of strategy parameters.

Parent selection

- Parent selection in evolution strategies **is not biased by fitness values**. Whenever a recombination operator requires a parent, it is drawn **randomly with uniform distribution from the population of** μ **individuals**.

- It should be noted that the ES terminology deviates from the GA terminology in its use of the word "parent". **In ES the whole population is seen as parent - often called the parent population**.

- In contrast, in GAs the term "parent'" denotes **a member of the population that has actually been selected to undergo variation (crossover or mutation)**.

Survivor selection

- After creating λ offspring and calculating their fitness, the best μ of them are chosen deterministically, either from the offspring only, called (μ,λ) selection, or from the union of parents and offspring, called $(\mu+\lambda)$ selection.

- Both the (μ,λ) and the $(\mu+\lambda)$ selection schemes are strictly deterministic and **are based on rank rather than an absolute fitness value**.

Survivor selection

The selection scheme that is generally used in evolution strategies is (μ,λ) selection, which is preferred over $(\mu+\lambda)$ selection for the following reasons:

- The (μ,λ) discards all parents and is therefore in principle able to leave (small) local optima, so it is advantageous in the case of multi-modal topologies.

- If the fitness function is not fixed, but changes in time, the $(\mu+\lambda)$ selection preserves outdated solutions, so it is not able to follow the moving optimum well.

- $(\mu+\lambda)$ selection hinders the self-adaptation mechanism with respect to strategy parameters to work effectively, because maladapted strategy parameters may survive for a relatively large number of generations when an individual has relatively good object variables and bad strategy parameters. In that case often all its children will be bad, so with elitism, the bad strategy parameters may survive.

Survivor selection

- The selective pressure in evolution strategies is very high because A is typically much higher than μ (a 1/7 ratio is recommended).

- The takeover time τ^* of a given selection mechanism is defined as **the number of generations it takes until the application of selection completely fills the population with copies of the best individual, given one copy initially**. Goldberg showed that $\tau^* = {}^{\ln \lambda}/_{\ln(\lambda/\mu)}$

- For a typical evolution strategy with $\mu = 15$ and $\lambda = 100$ this results in $\tau^* \approx 2$.

- For proportional selection in a genetic algorithm it is $\tau^* = \lambda \ln \lambda$, resulting in $\tau^* = 460$ for population size $\lambda = 100$.

Survivor selection-summary

- Applied after creating λ children from the μ parents by mutation and recombination
- Deterministically chops off the "bad stuff"
- Basis of selection is either:
  - The set of children only: (μ,λ)-selection
  - The set of parents and children: $(\mu+\lambda)$-selection

Survivor selection-summary

- $(\mu+\lambda)$-selection is an elitist strategy
- (μ,λ)-selection can "forget"
- Often (μ,λ)-selection is preferred for:
    - Better in leaving local optima
    - Better in following moving optima
    - Using the + strategy bad σ values can survive in $\langle x,\sigma \rangle$ too long if their host x is very fit
- Selective pressure in ES is very high ($\lambda \approx 7 \cdot \mu$ is the common setting). Because λ is typically much higher than μ (a 1/7 ratio is recommended).

