Center for Applied Systems Analysis

Genetic Algorithms, Evolution Strategies and AI

Tutorial MP3

Thomas Bäck

Center for Applied Systems Analysis (CASA) · Informatik Centrum Dortmund
Joseph-von-Fraunhofer-Str. 20 · D-44227 Dortmund

and

Leiden Institute for Advanced Computer Science
Niels Bohrweg 1 · NL-2333 CA Leiden

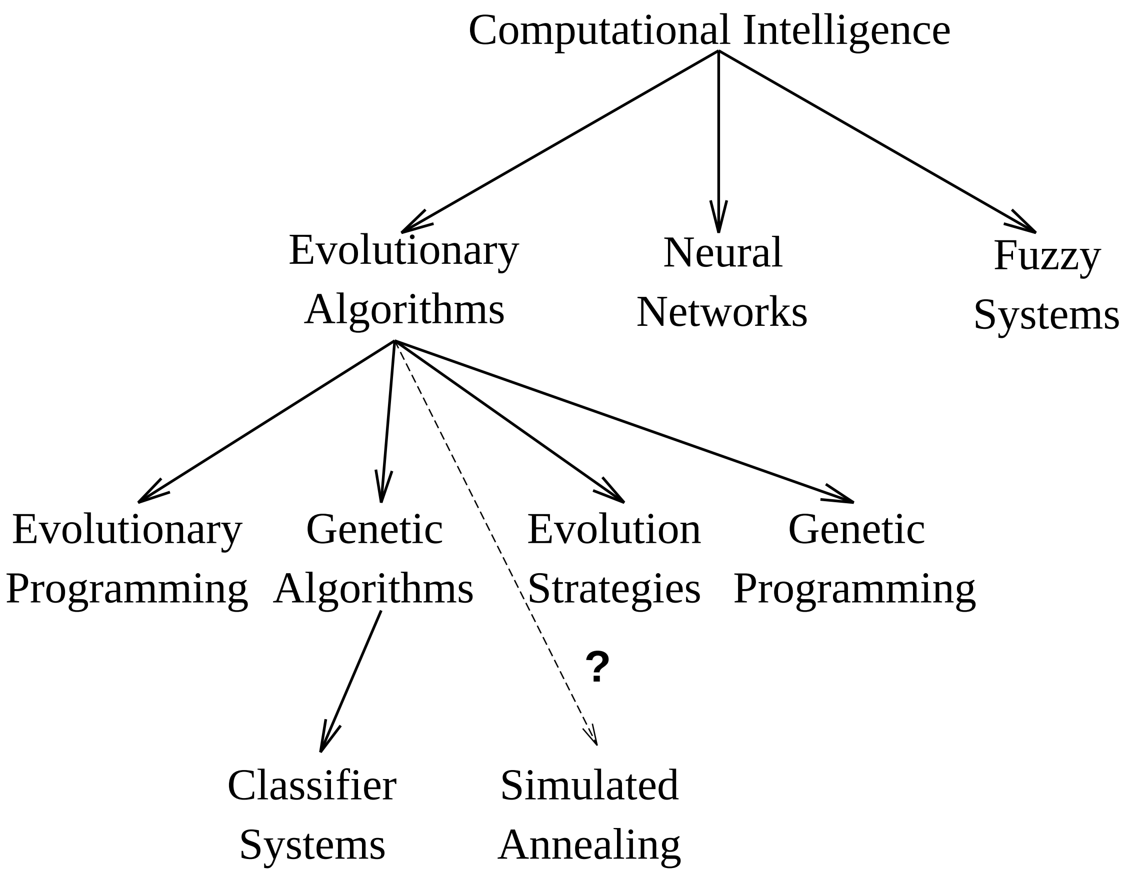
Tutorial Overview (1)

- Computational Intelligence and Evolutionary Algorithms
- General Characteristics of Evolutionary Algorithms
- Evolution Strategies:
  - Representation
  - Mutation
  - Recombination
  - Selection
  - Other Components & Algorithm
- Classification of Parameter Adaptation Methods

Tutorial Overview (2)

- Self-Adaptation in Evolution Strategies
- Self-Adaptation in Evolutionary Programming
- Some Theory of Evolution Strategies
- Application Examples of Evolution Strategies

Evolution Strategies



Evolutionary Algorithms

1. Set of candidate solutions (*individuals*): *Population*.
2. Generating candidates by:
   - Reproduction: Copying an individual.
   - Crossover (recombination): ≥ 2 parents $\rightarrow$ ≥ 2 children.
   - Mutation: 1 parent $\rightarrow$ 1 child.
3. Quality measure of individuals: *Fitness function, objective function*.
4. *Survival-of-the-fittest* principle.

Main components of EAs

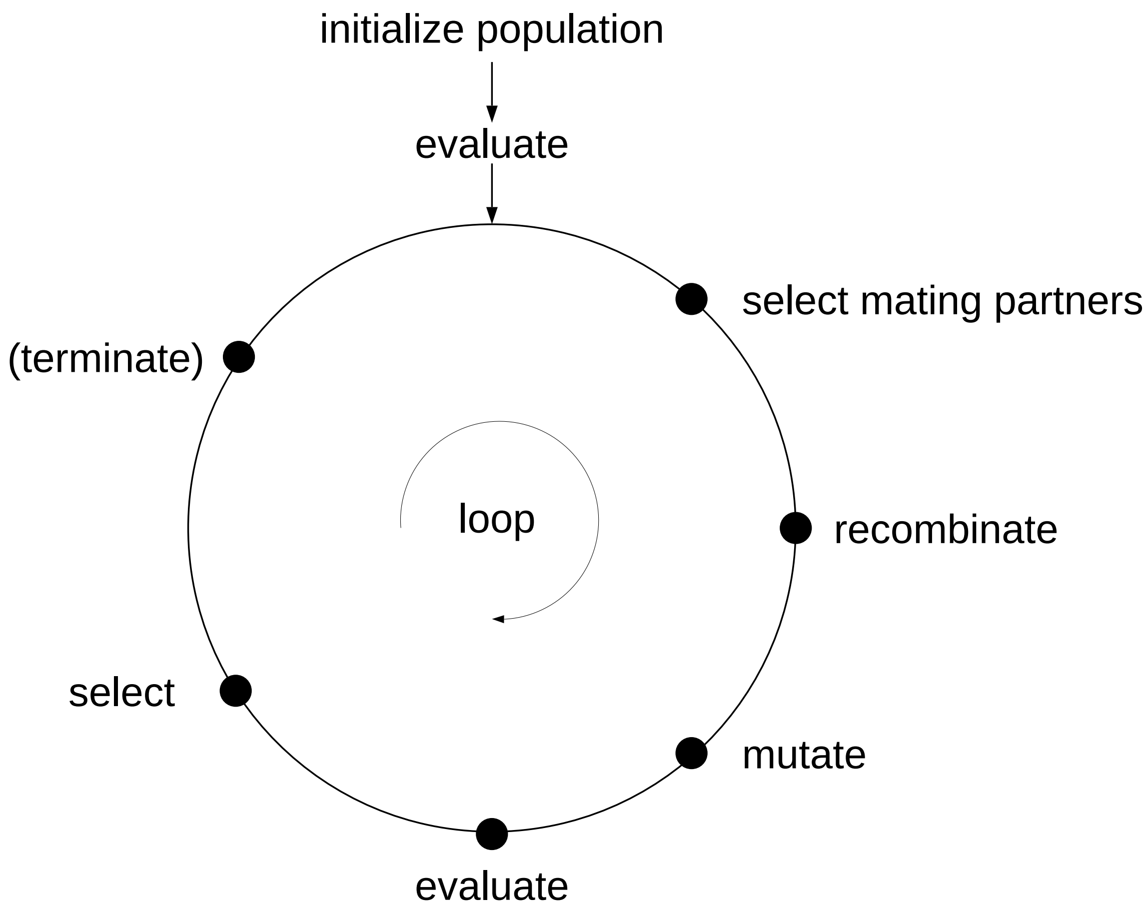
1. Representation of individuals: Coding.
2. Evaluation method for individuals: Fitness.
3. Initialization procedure for the 1st generation.
4. Definition of variation operators (mutation and crossover).
5. Parent (mating) selection mechanism.
6. Survivor (environmental) selection mechanism.
7. Technical parameters (e.g. mutation rates, population size).

‘Optimal’ Parameter Tuning:

- Experimental tests.
- Adaptation based on measured quality.
- $\Rightarrow$ Self-adaptation based on evolution !

Center for Applied Systems Analysis

The Evolution Loop



Algorithm Outline

$t := 0;$
initialize $P(t)$;
evaluate $P(t)$;
while not *terminate* **do**
  $P'(t) :=$ *select-mates*$(P(t))$;
  $P''(t) :=$ *variation*$(P'(t))$;
  evaluate$(P''(t))$;
  $P(t+1) :=$ *select*$(P''(t) \cup P(t))$;
  $t := t + 1;$
od

- *Variation* summarizes recombination and mutation.
- *Selection* can take old parents into account.

Advantages of EAs

- Widely applicable, also in cases where no (good) problem specific techniques are available:
  - Multimodalities, discontinuities, constraints.
  - Noisy objective functions.
  - Multiple criteria decision making problems.
  - Implicitly defined problems (simulation models).
- No presumptions with respect to the problem space.
- Low development costs; i.e. costs to adapt to new problem spaces.
- The solutions of EA's have straightforward interpretations.
- They can be run interactively (online parameter adjustment).

Disadvantages of EAs

- No guarantee for finding optimal solutions within a finite amount of time: True for all global optimization methods.
- No complete theoretical basis (yet), but much progress is being made.
- Parameter tuning is largely based on trial and error (genetic algorithms); solution: *Self-adaptation* (evolution strategies).
- Often computationally expensive: *Parallelism*.

Evolution Strategies: Main Characteristics

- Often continuous search spaces, $\mathbb{R}^n$.
- Emphasis on mutation: n-dimensionally normal-distributed, expectation zero.
- Various recombination operators.
- Deterministic (μ,λ)-selection.
- *Self-adaptation* of strategy parameters: First self-adaptive EA.
- Generation of an offspring surplus $\lambda \gg \mu$.

Representation (1)

Spaces:

- Search space:

$$\mathbb{R}^n$$

- Strategy parameter space (standard deviations and rotation angles of mutation): *Internal model*

$$\mathcal{S} = \mathbb{R}_+^{n_\sigma} \times [-\pi, \pi]^{n_\alpha}$$

- Individual space:

$$I = \mathbb{R}^n \times \mathcal{S}$$

Representation (2)

One individual:

$$\boxed{\vec{a} = (\underbrace{(x_1, \ldots, x_n)}_{\vec{x}}, \underbrace{(\sigma_1, \ldots, \sigma_{n_\sigma})}_{\vec{\sigma}}, \underbrace{(\alpha_1, \ldots, \alpha_{n_\alpha})}_{\vec{\alpha}}) \in I}$$

The three parts of an individual:

$$\begin{array}{lcll} \vec{x} & : & \text{Object variables} & \Rightarrow \text{Fitness } f(\vec{x}) \\ \vec{\sigma} & : & \text{Standard deviations} & \Rightarrow \text{Variances} \\ \vec{\alpha} & : & \text{Rotation angles} & \Rightarrow \text{Covariances} \end{array}$$

Representation (3)

A strategy parameter set

$$s = (\vec{\sigma}, \vec{\alpha}) \in \mathcal{S}$$

- Is *part of* an individual.
- Represents the probability density function (p.d.f.) for its mutation.

n_σ	n_α	Remark
1	0	standard mutation
n	0	standard mutations
n	$n \cdot (n-1)/2$	correlated mutations
$1 \leq n_\sigma \leq n$	$(n - \frac{n_\sigma}{2})(n_\sigma - 1)$	general case (correlated mutations)

Possible settings of n_σ and n_α.

Center for Applied Systems Analysis

Simple Self-Adaptive Mutation (1)

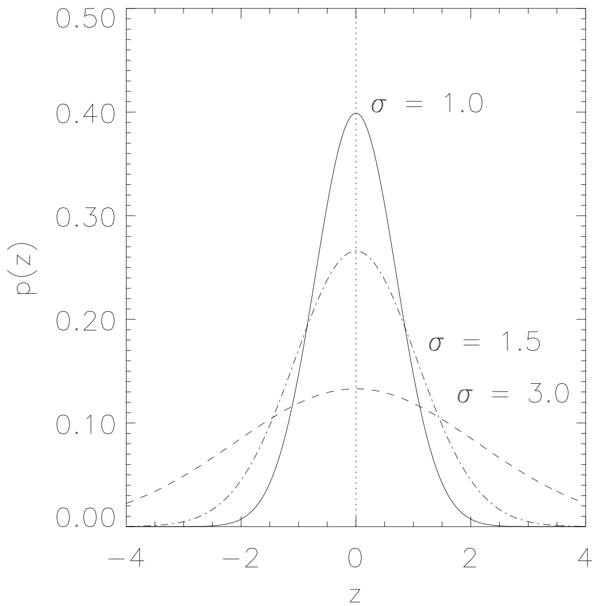
- Simple mutation makes use of normally distributed variations, $N(\xi, \sigma)$.

$$p(\Delta x_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\Delta x_i - \xi)^2}{2\sigma^2}\right)$$

- Expectation (ξ) is assumed to equal 0.
- Standard deviation (σ) must be adapted.

Simple Self-Adaptive Mutation (2)

The one-dimensional case:



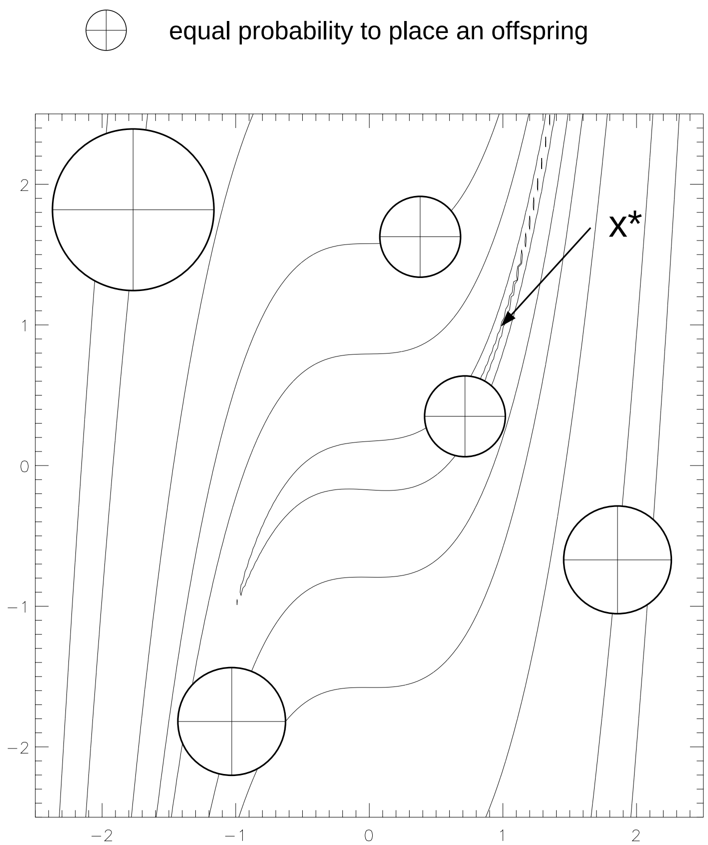
Simple Self-Adaptive Mutation (3)

- $n_\sigma = 1 \Rightarrow$ Low degree of freedom; one step size per individual.
- σ is mutated by multiplying by e^Γ, with Γ from a normal probability distribution.
- x_i is mutated by adding some Δx_i from a normal probability distribution.

$$\begin{array}{lcl} I & = & \mathrm{IR}^n \times \mathrm{IR}_+ \\ m'_{\{\tau_0\}}(\vec{x}, \sigma) & = & (\vec{x}', \sigma') \\ \tau_0 & \sim & 1/\sqrt{n} \end{array}$$

$$\boxed{\begin{array}{lcl} \sigma' & = & \sigma \cdot \exp(\tau_0 \cdot N(0,1)) \\ x'_i & = & x_i + \sigma' \cdot N_i(0,1) \end{array}}$$

Simple Self-Adaptive Mutation (4)



Simple mutations, $n = 2$, $n_\sigma = 1$, $(\Rightarrow n_\alpha = 0)$.

