# Computational Intelligence

Unit # 8



## Canonical Evolutionary Programming

- For I = 1 to m Do:
  - Use parent I to produce 1 child and add it to the offspring population
- End Do
- From the combined pool of 2m parents and children, select only the m individuals with highest fitness to survive.

# Canonical EP (Cont'd)

- Producing a very small number of offspring from a parent is not a very reliable sample of its potential for producing useful progeny.
- Computationally, it is quite easy to extend the EP paradigm to encompass this idea by just allowing the size of the offspring population n to be greater than m, the size of the parent population.



# Canonical Evolution Strategy

- Having done so raises two interesting questions:
  – How many offspring should each parent produce, and given that we are now using parents more efficiently,
  – Can we reduce the size of the parent population
- These ideas correspond directly to the early Evolutionary Strategy algorithms.
- Perhaps most striking about this early work was the focus on $(1 + \lambda)$-ES models in which the entire next generation was produced from a single parent.

# EP Differences

- EP emphasizes phenotypic evolution, instead of genotypic evolution.
- EP does not make use of any recombination operator. There is no exchange of genetic material.
- Selection is based on competition. Those individuals that perform best against a group of competitors have a higher probability of being included in the next generation.
- Parents and offspring compete for survival.
- The behavior of individuals is influenced by strategy parameters, which determine the amount of variation between parents and offspring.



# EP - Mutation

- As mutation is the only means of introducing variation in an EP population, it is very important that the design of a mutation operator considers the exploration–exploitation trade-off.
- The variation process should facilitate exploration in the early stages of the search to ensure that as much of the search space is covered as possible.
- After an initial exploration phase, individuals should be allowed to exploit obtained information about the search space to fine tune solutions.

## EP – Mutation (Cont'd)

- In general, mutation is defined as

$$x'_i(t) = x_i(t) + \Delta x_i(t)$$

- where $x'_i(t)$ is the offspring created from parent $x_i(t)$ by adding a step size $\Delta x_i(t)$ to the parent.
- The step size is noise sampled from some probability distribution, where the deviation of the noise is determined by a strategy parameter, $\sigma_i$ .
- Generally, the step size is calculated as

$$\Delta x_i(t) = \Phi(\sigma_i(t))\eta_i(t)$$

- where $\Phi : R \rightarrow R$ is a function that scales the contribution of the noise, $\eta_i(t)$.



## EP – Scaling Functions

- Based on the characteristics of the scaling function, Φ, EP algorithms can be grouped into three main categories of algorithms:
- ***non-adaptive EP,*** in which case $\Phi(\sigma) = \sigma$. In other words, the deviations in step sizes remain static.
- ***dynamic EP,*** where the deviations in step sizes change over time using some deterministic function, Φ.
- ***self-adaptive EP,*** in which case deviations in step sizes change dynamically.

# ES - Recombination

- The basic recombination scheme in evolution strategies involves two parents that create one child.
- To obtain $\lambda$ offspring recombination is performed $\lambda$ 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.



# ES – Step-Size Adjustment

- 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 to all mutations should be 1/5.
- If the ratio is greater then 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.

## ES - Survival Selection

- After creating $\lambda$ offspring and calculating their fitness, the best $\mu$ of them are chosen deterministically, either from the offspring only, called $(\mu, \lambda)$ selection, or from the union of the parents and offspring, called $(\mu + \lambda)$ selection.
- Both schemes are strictly deterministic and are based on rank rather than an absolute fitness value.



## ES - Survival Selection (Cont'd)

- $(\mu, \lambda)$ is typically preferred over $(\mu + \lambda)$ for the following reasons
  - The $(\mu, \lambda)$ discards all parents and is there in principle able to leave (small) local optima, so it is advantageous in the case of multimodal 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 because mis-adapted parameter may survive for a relatively large number of generations

# Summary EP

| | |
|---|---|
| Representation | Real-valued vectors |
| Parent Selection | Deterministic (each parent creates one offspring via mutation) |
| Recombination | None |
| Mutation | Gaussian perturbation |
| Survival Selection | Probabilistic (mu + mu) |



# Summary of ES

| | |
|---|---|
| Representation | Real-valued vectors |
| Parent Selection | Uniform random |
| Recombination | Discrete or intermediary |
| Mutation | Gaussian perturbation |
| Survival Selection | (mu, lambda) or (mu + lambda) |