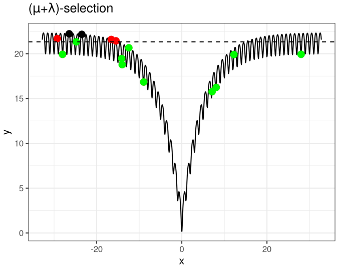
# Optimization in Machine Learning

# Evolutionary Algorithms
# ES / Numerical Encodings




**Learning goals**

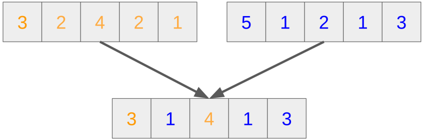
- Recombination
- Mutation
- A few simple examples

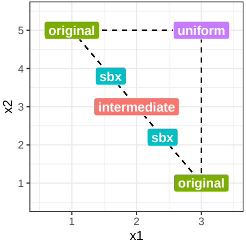
# RECOMBINATION FOR NUMERIC

Options for recombination of two individuals $\mathbf{x}, \tilde{\mathbf{x}} \in \mathbb{R}^d$:

- **Uniform crossover**: Choose gene $j$ of parent 1 with probability $p$ and of parent 2 with probability $1 - p$
- **Intermediate recombination**: Offspring is created from mean of two parents: $\frac{1}{2}(\mathbf{x} + \tilde{\mathbf{x}})$
- **Simulated Binary Crossover (SBX)**: generate **two offspring**

$$\bar{\mathbf{x}} \pm \frac{1}{2}\beta(\tilde{\mathbf{x}} - \mathbf{x}),\ \bar{\mathbf{x}} = \frac{1}{2}(\mathbf{x} + \tilde{\mathbf{x}}),\ \beta \in [0, 1] \text{ uniformly at random}$$



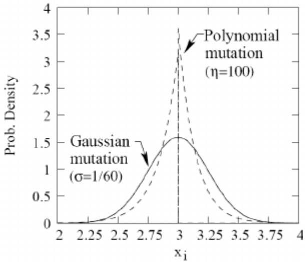



# MUTATION FOR NUMERIC

**Mutation:** Individuals get modified

**Example** for $\mathbf{x} \in \mathbb{R}^d$:

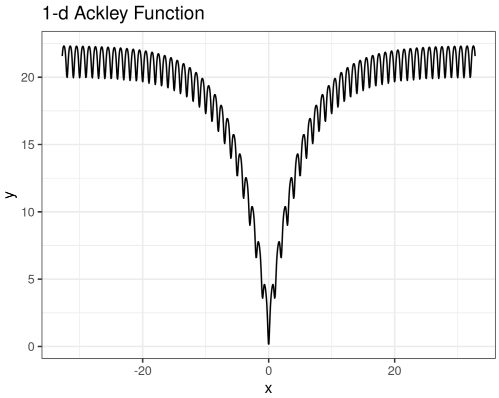
- **Uniform mutation:** Select random gene $x_j$ and replace it by uniformly distributed value (within feasible range).
- **Gauss mutation:** $\mathbf{x} \pm \mathcal{N}(0, \sigma\mathbf{I})$
- **Polynomial mutation:** Use a different distribution instead of normal distribution




**Source:** K. Deb, D. Deb. Analysing mutation schemes for real-parameter genetic algorithms, 2014

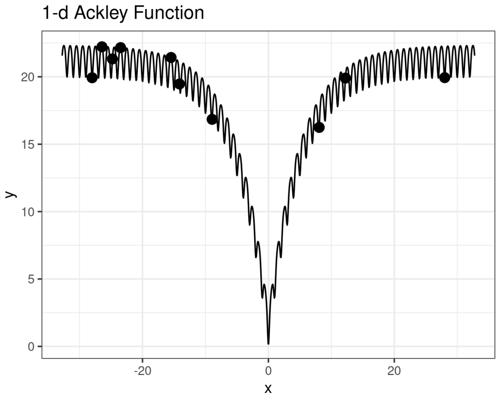
# EXAMPLE OF AN EVOLUTIONARY ALGORITHM

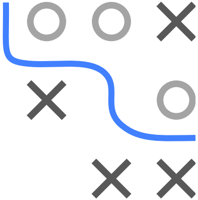
(Simple) EA on 1-dim Ackley function on $[-30, 30]$. Usually, for optimizing a function $f : \mathbb{R}^d \to \mathbb{R}$, individuals are encoded as real vectors $\mathbf{x} \in \mathbb{R}^d$.