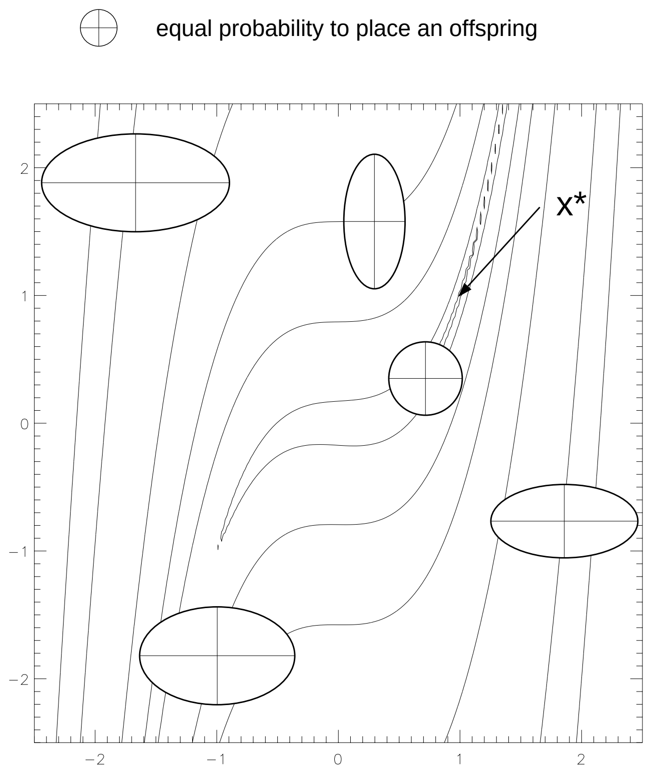
Simple Self-Adaptive Mutation (5)

- Now: $n_\sigma = n \Rightarrow$ Higher degree of freedom.
- Object variables x_i have their own, individual step sizes σ_i.

$$
\begin{array}{lcl}
I & = & \mathbb{R}^n \times \mathbb{R}^n_+ \\
m'_{\{\tau,\tau'\}}(\vec{x},\vec{\sigma}) & = & (\vec{x}',\vec{\sigma}') \\
\tau & \sim & 1/\sqrt{2\sqrt{n}} \\
\tau' & \sim & 1/\sqrt{2n}
\end{array}
$$

$$
\boxed{
\begin{array}{lcl}
\sigma'_i & = & \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\
x'_i & = & x_i + \sigma'_i \cdot N_i(0,1)
\end{array}
}
$$

Simple Self-Adaptive Mutation (6)



Simple mutations, $n = 2$, $n_\sigma = 2$, $(n_\alpha = 0)$.

Center for Applied Systems Analysis

Correlated Mutation (1)

- Correlated mutation uses the following probability distribution function for $\Delta\vec{x}$:

$$p(\Delta\vec{x}) = \sqrt{\frac{\det C}{(2\pi)^n}} \cdot \exp\left(-\frac{1}{2}\Delta\vec{x}^{\,T} \cdot C\Delta\vec{x}\right)$$

- C^{-1} is the covariance matrix:

$$\begin{aligned} c_{ii} &= \sigma_i^2 \\ c_{ij,(i\neq j)} &= \begin{cases} 0 & \text{no correlations} \\ \frac{1}{2}(\sigma_i^2 - \sigma_j^2)\tan(2\alpha_{ij}) & \text{correlations} \end{cases} \end{aligned}$$

- The pdf is just a generalized, n-dimensional normal distribution.

Correlated Mutation (2)

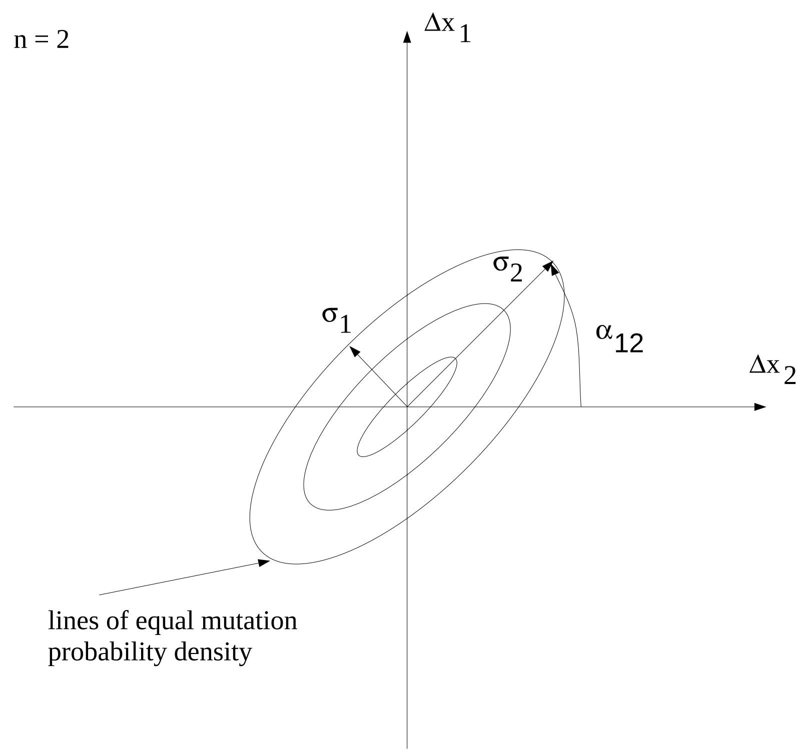


Illustration of the mutation ellipsoid for the case $n = 2$, $n_\sigma = 2$, $n_\alpha = 1$.

Correlated Mutation (3)

- Now: Up to $n \cdot (n+1)/2$ degrees of freedom facilitates learning of arbitrary preference directions.
- σ_i is mutated by multiplying by e^{Γ_i} with Γ_i from a normal probability distribution.
- α_j is mutated by adding some Δa_j from a normal probability distribution.
- $\vec{x}$ is mutated by adding some $\Delta\vec{x}$ from an n-dimensional normal distribution $\vec{N}(\vec{0}, C')$.

Correlated Mutation (4)

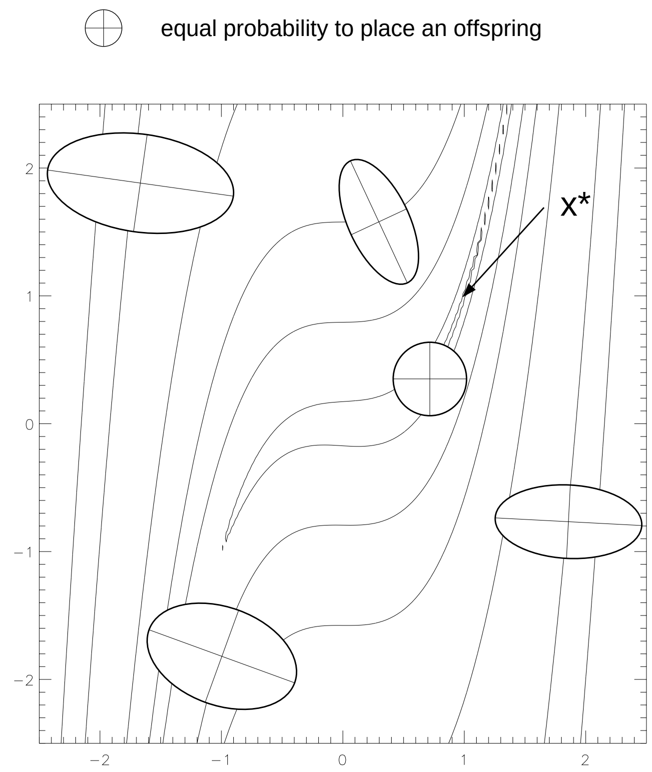
The formal description:

$$\begin{aligned} n_\alpha &= n \cdot (n-1)/2 \\ I &= \mathrm{IR}^n \times \mathrm{IR}^n_+ \times [-\pi, \pi]^{n_\alpha} \\ m'_{\{\tau,\tau'\,\beta\}}(\vec{x}, \vec{\sigma}, \vec{\alpha}) &= (\vec{x}', \vec{\sigma}', \vec{\alpha}') \\ \tau &\sim 1/\sqrt{2\sqrt{n}} \\ \tau' &\sim 1/\sqrt{2n} \\ \beta &\approx 5^\circ \end{aligned}$$

$$\boxed{\begin{aligned} \sigma'_i &= \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\ \alpha'_j &= \alpha_j + \beta \cdot N_j(0,1) \\ \vec{x}' &= \vec{x} + \vec{N}(\vec{0}, C') \end{aligned}}$$

Center for Applied Systems Analysis

Correlated Mutations (5)



Correlated mutations, $n = 2$, $n_\sigma = 2$, $n_\alpha = 1$.

Mutation Remarks (1)

Some remarks:

- Standard strategy: $n_\sigma = n$, $n_\alpha = 0$.
- For correlated mutations:
  - $\vec{\sigma}_c \sim \vec{N}(\vec{0}, C)$ is generated by a multiplication of the uncorrelated random vector $\vec{\sigma}_u$ by n_α rotation matrices (Schwefel 1981, Rudolph 1992).

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

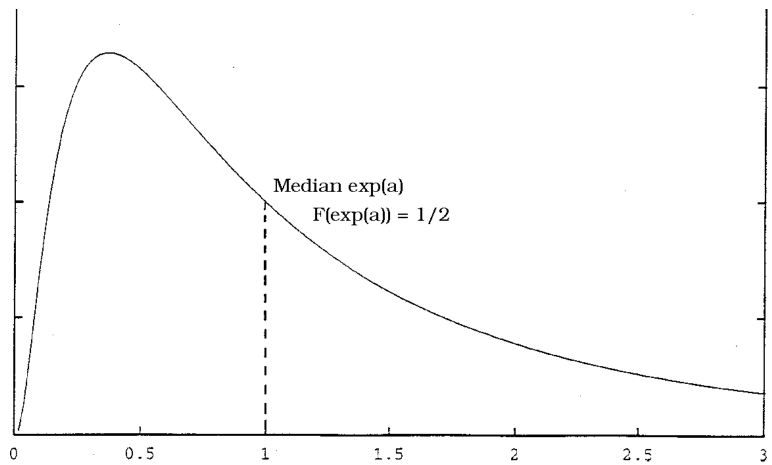
  - Exactly the feasible (positive definite) correlation matrices C can be created this way (Rudolph 1992).

Mutation Remarks (2)

Why log-normal distribution for σ_i-modification ?

Probability density function:

$$f_X(x) = \frac{1}{\sigma x \sqrt{2\pi}} \cdot \exp\left(-\frac{(\ln\ x - \mu)^2}{2\sigma^2}\right)$$



log-normal distribution, $\sigma = 1, \mu = 0$.

Mutation Remarks (3)

- Expectation:

$$\mathbf{E}(X) = \exp\left(\mu + \frac{\sigma^2}{2}\right)$$

- Median (defined by: $F_X(\exp(\mu)) = \frac{1}{2}$): $\exp(\mu)$ ($\mu = 0 \Rightarrow$ Median is one).

Advantages:

- Identical probability to sample x and $\frac{1}{x}$.
- Small changes more likely than large ones.
- σ_i are guaranteed to remain positive.

Evolution Strategies: Recombination (1)

Basic ideas:

- $I^\mu \to I$, μ parents yield 1 offspring.
- Is applied λ times, typically $\lambda \gg \mu$.
- Is applied to object variables as well as strategy parameters; often different for both.
- Per offspring gene two corresponding parent genes are involved.
- Two ways to recombine two parent alleles:
  - Discrete recombination: Choose one randomly.
  - Intermediate recombination: Average the values.
- Might involve two or (up to) μ parents (global recombination).

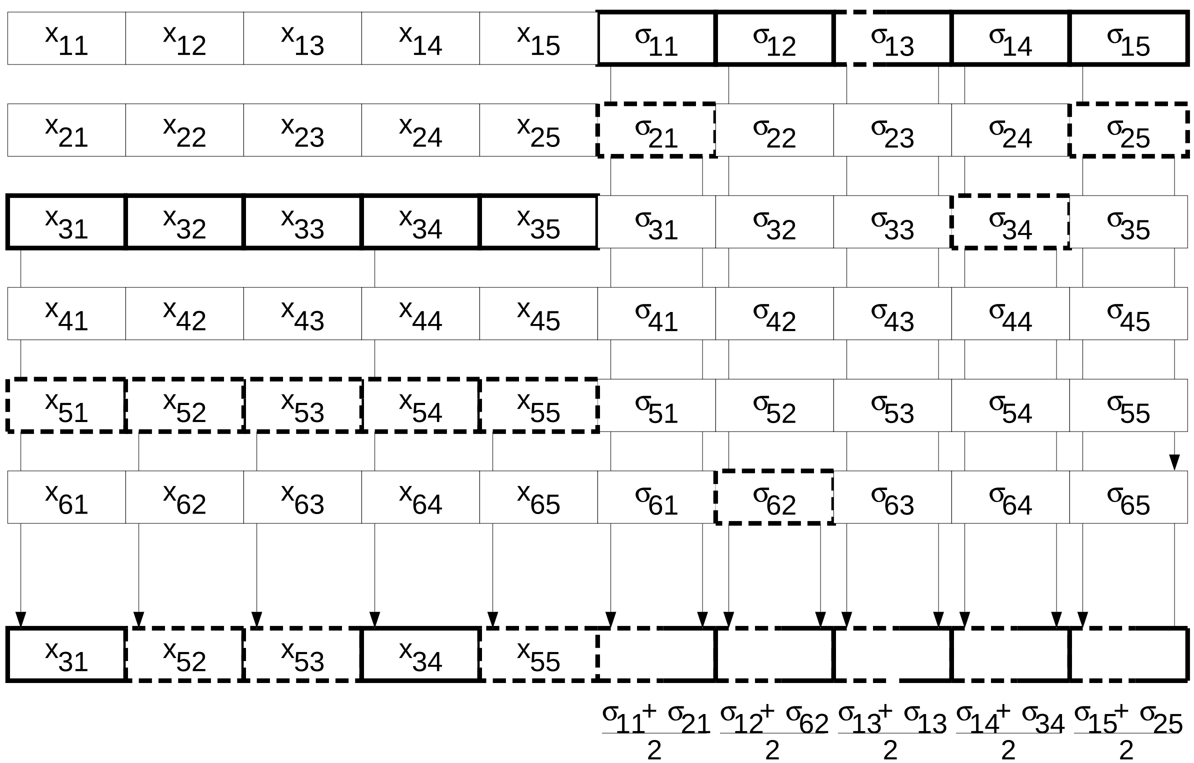
Evolution Strategies: Recombination (2)

Different methods:

- *Discrete*: Exchange of variables.
- *Intermediary*: Averaging of variables.
- In dual (2 parents) and global (up to μ parents) form:
  - Dual: Two parents are chosen at random for the creation of one offspring.
  - Global: One parent is chosen anew for *each component* of the offspring.
  - Recombination on $\vec{x}$, $\vec{\sigma}$, $\vec{\alpha}$ is usually different from each other !
  - Most commonly: Discrete recombination on object variables, global intermediate on strategy parameters.