$(\mathbf{1}+1)$-ES
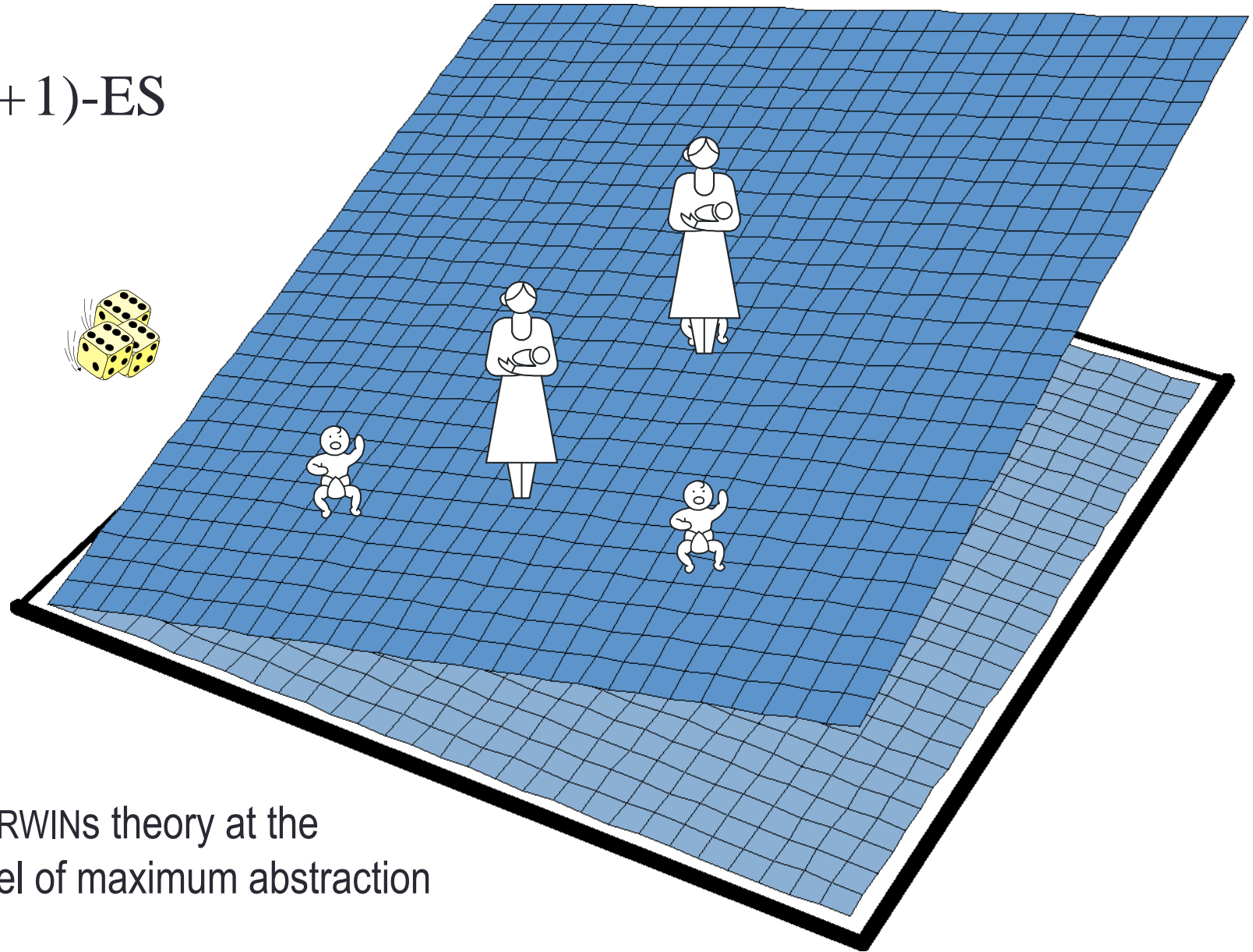
DARWINs theory at the
level of maximum abstraction

