**Evolution Strategies**

- Developed by Rechenberg, Schwefel – 1970’s
- Real-valued problems
- Strong emphasis on mutation for creating offspring
- Mutation = addition of normal random noise
- Self-adaptation of (mutation) parameters

**Evolution Strategies**

<table>
<thead>
<tr>
<th>Representation</th>
<th>Real-valued vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recombination</td>
<td>Discrete or intermediary</td>
</tr>
<tr>
<td>Mutation</td>
<td>Gaussian perturbation</td>
</tr>
<tr>
<td>Parent selection</td>
<td>Uniform random</td>
</tr>
<tr>
<td>Survivor selection</td>
<td>(μ, λ) or (μ+λ)</td>
</tr>
<tr>
<td>Specialty</td>
<td>Self-adaptation of mutation step sizes</td>
</tr>
</tbody>
</table>

**Early Evolution Strategies**

- Minimise \( f : \mathbb{R}^n \rightarrow \mathbb{R} \)
- Algorithm: “two-membered ES” using
  - Chromosomes are \( \mathbb{R}^n \) Vectors
  - Population size 1
  - Only mutation creating one child
  - Greedy selection

**Early Evolution Strategies**

BEGIN
set \( f = 0 \);
Create an initial point \( \{x_1, \ldots, x_n\} \in \mathbb{R}^n \);
REPEAT until (TERMINATION CONDITION is satisfied) DO
  draw \( z_i \) from a normal distr., for all \( i \in \{1, \ldots, n\} \) independently;
  \( y_i = x_i + z_i \), for all \( i \in \{1, \ldots, n\} \);
  IF \( f(y_i) \leq f(x_i) \) THEN
    \( y = y_i \);
  ELSE
    \( y = x_i \);
  FI
set \( f = f + 1 \);
END

**Early Evolution Strategies**

- \( z \) values drawn from normal distribution \( \mathcal{N}(0, \sigma) \)
  - \( \sigma \) is called mutation step size
- The ratio of successful mutations should be 1/5
- This rule resets \( \sigma \) after every k iterations by
  - \( \sigma = \sigma / c \) if \( p_s > 1/5 \)
  - \( \sigma = \sigma \) if \( p_s < 1/5 \)
  - \( \sigma = 1 \) if \( p_s = 1/5 \)
- where \( p_s \) is the \% of successful mutations, \( 0.817 \leq c \leq 1 \)
Illustration of normal distribution

Representation

- Chromosomes consist of three parts:
  - Object variables: \( x_1, \ldots, x_n \)
  - Strategy parameters:
    - Mutation step sizes: \( \sigma_1, \ldots, \sigma_n \)
    - Rotation angles: \( \alpha_1, \ldots, \alpha_k \)
  - Covariance matrix \( \{ \sigma_1, \ldots, \sigma_n \} \)
- Angles not always present
- Full size: \( < x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_k > = x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_k > \)
  - where \( k = n(n-1)/2 \)

Mutation

- Main mechanism: changing value by adding random noise drawn from normal distribution
- \( x_i' = x_i + N(0, \sigma) \)
- Key idea:
  - \( \sigma \) is part of the chromosome \( < x_1, \ldots, x_n, \sigma > \)
  - \( \sigma \) is also mutated into \( \sigma' \)
  - Thus: mutation step size \( \sigma \) is coevolving with the solution \( x \)

Mutation

- Net mutation effect: \( < x, \sigma > \rightarrow < x', \sigma' > \)
- Order is important:
  - first \( \sigma \rightarrow \sigma' \)
  - then \( x \rightarrow x' = x + N(0, \sigma') \)
- new \( < x', \sigma' > \) is "evaluated" twice
  - First: \( x' \) is good if \( f(x') \) is good
  - Second: \( \sigma' \) is good if the \( x' \) it created is good
- Reversing mutation order: the \( \sigma \) that created \( x' \) would have disappeared

Mutation

- Coevolution of \( x \) and \( \sigma \)
- Evolution of \( \sigma \) can be seen in different dimensions:
  - Time: different \( \sigma \)s are needed at different times
  - Space: different \( \sigma \)s are needed at different locations

Mutation 1: one \( \sigma \)

- Chromosomes: \( < x_1, \ldots, x_n, \sigma > \)
- \( \sigma' = \sigma e^{\tau N(0,1)} \)
- \( x_i' = x_i + N(0, \sigma') \)
- Learning rate: \( \tau = \frac{1}{\sqrt{n}} \)
- Boundary rule \( \sigma < \varepsilon_0 \rightarrow \sigma = \varepsilon_0 \)
Mutation 1: one $\sigma$