Evolution Strategies: Recombination (3)

Recombination illustrated



Evolution Strategies: Recombination (4)

Example:

- Population size 6, 5 object variables, 5 strategy parameters.
- Local discrete recombination on x_i: 2 parents sampled; random decision for each x_i.
- Global intermediary recombination on σ_i: 1st parent held fixed; 2nd sampled for each x_i; averaging of parental x_i values.

Evolution Strategies: Selection (1)

- Strictly deterministic, rank-based.
- The μ best ranks are handled equally.
- (μ,λ)-selection:
  - $\lambda \gg \mu$.
  - The μ best of the offspring population $(P''(t))$ survive.
  - Important for self-adaptation.
  - Applicable also for noisy objective functions, moving optima.

Evolution Strategies

- $(\mu+\lambda)$-selection:
  - $\lambda < \mu$ possible.
  - The μ best out of parents and offspring $(P''(t) \cup P(t))$ survive.
  - Hinders self-adaptation to work.
  - Keeps best solution.
- Selective pressure: Very strong.

Evolution Strategies: Selection (2)

Selective pressure measured by takeover time τ^*:

Definition:
Number of generations until repeated application of selection completely fills the population with copies of the best individual (Goldberg and Deb 1991).

Remarks:

- Result for (μ,λ)-selection (Bäck 1994):

$$\boxed{\tau^* = \frac{\ln \lambda}{\ln(\lambda/\mu)}}$$

- $\tau^* \approx 2$ generations for a (15,100)-ES.
- Proportional selection in GAs: $\tau^* \approx \lambda \ln \lambda = 460$ generations!

Center for Applied Systems Analysis

Evolution Strategies: Other components

- Initialization:
  - x_i, α_i: randomly
  - σ_i: $\delta x_i/\sqrt{n}$, with δx_i a very rough measure for the distance to the optimum.
- Termination:
  - Termination after a number of generations.
  - Or iff $\max\{f(\vec{x}_i(t))\} - \min\{f(\vec{x}_i(t))\} \leq c(P(t))$.
    - $c(P(t))$ absolute ($= \varepsilon_1 > 0$), or
    - $c(P(t))$ relative ($= \varepsilon_2 \cdot |\bar{f}|$).

Evolution Strategies: Algorithm

$t := 0;$
initialize $P(0) := \{\vec{a}_1(0), \ldots, \vec{a}_\mu(0)\} \in I^\mu$ **where** $I = I\!R^n \times \mathcal{S};$
evaluate $P(0)\ :\ \{f(\vec{x}_1(0)), \ldots, f(\vec{x}_\mu(0))\};$
while not *terminate*$(P(t))$ **do**
    recombine: $\vec{a}'_k(t) := r'(P(t))\ \forall k \in \{1, \ldots, \lambda\};$
    mutate: $\vec{a}''_k(t) := m'_{\{\tau,\tau',\beta\}}(\vec{a}'_k(t))\ \forall k \in \{1, \ldots, \lambda\};$
    evaluate $P''(t) := \{\vec{a}''_1(t), \ldots, \vec{a}''_\lambda(t)\} : \{f(\vec{x}''_1(t)), \ldots, f(\vec{x}''_\lambda(t))\};$
    select: $P(t+1) :=$ **if** (μ, λ)*-selection*
        then $s_{(\mu,\lambda)}(P''(t));$
        else $s_{(\mu+\lambda)}(P''(t) \cup P(t));$
    $t := t + 1;$
od

Classification of Adaptation in EAs (1)

According to (Hinterding, Michalewicz, Eiben, 1997):

Type of adaptation:

- Static (i.e., none: Constant parameter settings).
- Dynamic (i.e., parameters modified during run).
  - **D**eterministic:
    Parameter altered by some deterministic rule.
  - **A**daptive:
    Monitor progress, use feedback mechanism to determine direction and/or magnitude of change.
  - **S**elf-**A**daptive:
    Parameters encoded in individuals, undergo evolution.

Classification of Adaptation in EAs (2)

Level of adaptation:

- **E**nvironment:
  Fitness function changes.
- **P**opulation:
  Concerns global parameters which apply to all population members.
- **I**ndividual:
  Concerns strategy parameters which apply to single individuals.
- **C**omponent:
  Concerns strategy parameters local to some component of an individual.

Classification of Adaptation in EAs (3)

Combinations:

	Deterministic	Adaptive	Self-adaptive
Environment	E-D	E-A	E-SA
Population	P-D	P-A	P-SA
Individual	I-D	I-A	**I-SA**
Component	C-D	C-A	**C-SA**

Self-adaptation principles

- Biological model: Repair enzymes, mutator genes.
- No deterministic control: strategy parameters *evolve*.
- *Indirect* link between fitness and useful strategy parameter settings.
- Strategy parameters are conceivable as an *internal model* of the local topology.
- Typical approaches: I-SA and C-SA.
- Individual space:

$$I = M \times \mathcal{S}$$

  - M: Search space.
  - $\mathcal{S}$: Strategy parameter space.

The crucial claim (Schwefel 1987, 1992):

Self-adaptation of strategy parameters works

- Without exogenous control.
- By recombining/mutating the strategy parameters.
- By exploiting the implicit link between fitness and useful internal model.

Necessary conditions (found by experiments):

- Generation of a surplus, $\lambda > \mu$
- (μ, λ)-selection (to guarantee extinction of misadapted individuals.
- A not too strong selective pressure e.g., (15,100) where $\lambda/\mu \approx 7$, but clearly $\mu > 1$ is neccesary.
- Recombination also on strategy parameters (especially: intermediate recombination).

Empirical Test Design

- With simple functions (with predictable optimal σ_i values), check whether it works.
- Investigate impact of selection.
- Compare with optimal behavior (if known).

Test functions for experiments

- One common step size ($n_\sigma = 1$): Sphere model.

$$f_1(\vec{x}) = \sum_{i=1}^{n} x_i^2$$

- Appropriate scaling of variables ($n_\sigma = n$):

$$f_2(\vec{x}) = \sum_{i=1}^{n} i \cdot x_i^2$$

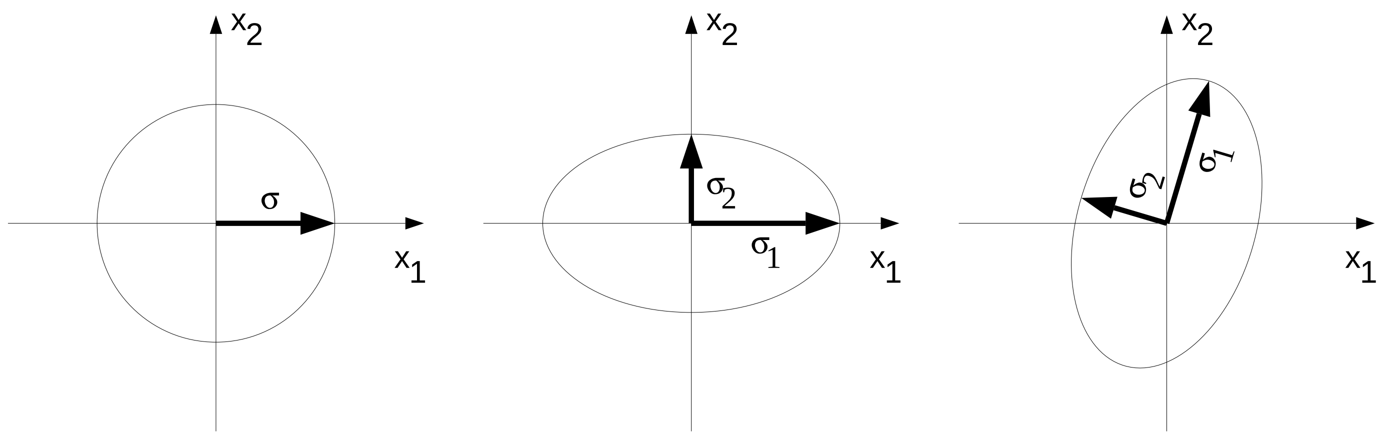
- A metric ($n_\sigma = n$, $n_\alpha = n \cdot (n-1)/2$):

$$f_3(\vec{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$$

Experiments

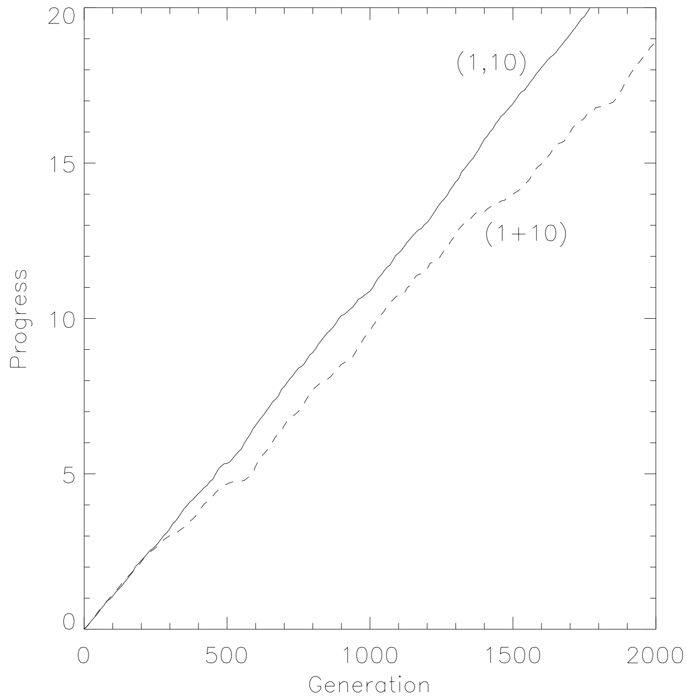
Sketch of the lines of equal probability density

- Left: Standard mutations, $n_\sigma = 1$.
- Middle: Standard mutations, $n_\sigma = 2$.
- Right: Correlated mutations, $n_\sigma = 2$, $n_\alpha = 1$.



Center for Applied Systems Analysis

Experimental Results on Sphere Model (1)



Convergence velocity of a $(1, 10)$-ES vs. that of a $(1 + 10)$-ES (sphere model f_1 with $n = 30$ and $n_\sigma = 1$).

Experimental Results on Sphere Model (2)

Progress measure:

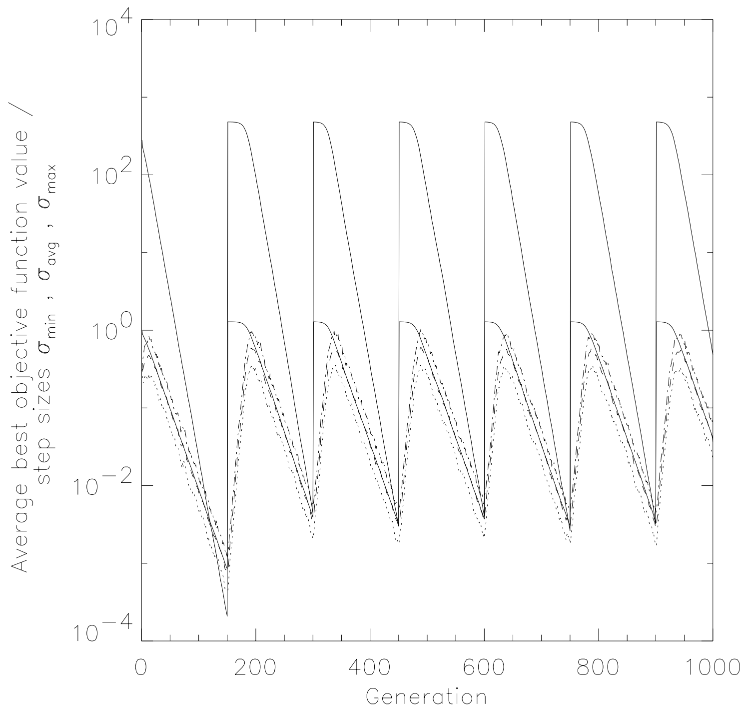
$$P_g = \log \sqrt{\frac{f_{\mathsf{min}}(0)}{f_{\mathsf{min}}(g)}}$$

- Counterintuitive: Elitist strategy is a bad choice.
- Misadapted σ might survive in an elitist strategy.
- Forgetting is necessary to prevent stagnation periods.

Time-Varying Sphere Model (1)

- Sphere model, $f(\vec{x}) = \|\vec{x} - \vec{x}^*\|^2 = R^2$.
- Optimum location $\vec{x}^*$ is shifted every 150 generations.
- (15,100)-ES, $n_\sigma = 1$, $n = 30$, no recombination.
- Simple model of a dynamic environment (with "catastrophes").

Time-varying Sphere Model (2)



Best objective function value and minimum, average, maximum and optimal standard deviation.

Time-varying Sphere Model (3)

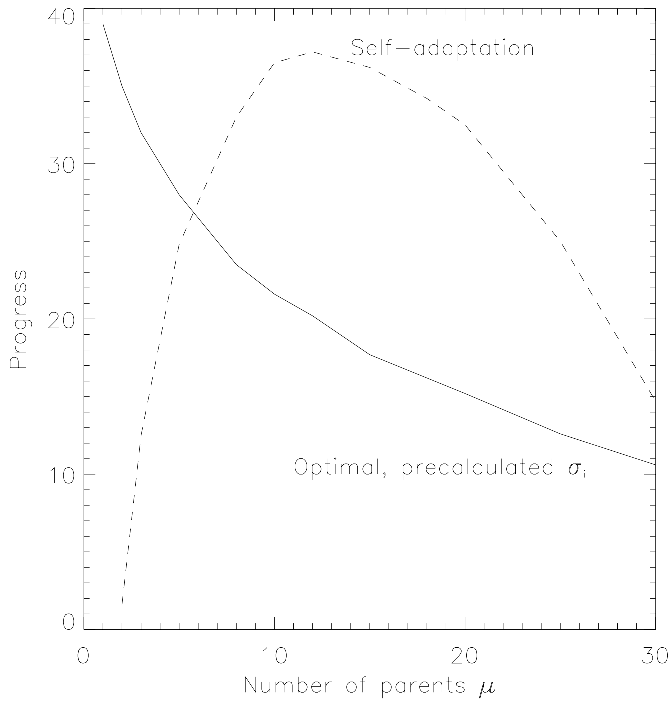
- Standard deviation σ adapts to the optimum value

$$\sigma_{opt} = c_{\mu,\lambda}\frac{R}{n} = c_{\mu,\lambda}\frac{\sqrt{f(\overline{x})}}{n}$$

- Transition time is $g \propto n$ (Beyer 1995).

⇒ The principle *learns* the optimal setting of the mutation rate ("internal strategy") without exogenous control.