$(\mathbf{1}, \lambda)$-ES

$\lambda = 6$
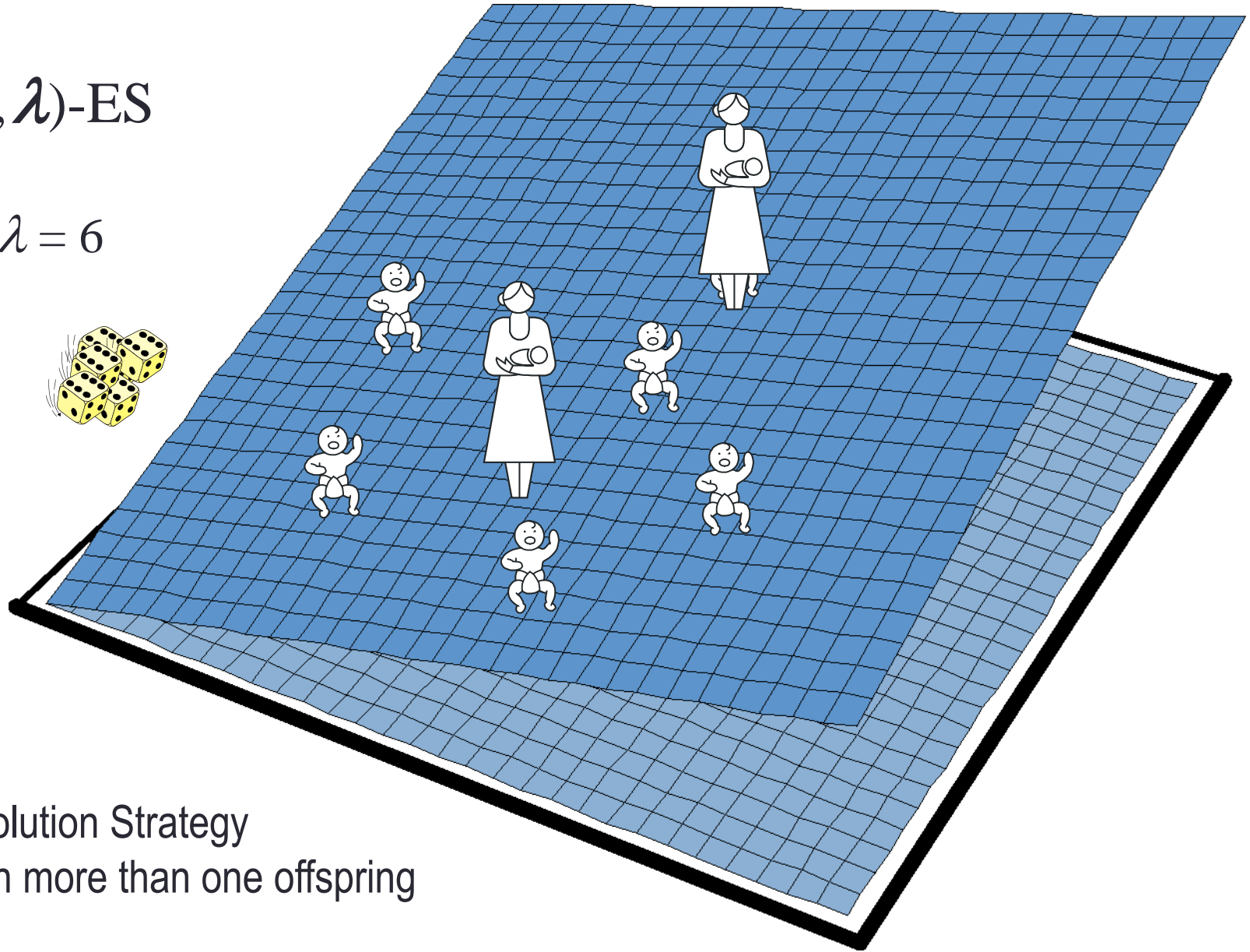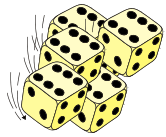
Evolution Strategy
with more than one offspring

