Evolutionary Strategies


ES: Overview

- Developed: Germany in the 1970's
- Early names: I. Rechenberg, H.-P. Schwefel
- Typically applied to:
  - numerical optimization
- ES features:
  - fast
  - good optimizer for real-valued optimization models
  - relatively well researched
- Special feature:
  - self-adaptation of (mutation) parameters is standard

ES Main Features

<table>
<thead>
<tr>
<th>Representation</th>
<th>Real-valued vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recombination</td>
<td>Discrete or intermediate</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>$(\mu, \lambda)$ or $(\mu + \lambda)$</td>
</tr>
<tr>
<td>Special feature</td>
<td>Self-adaptation of mutation step size</td>
</tr>
</tbody>
</table>

Recombination in ES

- Two parents create one child
- Discrete recombination – one of the parent alleles is chosen with equal chance of either parent
  - $z_i = x_i$ or $y_i$ randomly selected
- Intermediate recombination – the values of the parent alleles are averaged
  - $z_i = (x_i + y_i)/2$
Offspring (Survivor) Selection

- \((\mu, \lambda)\) selection, where \(\mu\) parents breed \(\lambda\) offspring, out of which the fittest \(\mu\) are used as parents for the next generation.

- \((\mu + \lambda)\) selection, where offspring and parents participate in the selection.

ES vs EP

**Evolutionary Strategy (ES) vs Evolutionary Programming (EP)**

<table>
<thead>
<tr>
<th>Representation</th>
<th>Real-valued vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recombination</td>
<td>Discrete or intermediate</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>((\mu, \lambda)) or ((\mu + \lambda))*</td>
</tr>
<tr>
<td>Special feature</td>
<td>Self-adaptation of mutation step sizes</td>
</tr>
</tbody>
</table>

**Evolutionary Programming (EP)**

<table>
<thead>
<tr>
<th>Representation</th>
<th>Real-valued vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recombination</td>
<td>None</td>
</tr>
<tr>
<td>Mutation</td>
<td>Gaussian perturbation*</td>
</tr>
<tr>
<td>Parent selection</td>
<td>Deterministic</td>
</tr>
<tr>
<td>Survivor selection</td>
<td>Probabilistic ((\mu + \mu))</td>
</tr>
<tr>
<td>Special feature</td>
<td>Self-adaptation of mutation step sizes (in meta-EP)</td>
</tr>
</tbody>
</table>

*Deterministic based on the rank order rather than fitness function

Example

- Problem: Min \(f : \mathbb{R}^n \rightarrow \mathbb{R}\)

- Algorithm: “two-membered ES” using
  - Vectors from \(\mathbb{R}^n\) directly as chromosomes
  - Population size 1
  - Only mutation creating one child
  - Greedy selection

Example: Pseudocode

Minimization Problem

Set \(t = 0\)
Create initial solution \(x^t = (x_1^t, \ldots, x_n^t)\)
REPEAT UNTIL \((\text{TERMIN.COND})\) satisfied DO
- Draw \(z_i\) from a normal distribution for all \(i = 1, \ldots, n\)
- \(y_i^t = x_i^t + z_i\)
- IF \(f(x^t) < f(y^t)\) THEN \(x^{t+1} = y^t\)
  ELSE \(x^{t+1} = x^t\)
FI
Set \(t = t + 1\)
OD
END
Example: Mutation Mechanism

- $z$ values drawn from normal distribution $N(\xi, \sigma)$
  - mean $\xi$ is set to 0
  - variance $\sigma$ is called mutation step size
- $\sigma$ is varied on the fly by the "1/5 success rule":
  - This rule resets $\sigma$ after every $k$ iterations by
    - $\sigma = \frac{\sigma}{c}$ if $p_s > 1/5$
    - $\sigma = \sigma \cdot c$ if $p_s < 1/5$
    - $\sigma = \sigma$ if $p_s = 1/5$
- where $p_s$ is the ratio of successful mutations to all mutations, $0.8 \leq c \leq 1$

$c$ = parameter

1/5 Success Rule

Note: Nowadays self adaptation rather than 1/5 success rule is used

1. The ratio of successful mutations to all mutations should be 1/5.
2. If the ratio is larger than 1/5, then the standard deviation should increase $(0.8 \leq c \leq 1)$.
3. If the is smaller than 1/5, then the standard deviation should decrease.