Self-Adaptation is Collective Learning (1)

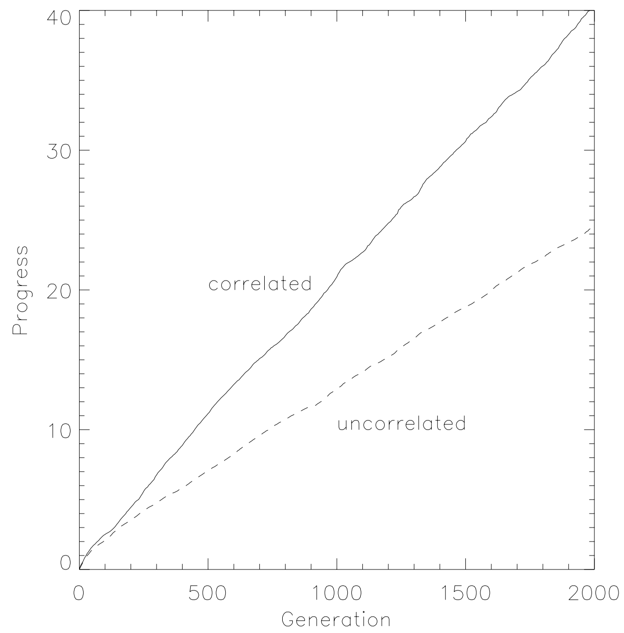


Average convergence velocity on f_2

Self-Adaptation is Collective Learning (2)

- $(\mu, 100)$-ES with $\mu \in \{1, \ldots, 30\}$
- $n_\sigma = n = 30$, and the optimum $\sigma_i \propto 1/\sqrt{i}$ is known.
- Optimum setting of σ_i: $\mu = 1$ performs best.
- Self-adaptation: $\mu = 12$ imperfect, diverse parents are as good as the optimal strategy.
- Individuals exchange information about their "internal models" by recombination.

Self-Adaptation of Covariances (1)



Convergence velocity of ES with correlated mutations vs. one with self-adaptation of standard deviations only, on f_3.

Self-Adaptation of Covariances (2)

- $(15, 100)$-ES, $n = n_\sigma = 10$, $n_\alpha = 45$.
- Recombination:
  - Intermediary on x_i.
  - Global intermediary on σ_i.
  - None on α_j (covariances).

> Covariances increase effectiveness in case of rotated coordinate systems.

Other Variants for Continuous Search Spaces

- Original EP:

$$\sigma' = \sigma \cdot (1 + \alpha \cdot N(0,1))$$

  Equivalent to log-normal with $n_\sigma = 1$, $\tau_0 = \alpha$ (Beyer 1995).

- Two-point distribution:

$$\sigma' = \begin{cases} \sigma \cdot \alpha & , \quad \text{if } u \sim U(0,1) \leq 1/2 \\ \sigma / \alpha & , \quad \text{if } u \sim U(0,1) > 1/2 \end{cases}$$

  (Mutational step size control after Rechenberg, $\alpha = 1.3$).

- Substitution of $N(0,1)$ by other distributions (e.g., one-dimensional Cauchy, Yao and Liu 1996).

Evolutionary Programming: Purpose

Simulate Evolution as a Learning Process to Generate Artificial Intelligence.

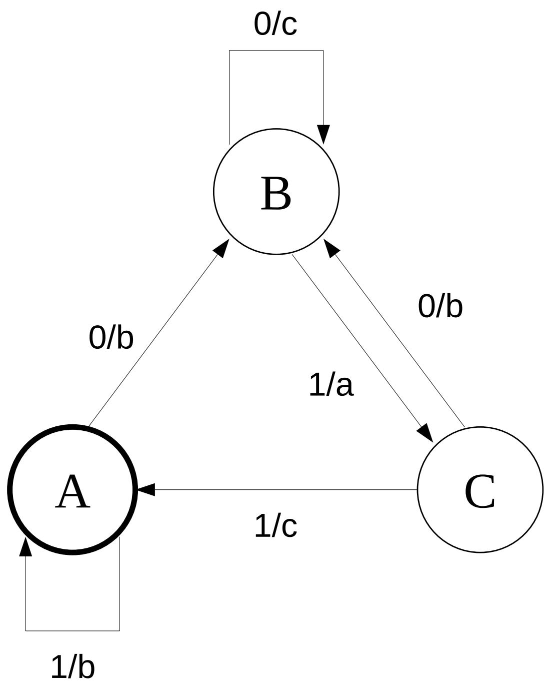
- Intelligence defined as the capability of a system to adapt its behavior to meet its goals in a range of environments (Fogel 1995).
- Intelligence viewed as adaptive behavior.
- Prediction of the environment is a prerequisite to intelligent behavior (prediction and response in the light of a given goal).
- Adaptation is not possible without a capability to predict.

Historical EP

Developed by L. Fogel (1962):

- Evolve a population of finite state machines (FSMs).
- FSMs provide successively better predictions of an environmental sequence.
- Predictions in light of a given goal.

Example of a Finite State Machine (1)

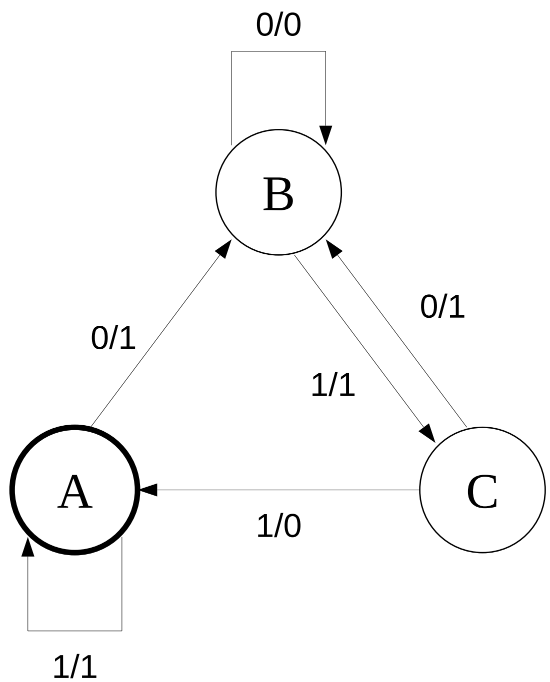


Example of a Finite State Machine (2)

- States $S = \{A, B, C\}$.
- Inputs $I = \{0, 1\}$, outputs $O = \{a, b, c\}$.
- Transition function $\delta : S \times I \to S \times O$.
- Transforms input stream into output stream.

Finite State Machines as Predictors

Performance measured on the basis of the machine's prediction capability, e.g. by $\text{output}_i = \text{input}_{i+1}$.



present state	C	B	C	A	A	B
input symbol	0	1	1	1	0	1
next state	B	C	A	A	B	C
output symbol	1	1	0	1	1	1

Initial state: C
Input string: 011101
Output string: 110111
Good predictions: 60 %

Search Operators

Mutation: Representation “naturally” determines the mutation operators:

- Change an output symbol.
- Change a state-transition.
- Add a state.
- Delete a state.
- Change the start state.

Crossover: None

Normally: All mutations with fixed probabilities p_i.

Here: Self-adaptation of p_i.

Self-adaptation of p_i

According to (Fogel, Angeline, Fogel, 1995):

- Associate p_i with each *component* of the FSM.
- Initial values of mutability parameters: $p_i^0 = 0.001$.
- Modification of strategy parameters p_i:

$$p_i' = p_i + \alpha \cdot N(0,1)$$

  ($\alpha = 0.01$).

- Two alternative methods:
  Selective $\leftrightarrow$ multi-mutational.

Selective Self-Adaptation

- Component selection for mutation based on

$$\mathcal{P}\{\text{Select comp. } i\} = \frac{p_i}{\sum p_k}$$

  (relative selection probabilities).

- Summation index k running over all components (related to the particular type of mutation).

- $p_i \geq \varepsilon = 0.001$ explicitly guaranteed.

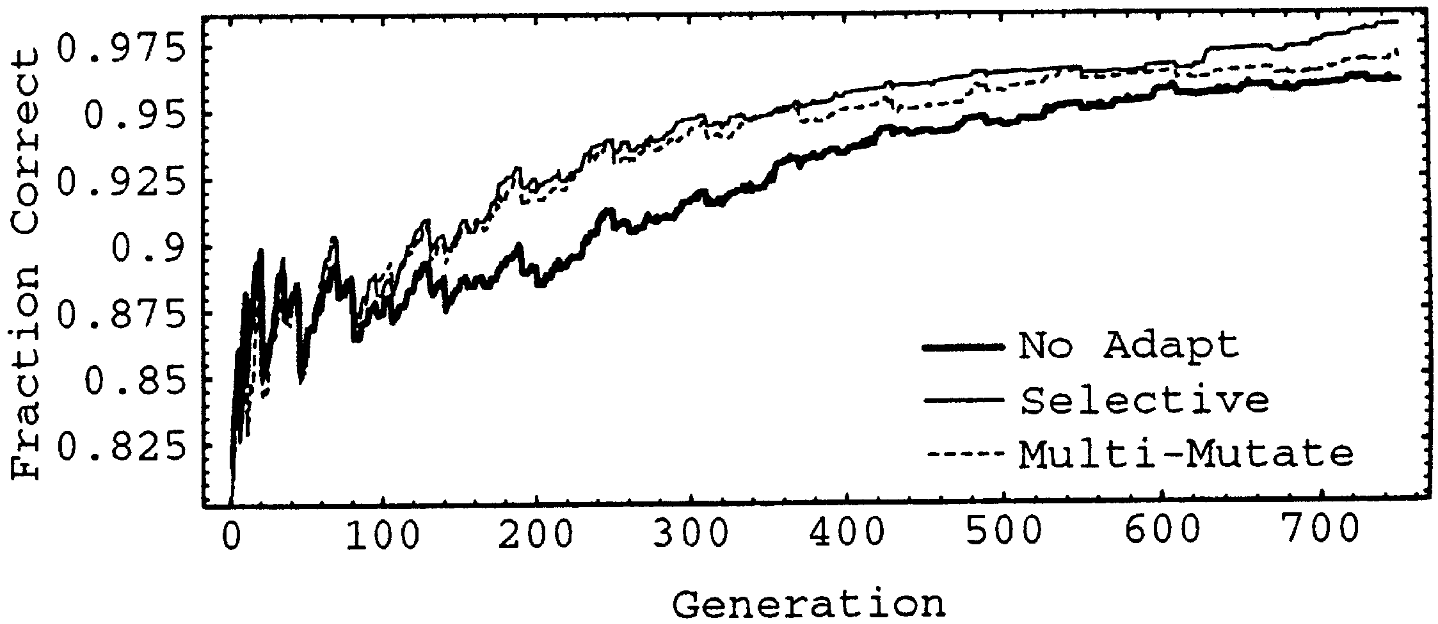
- Mutation of a component depends on p_i of other components.

Multi-Mutational Self-Adaptation

- The p_i are absolute mutation probabilities.
- $0.005 \leq p_i \leq 0.999$ explicitly guaranteed.
- Mutation of a component independent of p_i of other components.
- Greater diversity of offspring than selective self-adaptation.

Results (1)

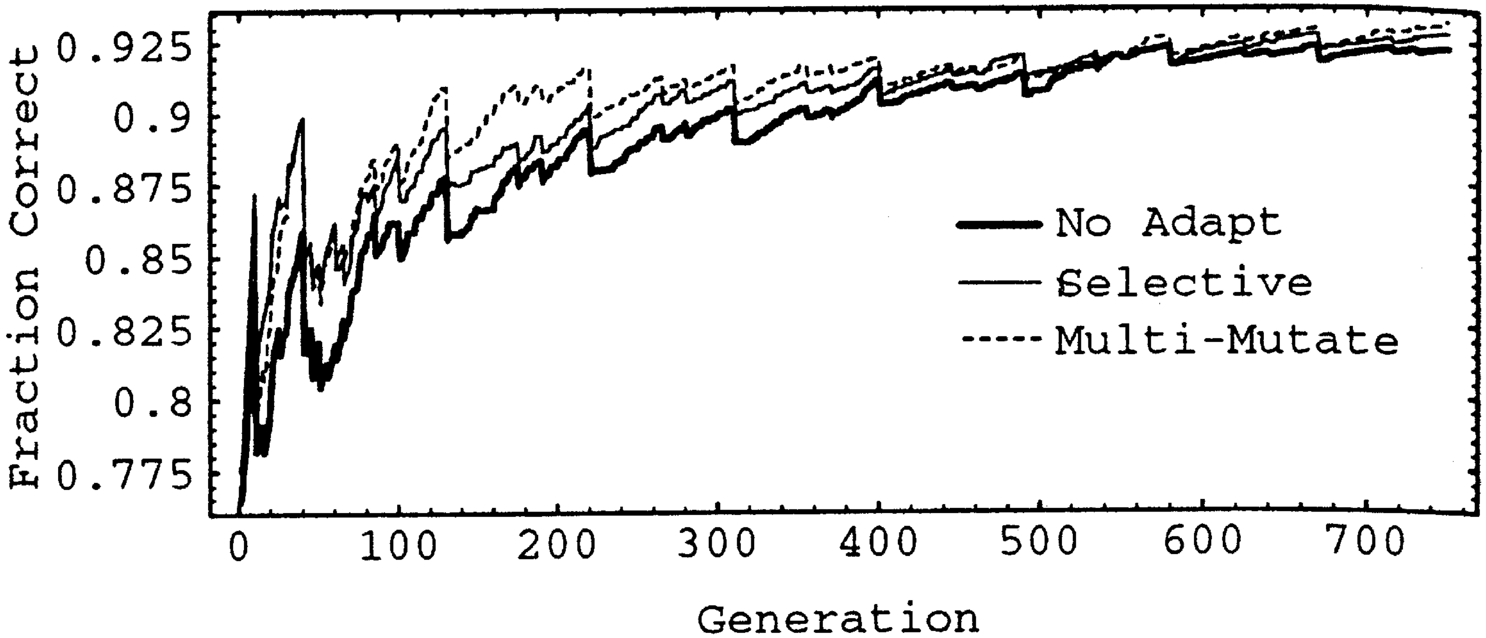
Simpler prediction experiment: $(101110011101)^*$



⇒ Selective self-adaptation slightly better.

Results (2)

More complex prediction experiment: $(101100111000110010)^*$



$\Rightarrow$ Multi-mutational self-adaptation better.

Conclusion

Multi-mutational self-adaptation

- explores a larger diversity, and therefore
- is more helpful on complex problems.

$\Rightarrow$ More work needed !

Modern EP

Applied for continuous parameter optimization

Similar to evolution strategy, with:

- Self-adaptation of n standard deviations (meta-EP).
- Self-adaptation of covariances (Rmeta-EP).
- $\mu = \lambda$ (i.e., parent and offspring population size are identical).
- No analogue of recombination.
- Probabilistic $(\mu + \mu)$-selection.