(μ, λ)-ES

$\mu = 2$
$\lambda = 7$
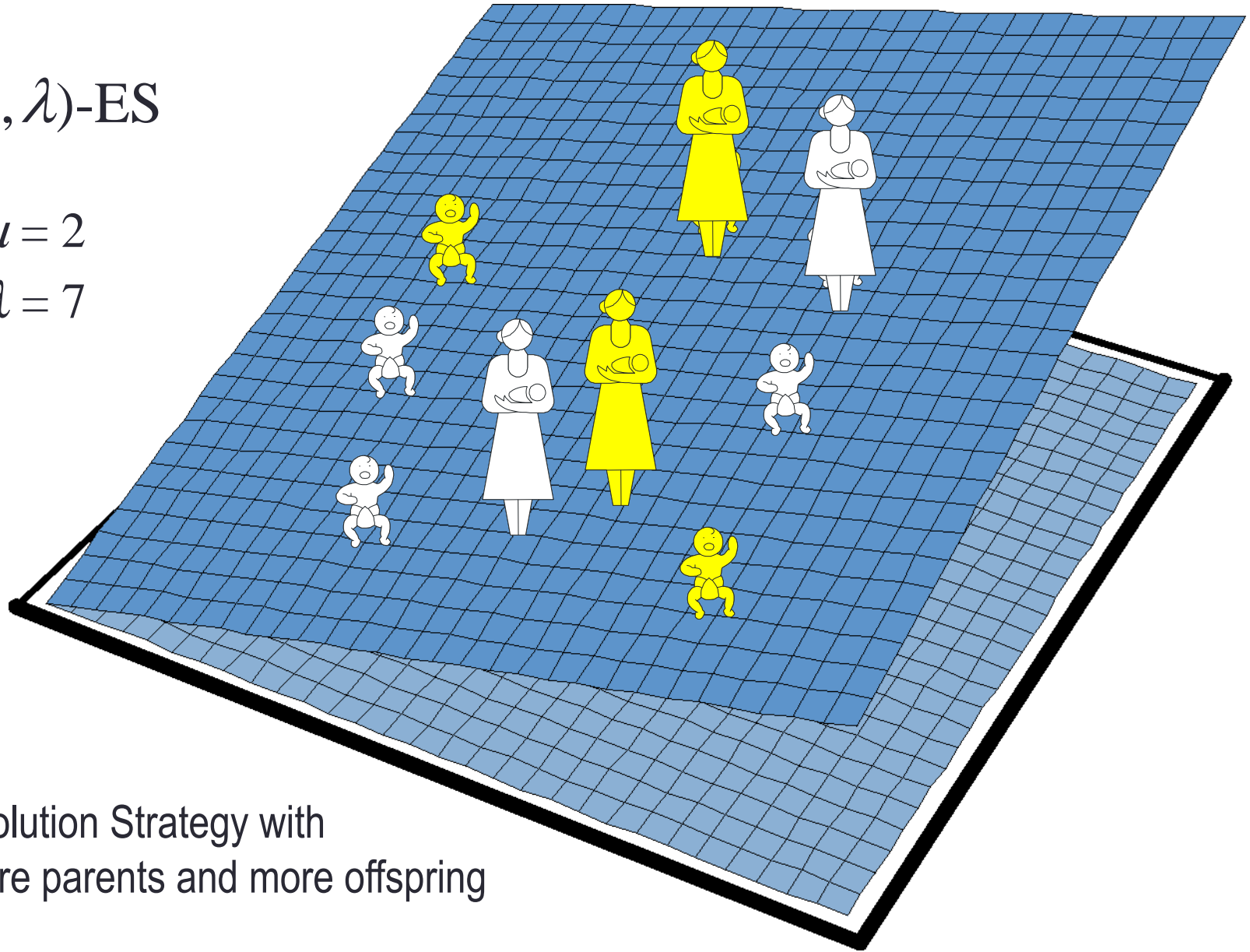
Evolution Strategy with
more parents and more offspring