# EXAMPLE OF AN EVOLUTIONARY ALGORITHM

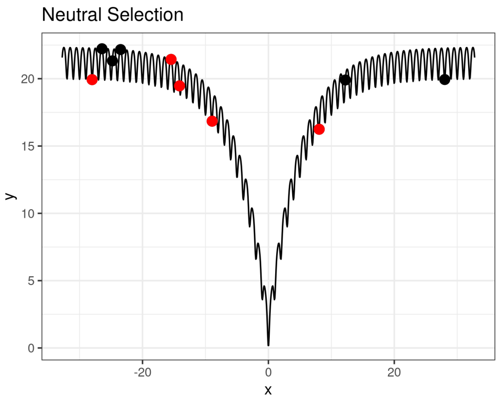
Random initial population with size $\mu = 10$

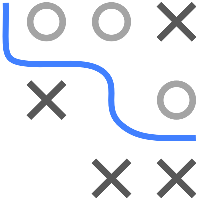


# EXAMPLE 1: ACKLEY FUNCTION

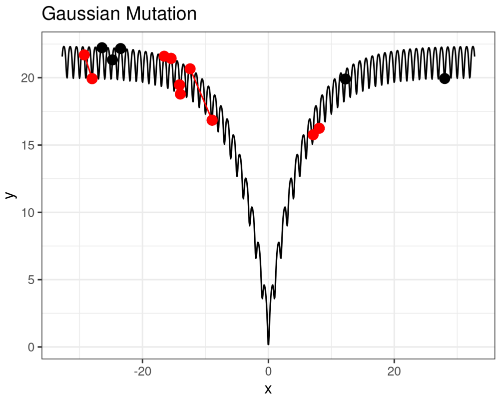
We choose $\lambda = 5$ offsprings by neutral selection (red individuals).



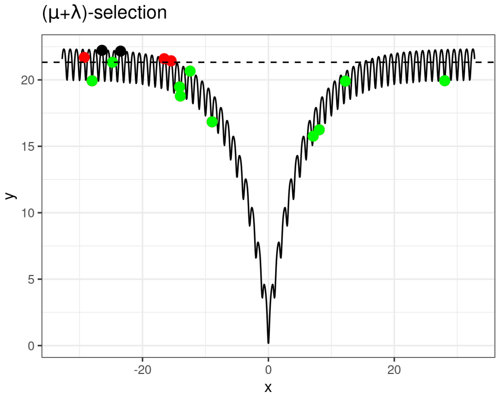
# EXAMPLE 1: ACKLEY FUNCTION

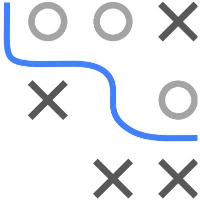
Use Gaussian mutation with $\sigma = 2$, but without recombination.



# EXAMPLE 1: ACKLEY FUNCTION

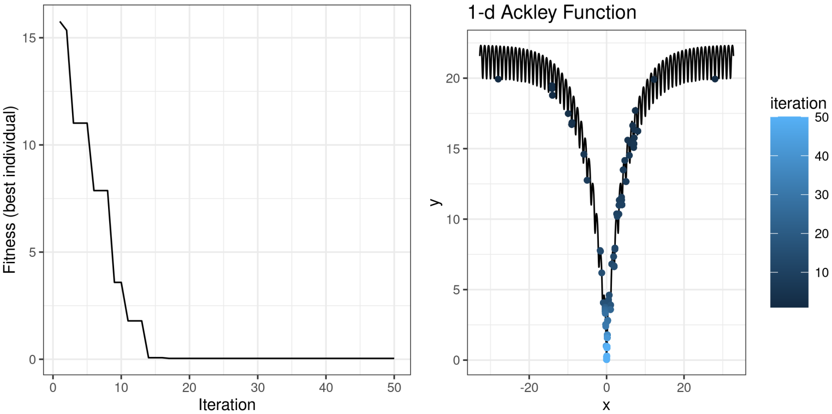
Use $(\mu + \lambda)$ selection. Selected individuals are marked in green.



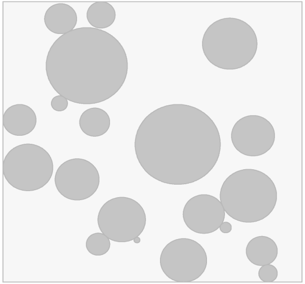
# EXAMPLE 1: ACKLEY FUNCTION

After 50 iterations:



# EXAMPLE 2: GRID OF BALLS

Consider a grid in which *n* balls with random radius are placed.

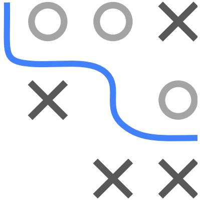


**Aim:** Find the circle with the largest possible radius in the grid that does **not** intersect with the other existing circles.

- What is the fitness function?
- How is the population defined?

Implementation: `https://juliambr.shinyapps.io/balls/`

# EXAMPLE 2: GRID OF BALLS / 2



In our example, the chromosome of an individual is the center of a circle, so the chromosomes are encoded as 2-dimensional real vectors $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$.

The population $P \subset \mathbb{R}^2$ is given as a set of circle centers.

The fitness function evaluates an individual $\mathbf{x} \in P$ based on the distance to the nearest neighboring gray circle $k$.

$$f(\mathbf{x}) = \min_{k \in \text{Grid}} \text{distance}(k, \mathbf{x}),$$

where the distance is defined as 0 if a circle center is within the radius of a circle of the grid.
This function is to be maximized: we are looking for the largest circle that does not touch any of the gray circles.