Successful mutation = Child is fitter than a parent

The 1/5 selection rule increases the solution efficiency (speeds up convergence)

Normal Distribution

http://mathworld.wolfram.com/NormalDistribution.html

Example: The Jet Nozzle Design

Problem: Optimize the shape of a jet nozzle
Approach: Random mutations to shape + selection
Genetic Operators: Mutation

The one dimensional case

<table>
<thead>
<tr>
<th>Chromosomes consist of three parts:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Variables: x₁, ..., xₙ</td>
</tr>
<tr>
<td>- Strategy parameters:</td>
</tr>
<tr>
<td>- Mutation step sizes: σ₁, ..., σₙ</td>
</tr>
<tr>
<td>- Rotation angles: α₁, ..., αₖ</td>
</tr>
<tr>
<td>- Not every component is always present</td>
</tr>
<tr>
<td>- Full size: ⟨ x₁, ..., xₙ, σ₁, ..., σₙ, α₁, ..., αₖ ⟩</td>
</tr>
<tr>
<td>where k = (n – m/2)(m - 1)</td>
</tr>
</tbody>
</table>

Representation

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 \( ⟨ x₁, ..., xₙ, \sigma ⟩ \)
  - \( \sigma \) is also mutated into \( \sigma' \) (presented later)
- Thus: mutation step size \( \sigma \) is coevolving with the solution \( x \)

Mutate \( \sigma \) First

- Net mutation effect: \( ( x, \sigma ) \rightarrow ( x', \sigma' ) \)
- Order is important:
  - first \( \sigma \rightarrow \sigma' \) (presented later)
  - then \( x \rightarrow x' = x + N(0, \sigma') \)
- Reversing mutation order this would not work
Mutation Case 1: Uncorrelated Mutation with One $\sigma$

- 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” $\tau \propto 1/\sqrt{n}$
- And we have a boundary rule $\sigma' < \varepsilon_0 \Rightarrow \sigma' = \varepsilon_0$

Mutation Case 2: Uncorrelated Mutation with $n$ $\sigma$’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:
  - $\tau'$ overall learning rate
  - $\tau$ coordinate-wise learning rate
- $\tau' \propto 1/(2n)^{\frac{1}{3}}$ and $\tau \propto 1/(2n^{\frac{1}{3}})^{\frac{1}{3}}$
- $\sigma'_i < \varepsilon_0 \Rightarrow \sigma'_i = \varepsilon_0$
Mutation Case 3: Correlated Mutation

- Chromosomes: \( \langle x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_n, \alpha_1, \ldots, \alpha_k \rangle \), where \( k = n \cdot \frac{(n-1)}{2} \)
- 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

Correlated Mutation cont’d

The mutation mechanism is then:
- \( \sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N(0,1)) \)
- \( \alpha_j' = \alpha_j + \beta \cdot N(0,1) \)
- \( x' = x + N(0,C') \)
  - \( x \) denotes the vector \( \langle x_1, \ldots, x_n \rangle \)
  - \( C' \) is the covariance matrix \( C \) after mutation of the \( \alpha \) values
- \( \tau' \approx \frac{1}{(2n)^{1/2}} \) and \( \tau \approx \frac{1}{(2n)^{1/2}} \) and \( \beta \approx 5^\circ \)
- \( \sigma_i' < \epsilon_0 \Rightarrow \sigma_i' = \epsilon_0 \) and
- \( |\alpha_{ij}'| > \pi \Rightarrow \alpha_{ij}' = \alpha_{ij}' - 2\pi \cdot \text{sign}(\alpha_{ij}') \)

Mutants with Equal Likelihood

Ellipse: Mutants having uneven chance of reaching optimum

Recombination Types

<table>
<thead>
<tr>
<th>Two fixed parents</th>
<th>Two parents selected for each ( i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z_i = (x_i + y_i)/2 )</td>
<td>Local intermediate \rightarrow Global intermediate</td>
</tr>
<tr>
<td>( z ) is ( x_i ) or ( y_i ) chosen randomly</td>
<td>Local discrete \rightarrow Global 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 undergoes 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 cont'd

- $(\mu + \lambda)$-selection is an elitist strategy.
- $(\mu, \lambda)$-selection can "forget".
- Often $(\mu, \lambda)$-selection is preferred for:
  - Better in leaving local optima
  - Better in following moving optima
  - With the $(\mu + \lambda)$-selection strategy bad $\sigma$ values can survive in $(x,\sigma)$ too long, if their host $x$ is very fit.
- Selection pressure in ES is very high ($\lambda = 7 \cdot \mu$ is the common setting).

### Self-adaptation Illustrated

- Given a dynamically changing fitness landscape (optimum location shifted every 200 generations).
- 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

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

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 $\sigma$ values used with equal chance
- Selection: (1,8) strategy

Example Application:
The 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
Example Application: 
The Ackley Function (Bäck et al. '93)

- The Ackley function (here used with \( n = 30 \)):
  \[
  f(x) = -20 \cdot \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + \varepsilon
  \]

- Evolution strategy:
  - Representation:
    - \(-30 < x_i < 30\) (coincidence of 30's!)
    - 30 step sizes
  - (30,200) selection
  - Termination: after 200000 fitness evaluations
  - Results: average best solution is \(7.48 \cdot 10^{-8}\) (very good)