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

$\mu = 2$
$\rho = 2$
$\lambda = 8$
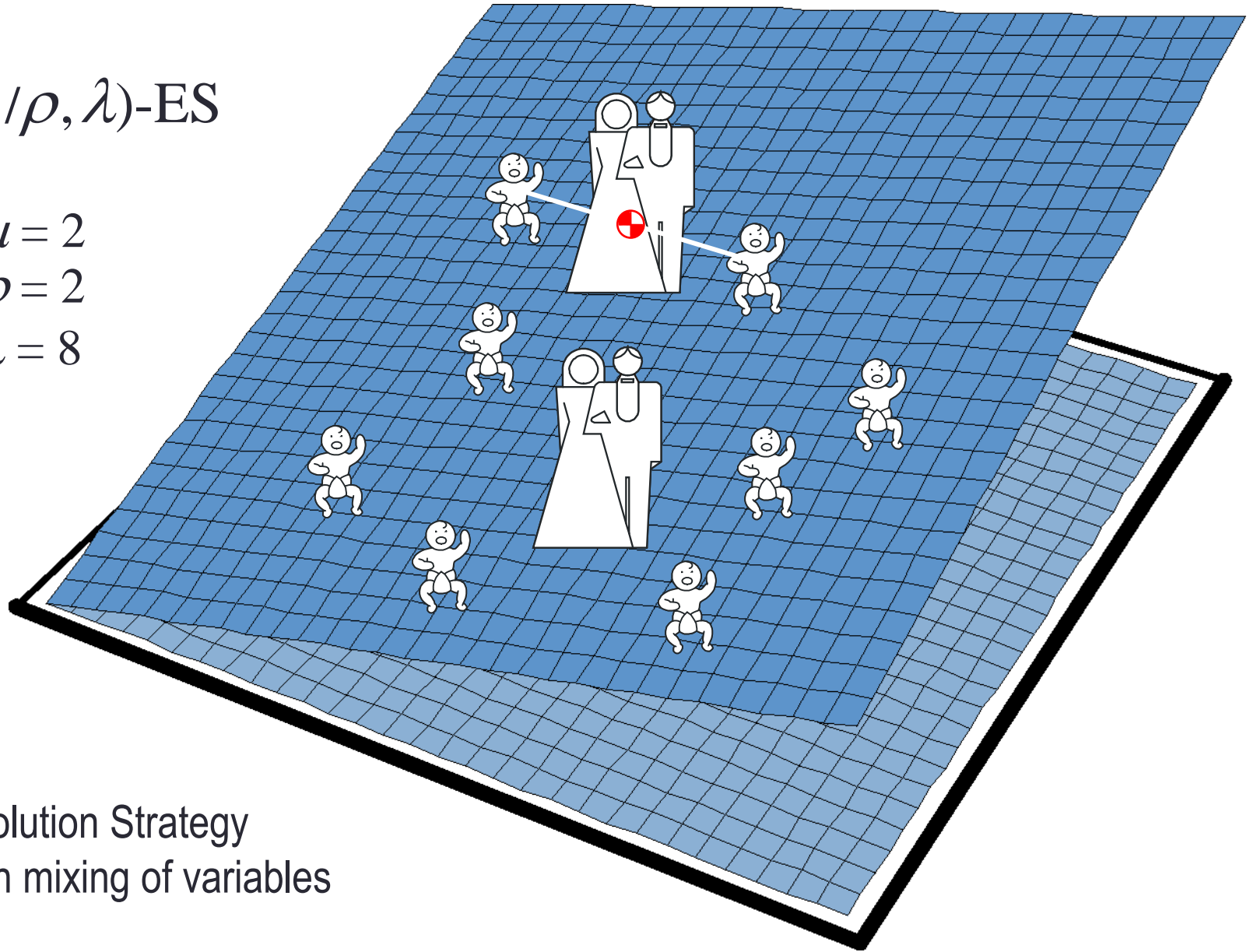
Evolution Strategy
with mixing of variables