Mutation operator

Modifies strategy parameters and object variables

$$
\begin{aligned}
\sigma_i' &= \sigma_i \cdot (1 + \alpha \cdot N_i(0,1)) \\
x_i' &= x_i + \sigma_i' \cdot N_i(0,1)
\end{aligned}
$$

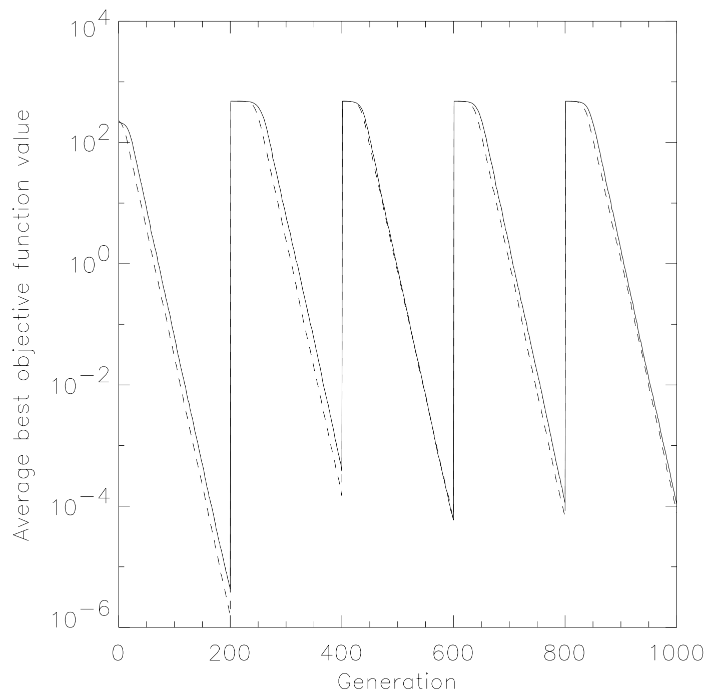
Recent results by Beyer (1995):

- For $\tau_0 = \alpha$ (small), $n_\sigma = 1$, the ES and EP method behave identically.
- Self-adaptation works for a variety of different pdf's for the modification of step sizes.

Center for Applied Systems Analysis

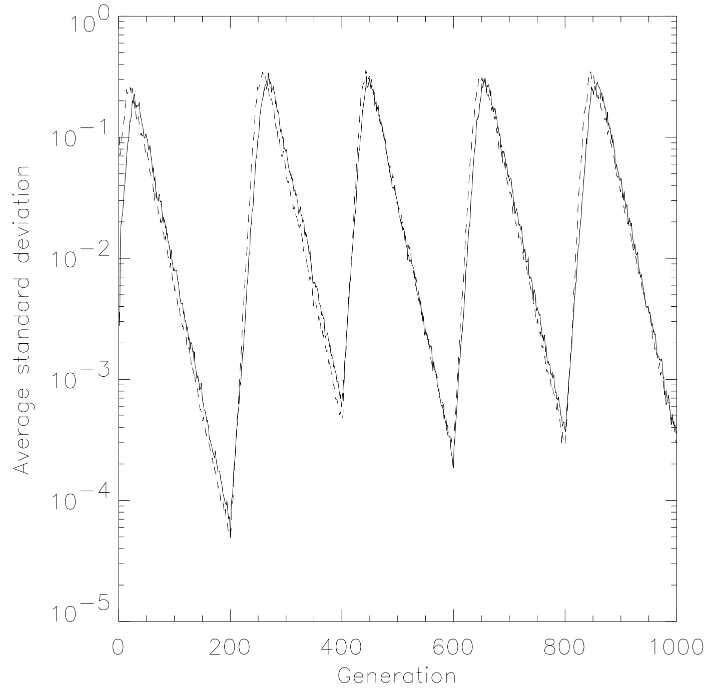
Experimental Test (1)

Best objective function value, time-varying sphere model, ES / EP:



Experimental Test (2)

Average mutation rate, time-varying sphere model, ES / EP:



Empirical Findings on Self-Adaptation (1)

1. Often, lognormal modifications outperform normal modifications.
   $\Rightarrow$ EP typically uses the ES method.
   (Saravanan 1994, Saravanan, Fogel 1994, Saravanan, Fogel, Nelson 1995).

2. On noisy objective functions, this behavior inverts (Angeline 1996).

3. It is important to modify σ_i first and use σ_i' to modify the object variables (Gehlhaar, Fogel 1996).

4. Self-adaptation works also with $(\mu + \lambda)$-selection.

5. Self-adaptation works also with $\mu = \lambda$.

6. Self-adaptation works also without recombination.
   The last three results from (Gehlhaar, Fogel 1996).

Empirical Findings on Self-Adaptation (2)

⇒ 1. confirms ES findings.

⇒ 2., 4., 5., 6. contradict ES findings.

- Definition of self-adaptation ?
- Quantitative measurement of self-adaptation ?
- Assessment for more complex objective functions ?
  (Until now only by experiment).
- Relation to learning in AI ?

Self-Adaptation: Conclusions

- Powerful & robust parameter control scheme.
- Optimal conditions concerning selection, population size, etc.?
- Perfect adaptation vs. useful diversity — or a mixture ?
- Optimal speed of self-adaptation (i.e., learning rate settings) ?
- Few theoretical results.

Self-Adaptation: Individuals as Agents

- Individuals are *autonomous*; internal control of their behavior (mutation).
- Individuals *communicate* by exchanging partial information (recombination).
- Individuals are *reactive* to their environment (objective function).
- Further possibilities:
  - Spatial communication structure (graph).
  - Parallel implementation.
  - More complex internal strategies; including symbolic representation.

Center for Applied Systems Analysis

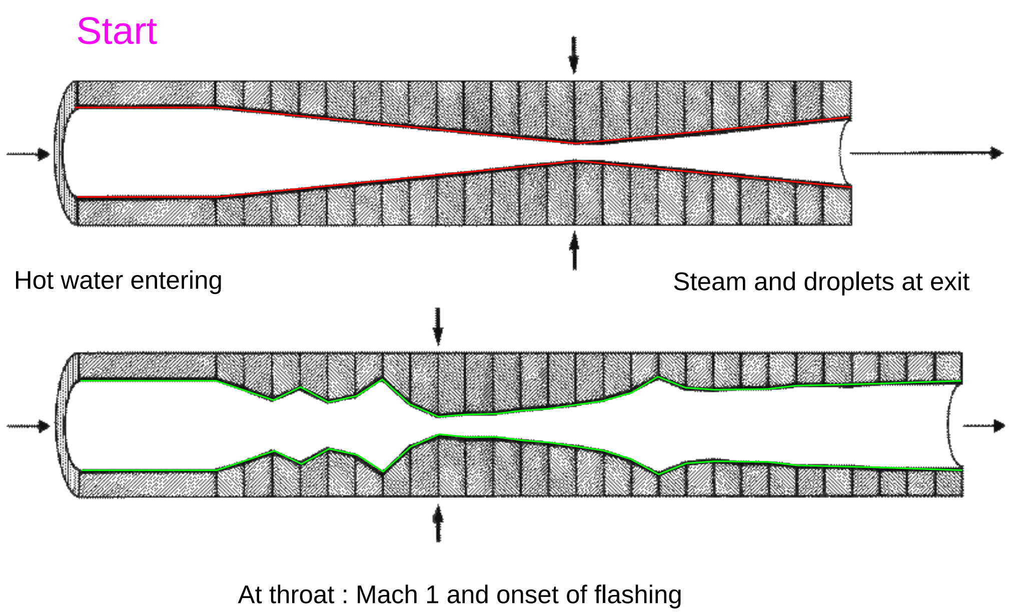
Application Fields

- Experimental optimization & optimization with subjective evaluation, e.g.:
  - Coffee recipes; general food recipes.
  - Biochemical fermentation processes.
  - Wind tunnel experiments.
  - Two-phase nozzle optimization experiments.
- Technical optimization:
  - Design & Production.
  - Logistics.
  - Control of dynamic processes.

Application Fields

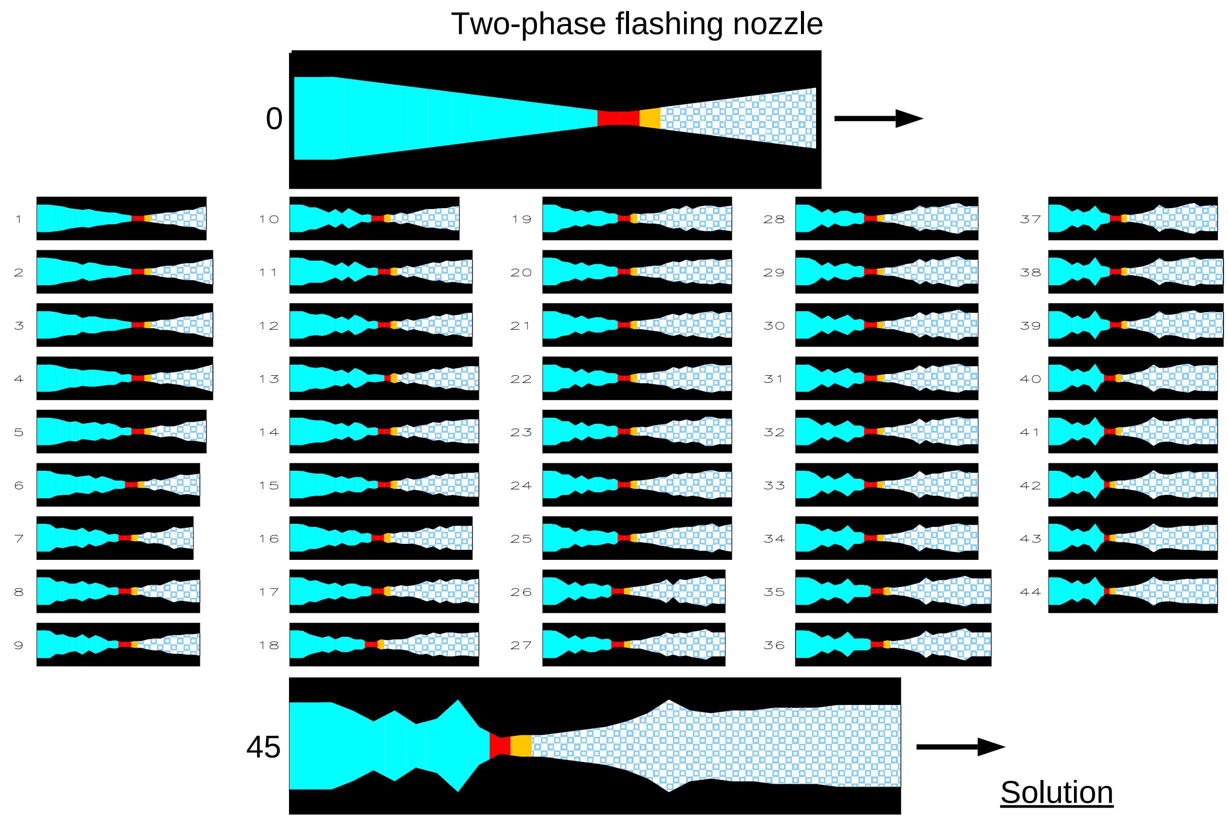
- Structure optimization, e.g.:
  - Structure & parameters of plants.
  - Connection structure & weights of neural nets.
  - Number of thicknesses of layers in multilayer structures.
- Data analysis, e.g.:
  - Clustering (number & centers of clusters).
  - Fitting models to data.
  - Time series prediction.

Application 1: Hot Water flashing nozzle (1)

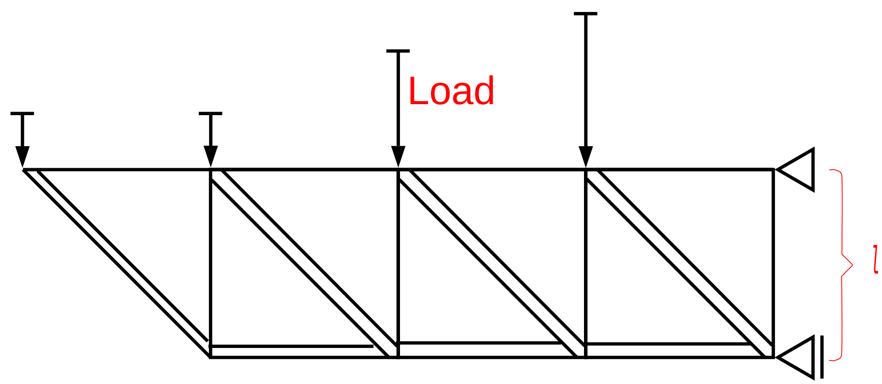


1968 AEG

Application 1: Hot Water flashing nozzle (2)



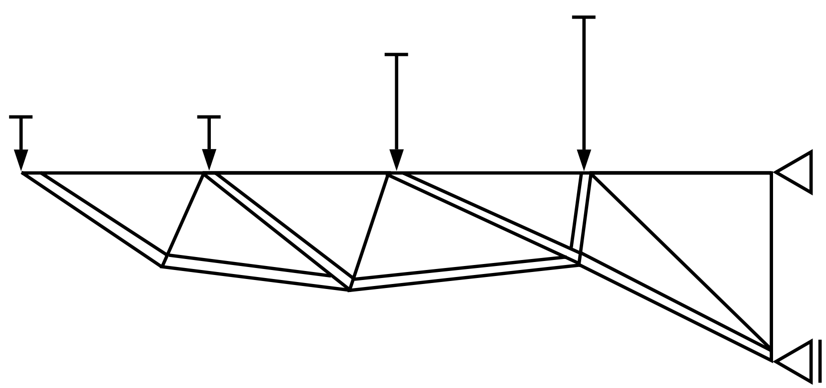
Application 2: Minimal weight truss layout



Start

922 kp

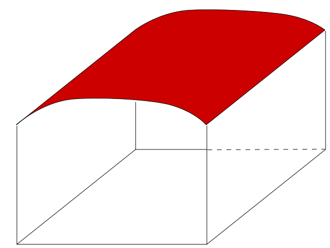
(LP minimum)



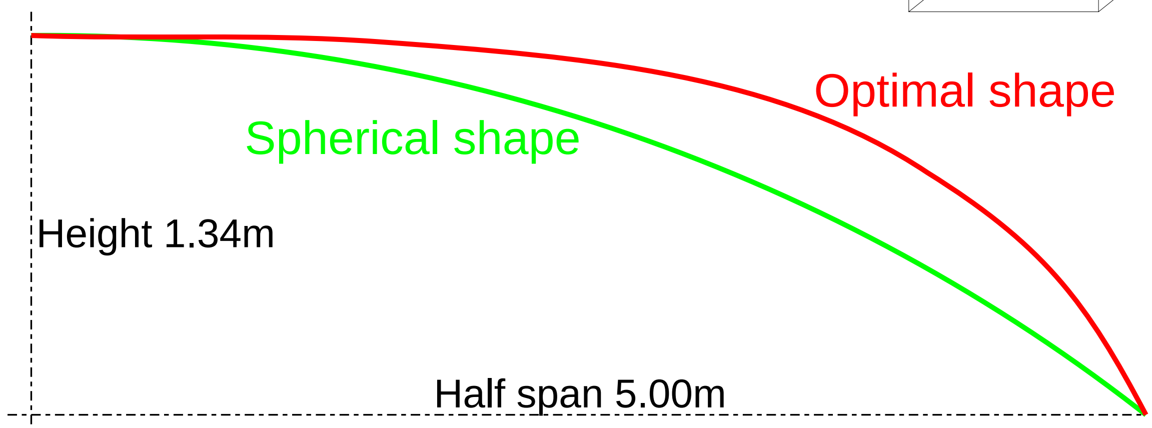
Optimum

738 kp

Application 3: Concrete shell roof



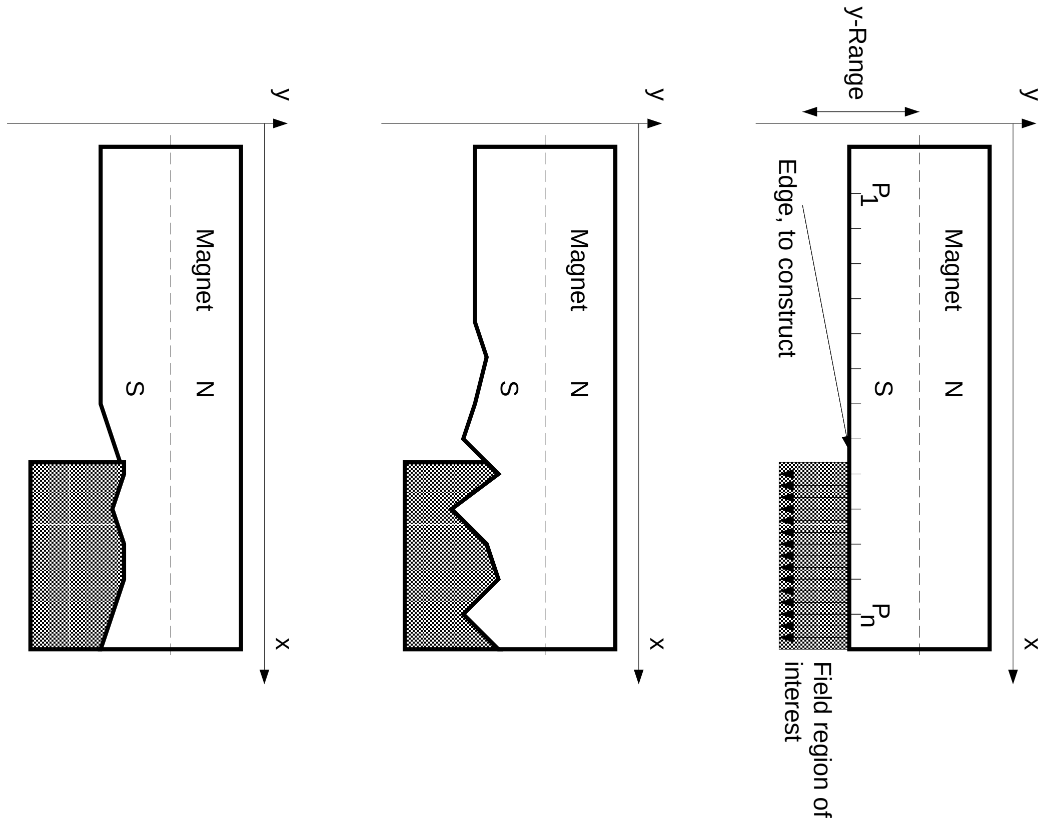
under own and outer load (snow and wind)