- Circle: all components have same probability distribution

Mutation 2: $n$ $\sigma$’s

- Chromosomes: $<x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_n>$
- $\sigma_i' = \sigma_i e^{\tau N(0,1)} + \tau N(0,1)$
- $x_i' = x_i + \sigma_i' N_i(0,1)$
- Two learning rate parameters:
  - $\tau = \frac{1}{\sqrt{2\pi}}$
  - $\tau = \frac{1}{\sqrt{2\sqrt{\pi}}}$
- And $\sigma_i' < \epsilon \Rightarrow \sigma_i' = \epsilon$

Mutation case 3: Correlated mutations

- Chromosomes: $<x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_n, \alpha_1, \ldots, \alpha_k>$
- 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
  - $c_{ij} = \frac{\beta}{2} \cdot (\sigma_i^2 - \sigma_j^2) \cdot \tan(2\alpha_{ij})$ if $i \neq j$ are correlated

The mutation mechanism is then:

- $\sigma_i' = \sigma_i e^{\tau N(0,1)} + \tau N(0,1)$
- $\alpha_j' = \alpha_j + \beta \cdot N_j(0,1)$
- $x' = x + N(0,C')$ << multi-normal distribution
  - $x$ stands for the vector $<x_1, \ldots, x_n>$
  - $C'$ is the covariance matrix $C$ after mutation of the $\sigma$ values
- $\tau$ and $\tau'$ as before, $\beta = \frac{5}{4}$
- $\alpha_i' < \epsilon \Rightarrow \alpha_i' = \epsilon$
- $|\alpha_i'| > \epsilon \Rightarrow \alpha_i' = \epsilon \cdot \sign(\alpha_i')$, i.e., keep it in $[-\pi, \pi]$
Correlated mutations cont’d

Is it worth it?
Imagine the ridges in the Mexican Hat function (see Benchmarks.nb).
Is it not a good idea to be able to increase the probability to generate individuals along the ridge, while decreasing it on the slopes?

Recombination

• Creates one child
• Acts per variable / position:
  - Averaging parental values, or
  - Selecting one of the parental values
• From two or more parents:
  - Using two selected parents to make a child
  - Selecting two parents for each position anew

Names of Recombinations

<table>
<thead>
<tr>
<th>Two fixed parents</th>
<th>Two parents selected for each i</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z_i = \frac{x_i + y_i}{2} )</td>
<td>Local intermediary</td>
</tr>
<tr>
<td>( z_i ) is ( x_i ) or ( y_i ), chosen randomly</td>
<td>Local discrete</td>
</tr>
</tbody>
</table>

Parent selection

• Parents are selected by uniform random distribution whenever an operator needs one/some
• Thus: ES parent selection is unbiased - every individual has the same probability to be selected
• Note that in ES “parent” means a population member (in GA’s: a population member selected to undergo variation)

Survivor Selection

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

Survivor Selection

• \( (\mu + \lambda) \)-selection is an elitist strategy
• \( (\mu, \lambda) \)-selection can “forget”
• Often \( (\mu, \lambda) \)-selection is preferred for:
  - Better at leaving local optima
  - Better at following moving optima
  - Using the + strategy bad \( \sigma \) values can survive in \( <x, \sigma> \) too long if their host \( x \) is very fit
Survivor Selection

- Selective pressure in ES is very high
  \((\lambda / \mu = 7\) recommended)
- ES: \(\mu = 15, \lambda = 100\)
  \[\tau = \frac{\ln \lambda}{\ln \mu} \approx \frac{\ln 100}{\ln 15} \approx 7\] takeover time
- GA: \(\mu = 15, \lambda = 100\)
  \[\tau = \lambda \ln \lambda = 100 \ln 100 = 460\]

Self-adaptation illustrated

- Given a dynamically changing fitness landscape
  (optimum location shifted every 200 generations)
- Self-adaptive ES is able to
  - follow the optimum
  - 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

- \(\mu > 1\) to carry different strategies
- \(\lambda > \mu\) to generate offspring surplus
- Not “too” strong selection, e.g., \(\mu / \lambda = 7\)
- \((\mu, \lambda)\)-selection to get rid of misadapted \(\sigma\)'s
- Mixing strategy parameters by (intermediary) recombination on them

Exercise

- Download an EC framework
- See references in the course web page - choose the language of your preference.
- Using ES, minimize Ackley’s function, first for \(n=2\), then for \(n=30\).
- Compare your results with those mentioned in Eiben’s book.
- For the 2D case, trace the \(\sigma\)'s of the best individual of each generation; plot contour lines of the function, and the location of the best individual of each generation, drawn as an ellipse whose radii are proportional to its \(\sigma\)'s.