$$[\mu', \lambda'(\mu, \lambda)^\gamma] - \mathrm{ES}$$

$$\mu' = 1$$
$$\lambda' = 2$$
$$\mu = 1$$
$$\lambda = 5$$
$$\gamma = 4$$

New founder populations
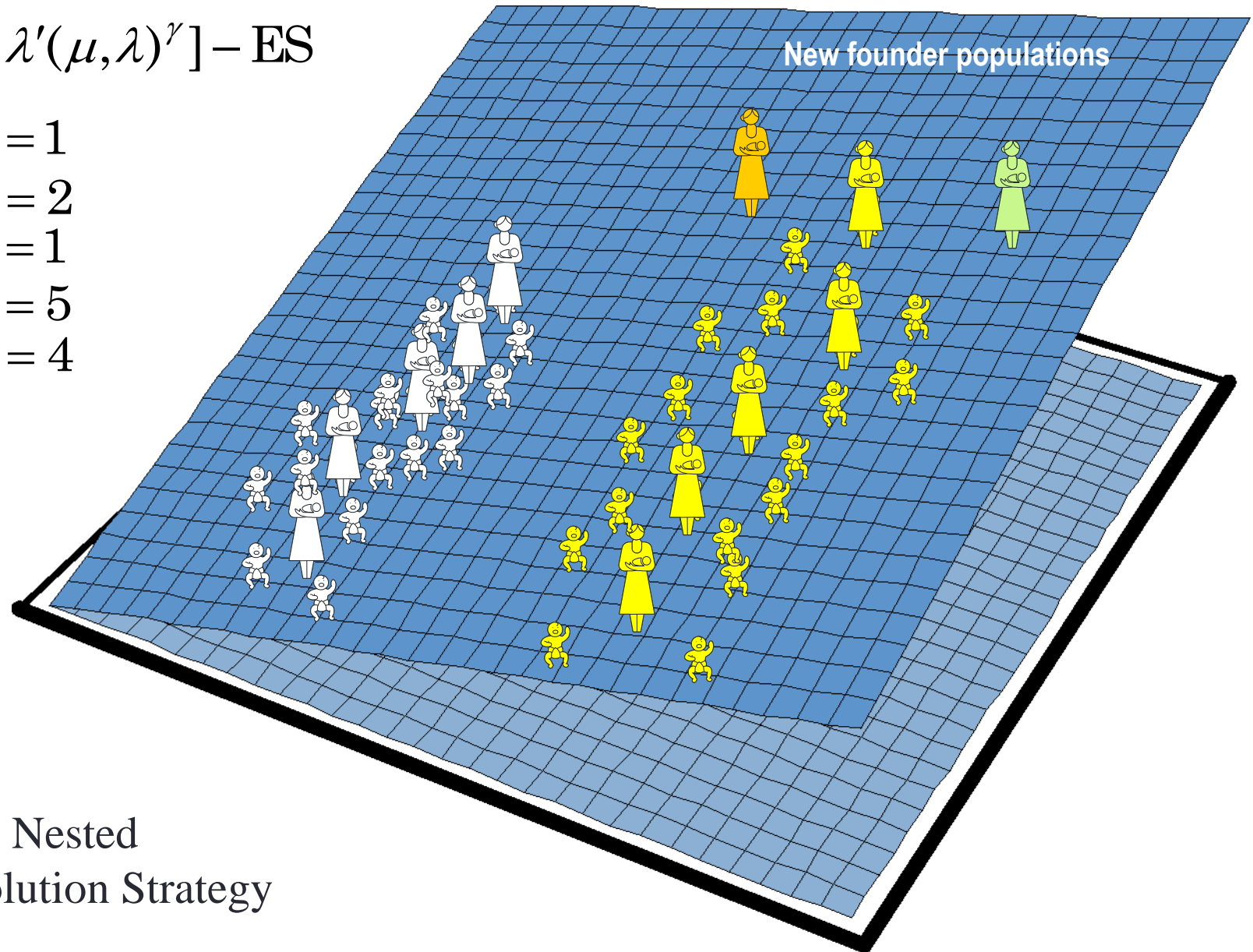
The Nested
Evolution Strategy

Self-adaptation

- One of the main contributions of evolution strategies to the field of EC is **self- adaptation**.

- This feature was introduced first in ES, and its benefits have been clearly shown by ES research, **not only for real-valued, but also for binary and integer search spaces**.

- This has had an inspiring effect on other EC branches that started incorporate self-adaptive features in other style EAs. Besides, experimental evidences showing that an **ES with self-adaptation outperforms the same ES without self-adaptation**, there are also theoretical results backing up this claim.

- we can conclude that self-adaptation works in the sense that it is able to find the near-optimal step sizes.

Self-adaptation

Theoretical and experimental results agree on the fact that for a successful run the δ values must decrease over time.

The intuitive explanation for this is that **in the beginning of a search process a large part of the search space has to be sampled in an explorative fashion to locate promising regions (with good fitness values)**. Therefore, large mutations are appropriate in this phase.

As the search proceeds and optimal values are approached, only fine tuning of the given individuals is needed; thus smaller mutations are required.

Self-adaptation

Another kind of convincing evidence for the power of self-adaptation is provided in the context of **changing fitness** landscapes.

In this case, where the **objective function is changing**, the evolutionary process is aiming at a moving target. When the objective function changes, the present population needs to be reevaluated, and quite naturally the given individuals may have a low fitness, since they have been adapted to the old objective function.

Often the mutation step sizes will prove ill-adapted: they are too low for the new exploration phase required.

Self-adaptation illustrated

- The experiment presented in [201] illustrates how self-adaptation is able to reset the step sizes after each change in the objective function (Fig. 4.5).

- Moving optimum ES experiment on the sphere function with $n = 30$, $n_{\sigma} = 1$.

- The location of the optimum is changed after every 200 generations (x-axes) with a clear effect on the average best objective function values (y-axis, left) in the given population.

- Self-adaptation is adjusting the step sizes (y-axes, right) with a small delay to larger values appropriate for exploring the new fitness landscape, whereafter the values of a start decreasing again once the population is closing in on the new optimum.

- Self-adaptive ES is able to
  - follow the optimum and
  - adjust the mutation step size after every shift !