max $|m_\varphi| \rightarrow$ min
Orthogonal bending strength

Savings : 36% shell thickness
27% armation

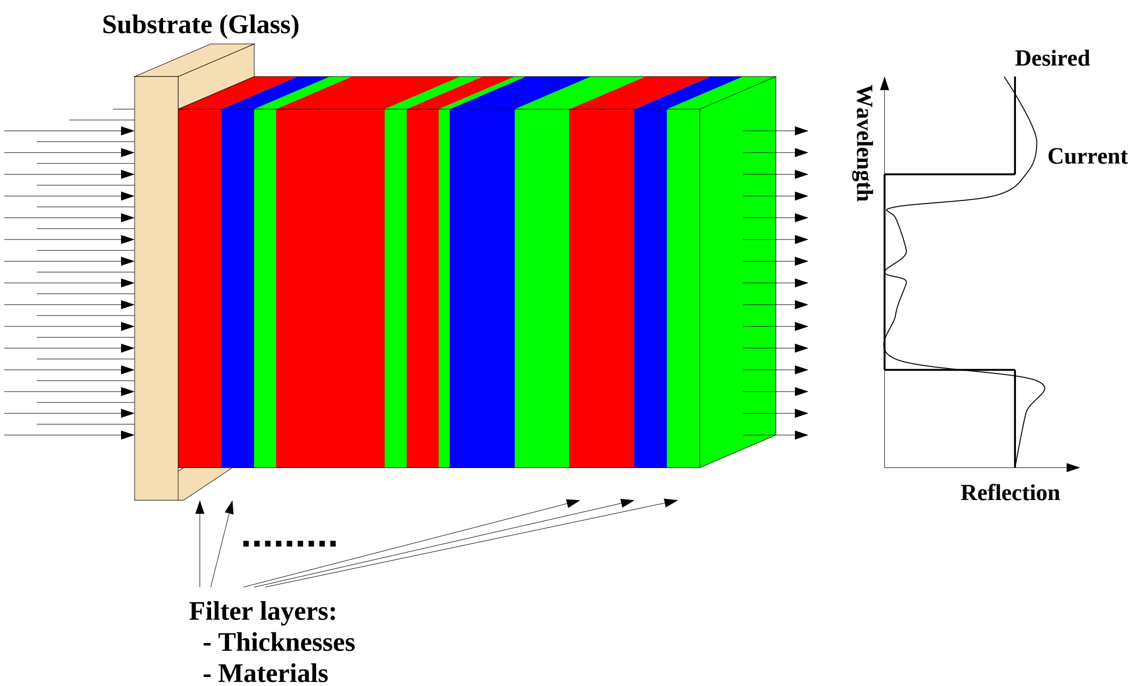
Application 4: Dipole Magnet Structure (1)



Application 4: Dipole Magnet Structure (2)

- Analysis of the magnetic field by Finite Element Analysis (FEM).
- Minimize sum of squared deviations from the ideal.
- Individuals: Vectors of positions $(y_1, \ldots, y_n)$.
- Middle: 9.82% better than upper graphic; bottom: 2.7% better.

Optical Multilayers (1)



Goal: Find a filter structure such that the real reflection behavior matches the desired one as close as possible.

Optical Multilayers (2)

Problem parameters:

- Thicknesses $\vec{d} = (d_1, \ldots, d_n)$ of layers.
- Layer materials $\vec{\eta} = (\eta_1, \ldots, \eta_n)$ (integer values).
- Number of layers n.

$\Rightarrow$ Mixed-integer, variable-dimensional problem.

Optical Multilayers (3)

Objective function:

$$\boxed{f(\vec{d},\vec{\eta}) = \int_{\lambda_d}^{\lambda^u} \left[R(\vec{d},\vec{\eta},\lambda) - \tilde{R}(\lambda)\right]^2 d\lambda}$$

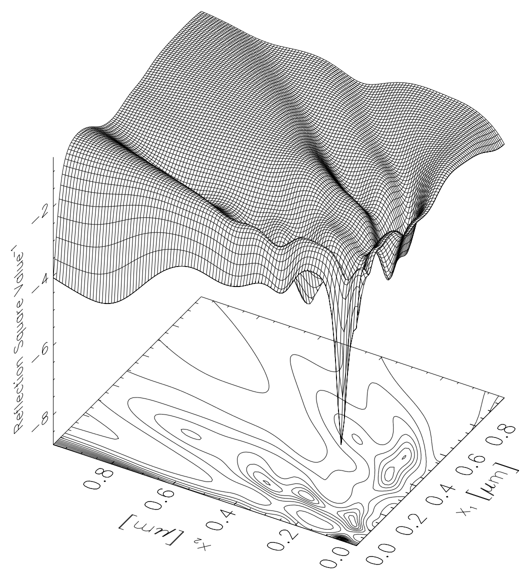
- $R(\vec{d},\vec{\eta},\lambda)$:
  Reflection of the actual filter for wavelength λ.
  Calculation according to *matrix method*.

- $\tilde{R}(\lambda)$: Desired reflection value.

Center for Applied Systems Analysis

Optical Multilayers (4)

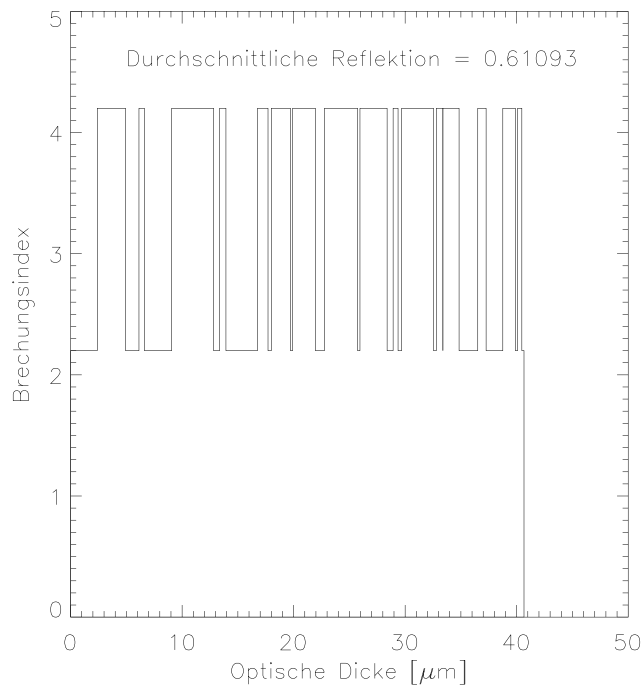
Example topology: Only layer thicknesses vary; $n = 2$.



Center for Applied Systems Analysis

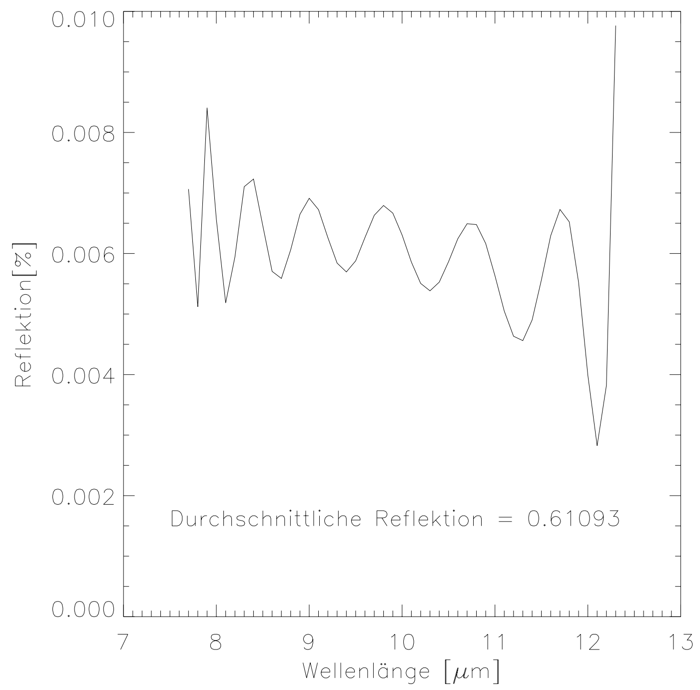
Optical Multilayers (5)

Example structure:



Optical Multilayers (6)

Example reflection:



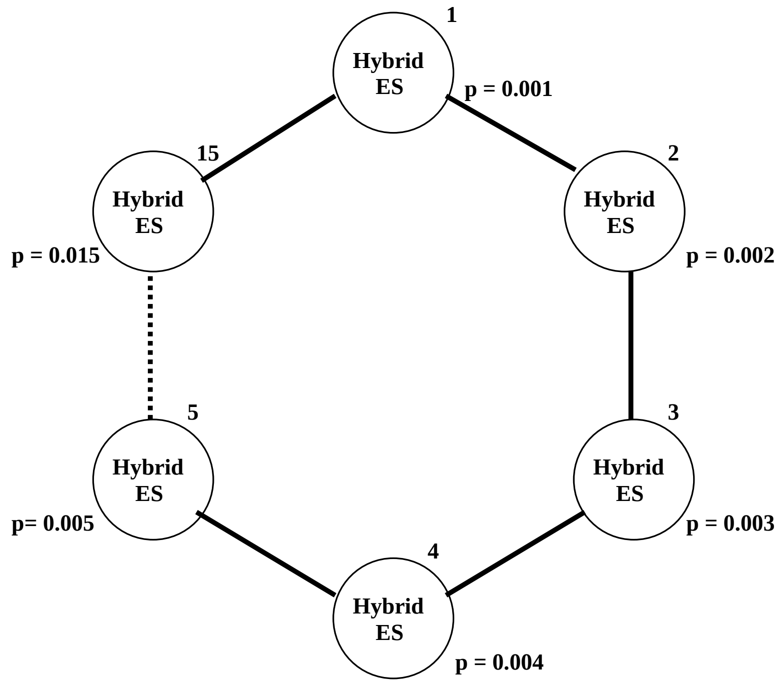
Optical Multilayers (7)

Existing Methods:

- Refinement methods:
  - Initial design constructed by an expert.
  - Local optimization of the initial design.
- Synthesis methods:
  - Without initial design (random start).
  - Automatical global optimization.

Optical Multilayers (8)

Parallel evolutionary algorithm:



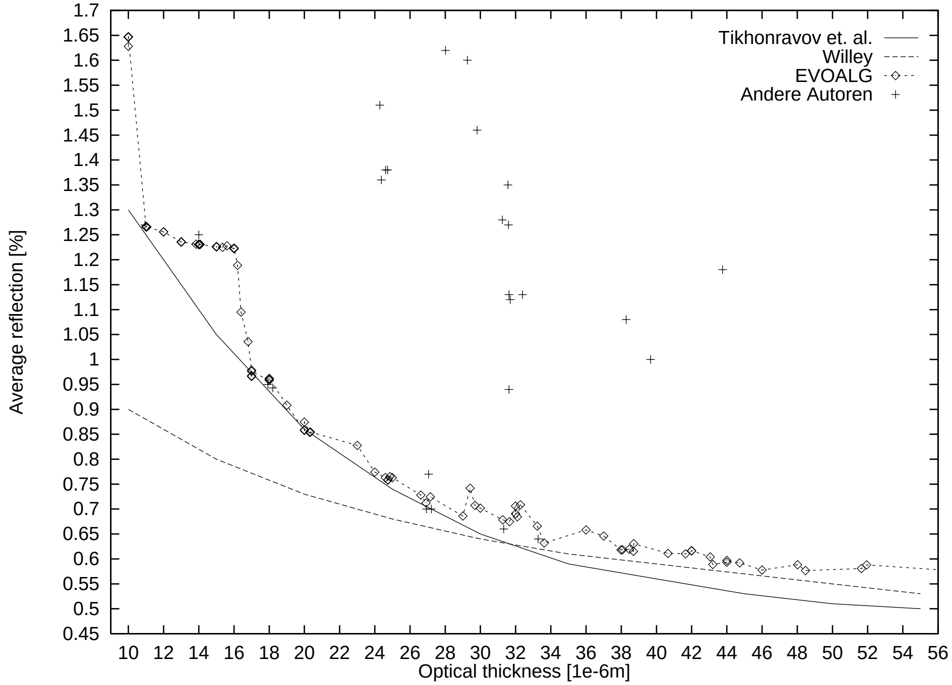
Optical Multilayers (9)

Parallel evolutionary algorithm:

- Per node: EA for mixed-integer representation.
- *Isolation* and *migration* of best individuals.
- Mutation of discrete variables: Fixed p_m per population.

Optical Multilayers (10)

Reference Results:



Comparison of literature results, theoretical predictions, and EA results.
⇒ Excellent algorithm for the synthesis of filters.

Parallel time series prediction (1)

Goals:

- Combine traditional statistical methods for time series analysis with parallel computational intelligence approaches.
- Estimate the parameters of a model chosen by experts by means of parallel evolutionary algorithms.
- Test the feasibility of the approach with classical statistical models (ARMA-models).
- Use the approach for the long-term sales forecast model of Lewandowski.

Parallel time series prediction (2)

ARMA-problem:

$$x_t = \alpha_1 x_{t-1} + \ldots + \alpha_p x_{t-p} + \varepsilon_t - \beta_1 \varepsilon_{t-1} - \ldots - \beta_q \varepsilon_{t-q}$$

- Parameters to be estimated:
  - p, q
  - $\alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q$

  $\Rightarrow$ mixed-integer problem of variable dimension.

- Estimation of error series (ε_t):
  $\widehat{\varepsilon}_t$ estimated in iterative process during the evolutionary algorithm: residuals of actual generation become estimates for error series for following generation.

Application Examples

Parallel time series prediction (3)

Coding (for the example of the α-vector):

genotype

n = 8

x	2.3	-2.1	4.1	0.2	1.7	-7.6	1.0	1.3
b	1	1	0	1	0	0	0	1

phenotype

p = 5

α_1	α_2	α_3	α_4	α_5
0.2	4.1	0.2	-4.9	1.3

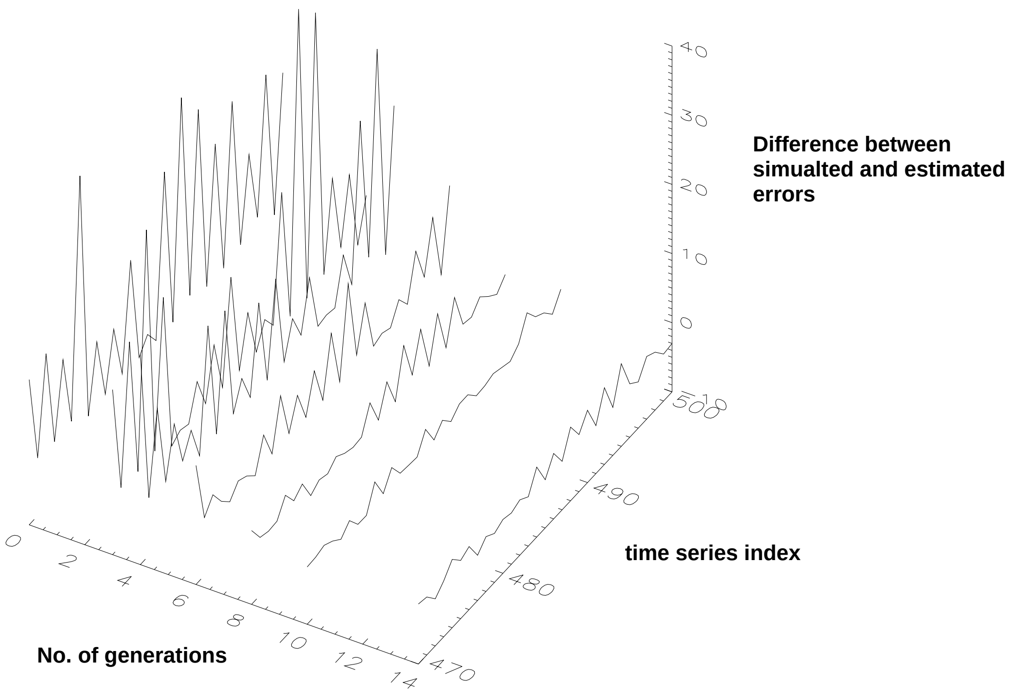
Parallel time series prediction (4)

Fitness function:

$$\sum[x_t - (\alpha_1 x_{t-1} + \ldots + \alpha_p x_{t-p} + \widehat{\varepsilon}_t - \beta_1 \widehat{\varepsilon}_{t-1} - \ldots - \beta_q \widehat{\varepsilon}_{t-q})]^2$$

$\Rightarrow$ Least-squares function)

Parallel time series prediction (5)



Parallel time series prediction (6)

Results:

- Estimation of error series very successful
- Least squares difference of identified models can compete with statistical software (SAS).
- About 20% of model orders p and q identified correctly.

Parallel time series prediction (7)

Parallel forecasting for sales planning:

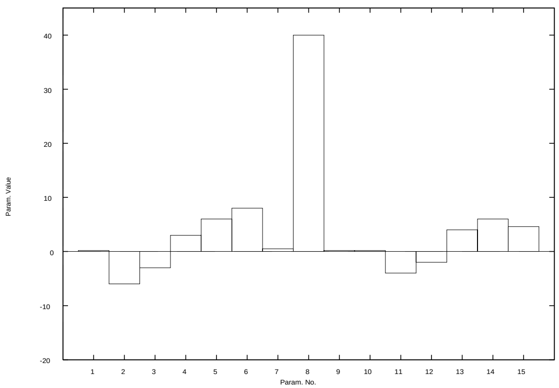
- Problems:
  - All influencing parameters have to be considered
    $\Rightarrow$ High complexity.
  - Updates needed in high frequency (daily, weekly)
    $\Rightarrow$ Time for calculations very important.
- Example: Forecasting the sales of a passenger car
  - Influencing variables: Price, Standard equipment, Model policy
  - Other factors as e.g. the economy (in form of the gross domestic product, unemployment rate, etc.) have to be taken into consideration.
- Model: Long-term Lewandowski model.

Parallel time series prediction (8)

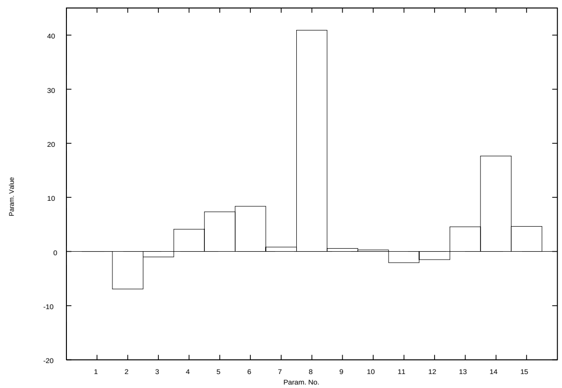
Results for passenger car problem:

- Model not easy to analyse, therefore treated as black box.
- Quality criterion: reached fit in comparison to a parameter setting based on expert knowledge.
- Comparison of parameter profiles:
  - Mean error in the past for parameter setting
    - (a): 0.99 (expert's parameter setting).
    - (b): 0.26 (optimized profile with initialisation by an expert).
    - (c): 0.25 (optimized profile with random initialisation).

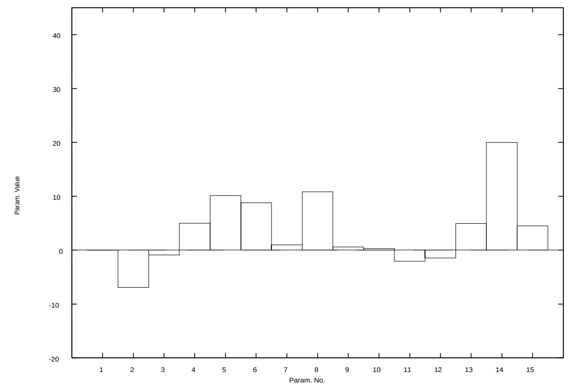
Application Examples



(a)



(b)



(c)

The mutation vector (1)

$$\triangle\vec{x} = \vec{z} = (z_1, \ldots, z_n)$$

$Z_1, \ldots, Z_n$: $(0, \sigma)$-normally distributed random variables.

$$\Rightarrow S^2 = \sum_{i=1}^{n} Z_i^2 \text{ is } \chi^2\text{-distributed.}$$

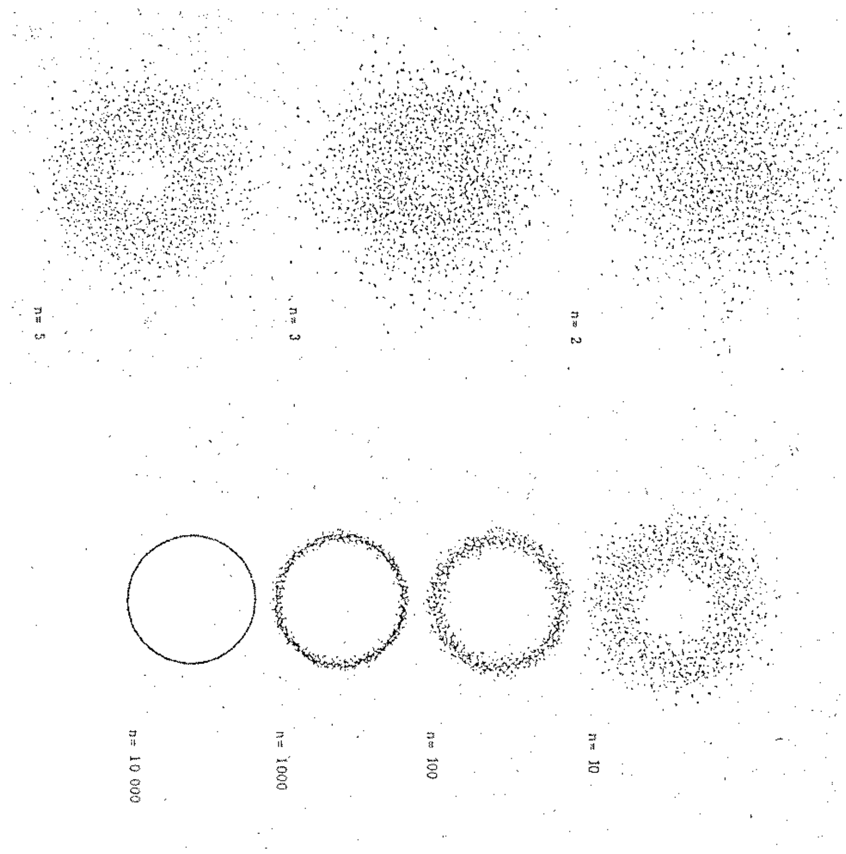
Random variable $S = \sqrt{S^2}$:
Length of the mutation vector $\vec{z}$.

After some math:

$$E(S) \approx \sigma\sqrt{n} \quad , \quad V(S) = \frac{1}{2}\sigma$$

- Variance $V(S)$ is independent of n.
- For large n: Offspring located on hypersphere of radius $E(S) \approx \sigma\sqrt{n}$.

The mutation vector (2)



Convergence velocity: Definition

Convergence velocity: Expectation of the distance towards the optimum covered per generation.

$$\varphi = E(\|\vec{x}^* - \vec{x}_t\| - \|\vec{x}^* - \vec{x}_{t+1}\|)$$

Alternatively:

$$\tilde{\varphi} = E(|f(\vec{x}^*) - f(\vec{x}_t)| - |f(\vec{x}^*) - f(\vec{x}_{t+1})|)$$

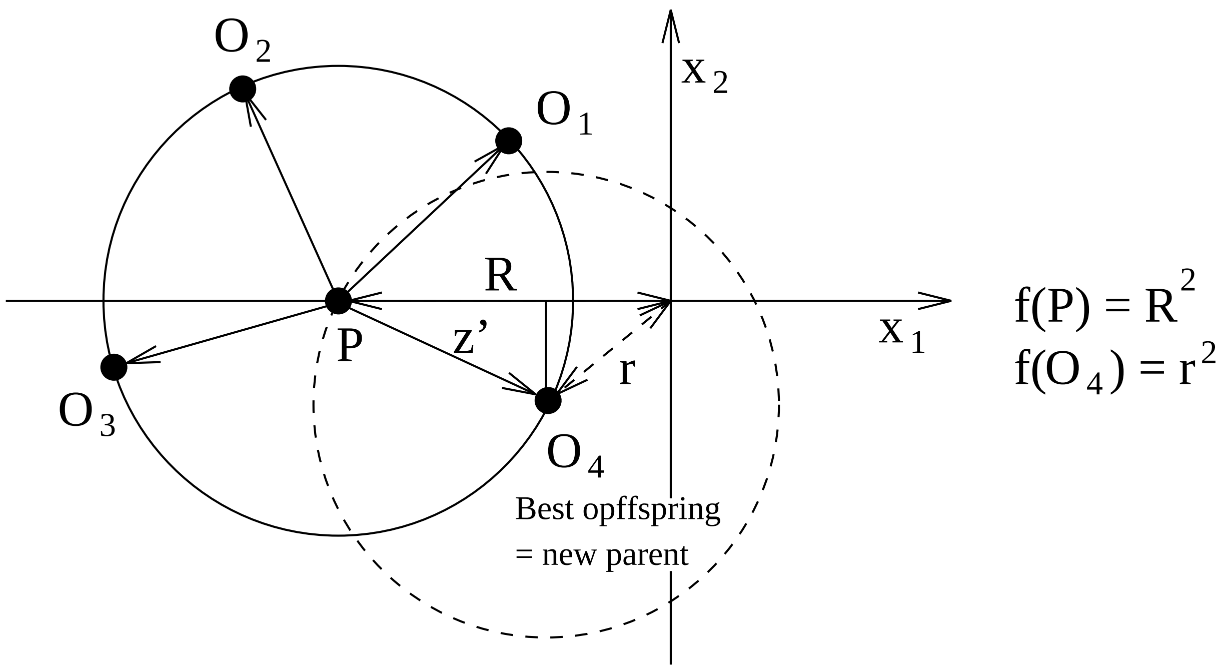
Convergence velocity of multi-membered ESs (1)

Simplifications:

- No self-adaption.
- One step-size.
- No recombination.
- $\mu = 1$

$\Rightarrow$ $(1 + \lambda)$-strategies, $(1, \lambda)$-strategies.

Convergence velocity of multi-membered ESs (2)



(1,4)-strategy, sphere model

Convergence velocity of multi-membered ESs (2)

Definition:

$Z_1, Z_2, \ldots, Z_\lambda$ i.i.d. random variables with p.d.f. $p(z)$.

$$Z_{1:\lambda} \leq Z_{2:\lambda} \leq \ldots \leq Z_{\lambda:\lambda}$$

is called order statistics of the Z_i. $p_{v:\lambda}(z)$ denotes the p.d.f. of $Z_{v:\lambda}$.