Self-adaptation illustrated cont'd
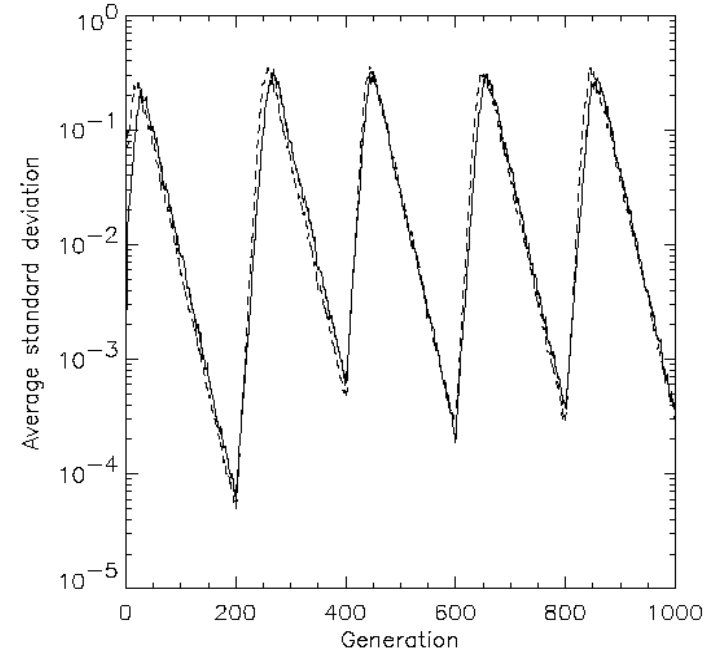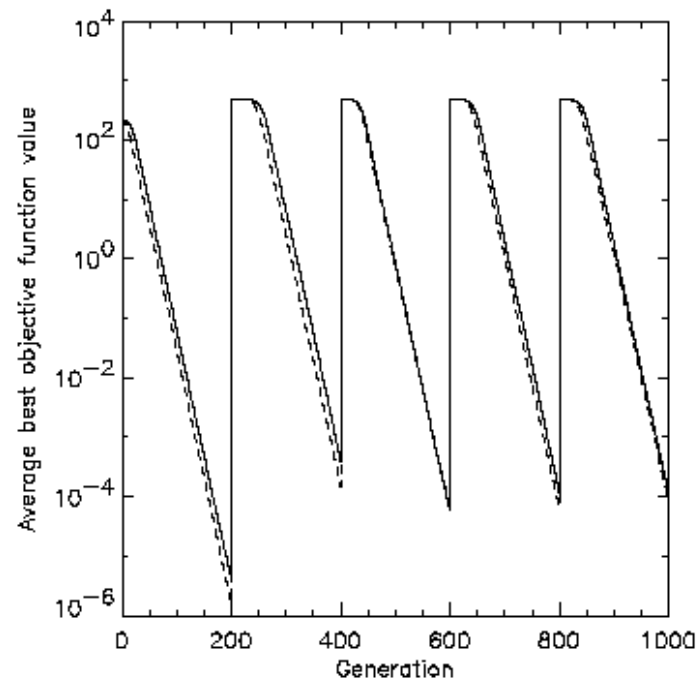
Changes in the fitness values (left) and the mutation step sizes (right)

Prerequisites for self-adaptation

Over the last decades much experience has been gained over self-adaptation in ES. The accumulated knowledge has identified necessary conditions for self-adaptation:

- $\mu > 1$ so that different strategies are presented
- $\lambda > \mu$ to generate offspring surplus
- Not "too" strong selection, e.g., $\lambda \approx 7 \cdot \mu$
- (μ,λ)-selection to get rid of maladapted σ's (to guarantee extinction of maladapted individuals)
- Mixing strategy parameters by especially **intermediate recombination** on them.

Example application: the Ackley function (Bäck et al '93)

- The Ackley function (here used with n =30):

$$f(x) = -20 \cdot \exp\left(-0.2\sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$$

- Evolution strategy:
  - Representation:
    - $-30 < x_i < 30$ (coincidence of 30's!)
    - 30 step sizes
  - (30,200) selection
  - Termination : after 200,000 fitness evaluations
  - Results: average best solution is $7.48 \cdot 10^{-8}$ (very good)

Example application: the cherry brandy experiment

- In a classic experiment to illustrate the fact that human beings can sometimes have a role to play within the evolution of good solutions, Herdy [193] describes how a group of students is used to act as a subjective evaluation function for an ES.

- The aim of the evolution is to discover appropriate quantities of clear water, red, yellow, and blue dye that when put in a test tube and mixed will create 30 ml of liquid with the same colour as a well-known brand of cherry brandy.

- The representation chosen is a simple vector of length 4 (one component representing the quantity of each ingredient used), and during the genotype to phenotype mapping the values are scaled to yield the 30-ml final volume.

- When humans are included "in the loop" of an evaluation process, a common finding is that they tend to **get bored quickly** and **do not deliver consistent judgments**.

Example application:
the cherry brandy experiment

- For this reason the algorithm is carefully designed to deliver fast convergence, and the students are just required to pick the single best offspring for survival. In this way, a relatively high selection pressure is used: a (1,8) strategy (the students are grouped into eights).

- Because of the short run lengths, **self-adaptation of step sizes is not used** (it would take too long), and only a single step size is used for each offspring.

- These are chosen so that **one third have $\delta = 1$**, **one third have a lower value**, and **one third a higher value**.

Example application: the cherry brandy experiment

- Task to create a colour mix yielding a target colour (that of a well known cherry brandy)

- Ingredients: water + red, yellow, blue dye

- Representation: $\langle$ w, r, y ,b $\rangle$ no self-adaptation!

- Values scaled to give a predefined total volume (30 ml)

- Mutation: lo / med / hi σ values used with equal chance

- Selection: (1,8) strategy

Example application: cherry brandy experiment cont'd

- Fitness: students effectively making the mix and comparing it with target colour
- Termination criterion: student satisfied with mixed colour
- Solution is found mostly within 20 generations
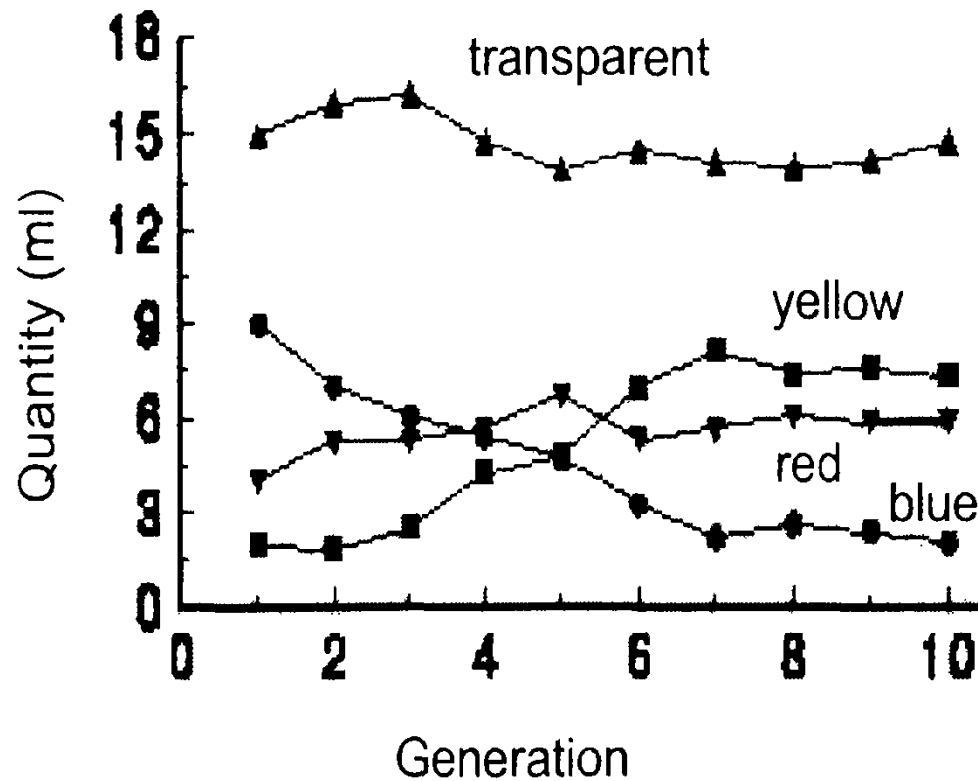- Accuracy is very good

Fig. 4.6. Evolution of a mixture of dyes to produce a cherry brandy colour using subjective evaluation