Idea:

Best offspring individual has

- smallest value of $r \Rightarrow r_{1:\lambda}$
- largest value of $z' \Rightarrow Z'_{\lambda:\lambda}$

Z': projection into direction of origin.

$$Z'_{v:\lambda} \sim N(0, \sigma)$$

$$Z_{v:\lambda} \sim N(0, 1)$$

Convergence velocity of multi-membered ESs (3)

$$\begin{aligned} \tilde{\varphi}_{(1\overset{+}{,}\lambda)} &= E(R^2 - r^2_{1:\lambda}) \\ r^2_{v:\lambda} &= l^2 + R^2 - 2R \cdot Z'_{\lambda-v+1:\lambda} \end{aligned}$$

Some math:

$$\begin{aligned} \tilde{\varphi}_{(1\overset{+}{,}\lambda)} &= E(2R \cdot Z'_{\lambda:\lambda} - \sigma^2 n) = E(2R\sigma \cdot Z_{\lambda:\lambda} - \sigma^2 n) \\ &= \int_{z_{min}}^{\infty} (2R\sigma \cdot z - \sigma^2 n) \cdot p_{\lambda:\lambda}(z)\, dz \\ &= 2R\sigma \int_{z_{min}}^{\infty} z \cdot p_{\lambda:\lambda}(z)\, dz - \sigma^2 n \int_{z_{min}}^{\infty} p_{\lambda:\lambda}(z)\, dz \end{aligned}$$

Convergence velocity of multi-membered ESs (4)

With:

$$p_{\lambda:\lambda}(z) = \lambda\, \phi(z)\, (\Phi(z))^{\lambda-1} = \frac{d}{dz}\, (\Phi(z))^{\lambda}$$

It follows that:

$$\tilde{\varphi}_{(1 \overset{+}{,} \lambda)} \;=\; 2R\sigma \int\limits_{z_{min}}^{\infty} z \cdot \frac{d}{dz}\, (\Phi(z))^{\lambda}\, dz - \sigma^2 n \int\limits_{z_{min}}^{\infty} \frac{d}{dz}\, (\Phi(z))^{\lambda}\, dz$$

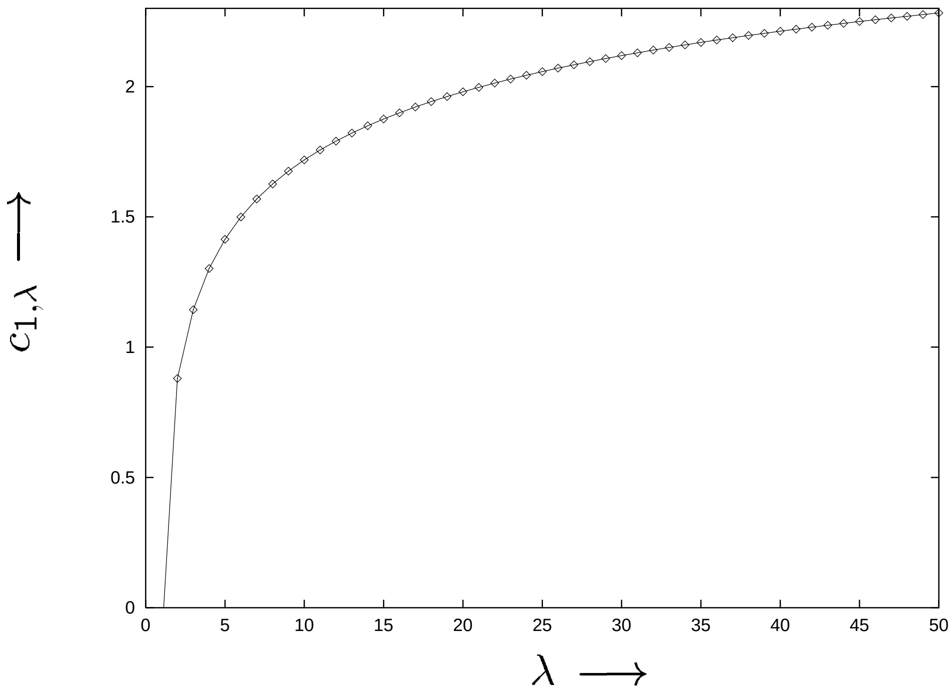
Convergence velocity of $(1, \lambda)$-ESs (1)

When accepting everything (non-elitist), $z_{min} = -\infty$.

$$\boxed{\tilde{\varphi}_{(1,\lambda)} = 2R\sigma \cdot c_{1,\lambda} - \sigma^2 n}$$

$$c_{1,\lambda} := E(Z_{\lambda:\lambda}) \begin{cases} \text{progress coefficent (Rechenberg)} \\ \text{selection intensity (Mühlenbein)} \end{cases}$$

Convergence velocity of $(1, \lambda)$-ESs (2)



The progress coefficient $c_{1,\lambda}$.

- Asymptotic behaviour: $c_{1,\lambda} \approx \sqrt{2\, ln\, \lambda}$.

Convergence velocity of $(1, \lambda)$-ESs (3)

Normalisation of $\tilde{\varphi}$, with $\varphi \approx \frac{\tilde{\varphi}}{2R}$, $\varphi' = \frac{\varphi_n}{R}$, $\sigma' = \frac{\sigma n}{R}$

$$\boxed{\varphi'_{1,\lambda} = c_{1,\lambda}\sigma' - \tfrac{1}{2}\sigma'^2}$$

- Optimal standard deviation:

$$\sigma'_{opt} = c_{1,\lambda}$$

- Maximum convergence velocity:

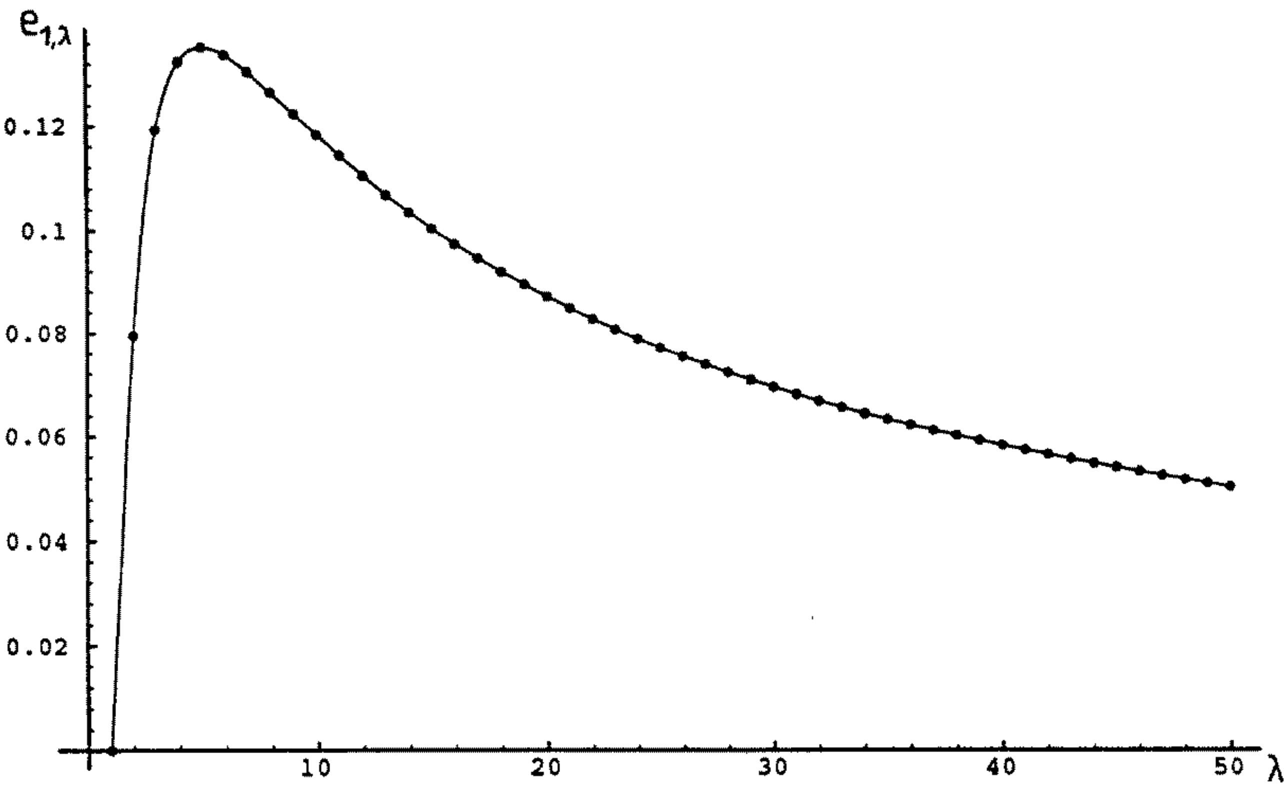
$$\varphi'_{max} = \frac{1}{2}c^2_{1,\lambda} \approx ln\,\lambda$$

Evolution condition: $\sigma' < 2c_{1,\lambda}$ (Guarantees $\varphi' > 0$).

Center for Applied Systems Analysis

Evolution efficiency

Maximum progress per individual: $e_{1,\lambda} = \varphi'_{max}/\lambda$



Convergence velocity of $(1+\lambda)$-ESs

- From $r \leq R$ it follows that

$$z_{min} = \frac{\sigma n}{2R}.$$

- Thus:

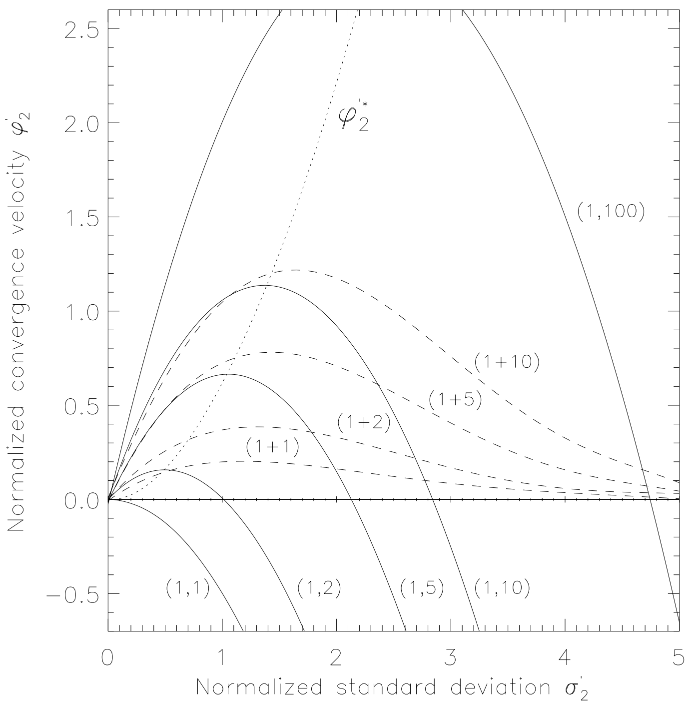
$$\varphi'_{(1+\lambda)} = \sigma' c_{1+\lambda}(\sigma') - \frac{{\sigma'}^2}{2}(1 - \Phi^{\lambda}(\frac{\sigma'}{z}))$$

- Where

$$c_{1+\lambda}(x) = \int_{\frac{x}{2}}^{\infty} z \, \frac{z}{dz} \, \Phi^{\lambda}(z) \, dz$$

No further analytical simplifications are possible.

Convergence velocity: illustration



Normalized convergence velocity φ' as a function of normalized standard deviation σ'.

Convergence velocity of (μ, λ)-ESs (1)

Simplifications:

- No self-adaptation.
- One step-size.
- Recombination:
  - center of mass recombination μ/μ_I (intermediary), or
  - global discete recombination μ/μ_D.

Convergence velocity of (μ, λ)-ESs (2)

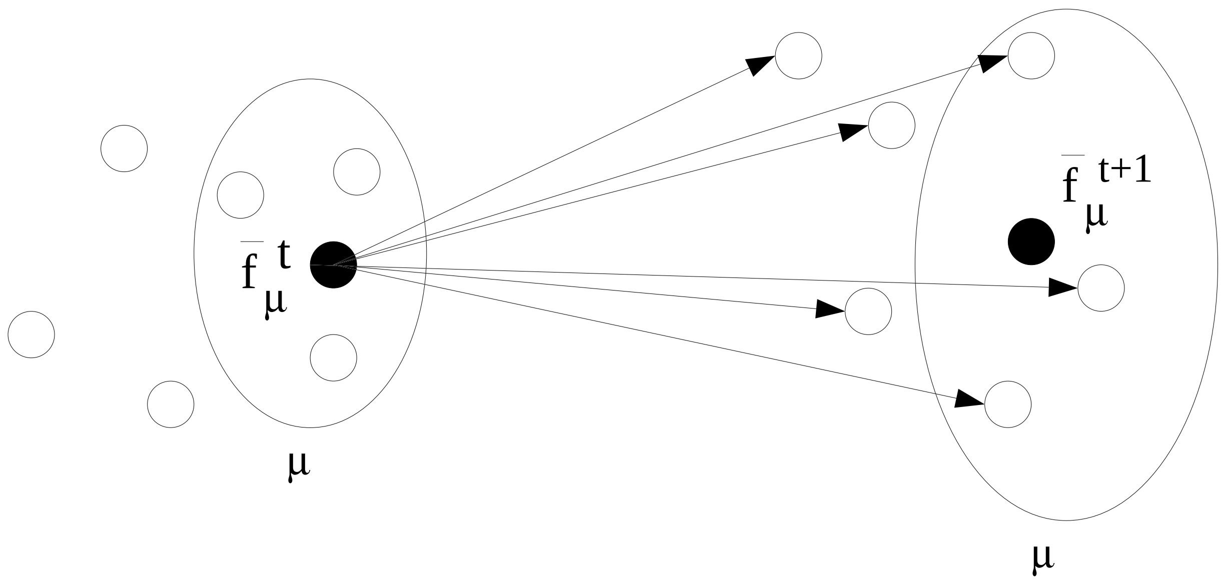


Illustration of center of mass recombination

.

Convergence velocity of (μ, λ)-ESs (3)

$$\begin{aligned} \varphi_{\mu,\lambda} &= \langle R \rangle - E(\langle \tilde{R} \rangle_{\mu,\lambda}) \\ &= \frac{1}{\mu} \sum_{v=1}^{\mu} R_v - \frac{1}{\mu} \sum_{v=1}^{\mu} r_{v:\lambda} \end{aligned}$$

Where:

- $\langle R \rangle$: Average distance to the optimum of parents.
- $\langle \tilde{R} \rangle_{\mu,\lambda}$: Average distance to the optimum of the μ best ofspring.

Center for Applied Systems Analysis

Convergence velocity of $(\mu/\mu_I, \lambda)$-ESs (1)

Without derivation (Rechenberg '94, Beyer '96):

$$\boxed{\varphi'_{\mu/\mu_I,\lambda} = c_{\mu,\lambda} \cdot \sigma' - \frac{\sigma'^2}{2\mu}}$$

(For $\sigma' \ll n, \mu^2 \ll n$)

- Optimal standard deviation:

$$\sigma'_{opt} = \mu \cdot c_{\mu,\lambda}$$

- Maximum convergence velocity:

$$\varphi'_{max} = \frac{1}{2}\mu \cdot c^2_{\mu,\lambda}$$

Center for Applied Systems Analysis

Convergence velocity of $(\mu/\mu_I, \lambda)$-ESs (2)

Progress coefficient ($Z_{v:\lambda} \sim N(0,1)$):

$$c_{\mu,\lambda} = \frac{1}{\mu} \sum_{v=\lambda-\mu+1}^{\lambda} E(Z_{v:\lambda}) \approx \frac{\lambda}{\mu} \cdot \phi(\Phi^{-1}(1 - \frac{\mu}{\lambda}))$$

$$\approx O\left(\sqrt{ln\frac{\lambda}{\mu}}\right)$$

Conjecture:

$$\boxed{\varphi'_{max} \approx \mu \cdot ln\frac{\lambda}{\mu}}$$

Convergence velocity of $(\mu/\mu_D, \lambda)$-ESs (1)

Without derivation (Rechenberg '94, Beyer '96):

$$\boxed{\varphi'_{(\mu/\mu_D,\lambda)} = \sqrt{\mu} \cdot c_{\mu,\lambda}\sigma' - \frac{\sigma'^2}{2} \text{ (For } \sigma' \ll n, \mu^2 \ll n)}$$

- Optimal standard deviation:

$$\sigma'_{opt} = \sqrt{\mu} \cdot c_{\mu,\lambda}$$

- Maximum convergence velocity:

$$\varphi'_{max} = \frac{1}{2}\mu \cdot c^2_{\mu,\lambda}$$

Again: $\varphi'_{max} \approx \mu \cdot ln\frac{\lambda}{\mu}$

Interpretation of results

- Genetic repair (Beyer '96):
  μ/μ_I-recombination decreases the harmful part of mutation.

- Incest taboo:
  μ/μ_I-recombination is only useful, if parents are different from each other.

- Implicit genetic repair:
  μ/μ_D-recombination estimates the center of mass corresponding to a species centered around the wild-type.

Conclusions

Summary:

- ESs are powerful search and optimization methods.
- Applicable e.g. to data analysis, fuzzy systems, neural networks etc.
- Self-adaptation is an important, distinguishing feature (learning of internal models).
- A powerful theory is available for ESs; focusing on convergence velocity and global convergence with probability one.
- Individuals can be seen as agents (especially in parallel spatial implementations).