AI LAB 2025

AN INTRODUCTION TO ARTIFICIAL INTELLIGENCE
SHAY BUSHINSKY, SPRING 2025

Genetic Algorithms:

Selection & Scaling

IN THIS LECTURE

1. Genetic Algorithm Selection & Scaling

SELCTION IN GA

- Key to choosing parents & survivors
- Balances selection pressure vs. genetic diversity
- Drives exploration and exploitation
- Prevents premature convergence

THE SELECTION PROCESS & ITS FACTORS

- High-quality individuals favored for mating
- Diversity maintained to explore solution space
- Two phases: Parent and survivor selection
- Measured via selection pressure & fitness variance

RAW VS PROPORTIONAL FITNESS

Raw Fitness

- ➤ The original, unmodified evaluation of an individual's quality.
- ➤ Can be noisy, skewed, or have a wide range.
- ➤ Example: Total profit, accuracy score, time to complete task.

Proportional Fitness

- ➤ Normalized value:

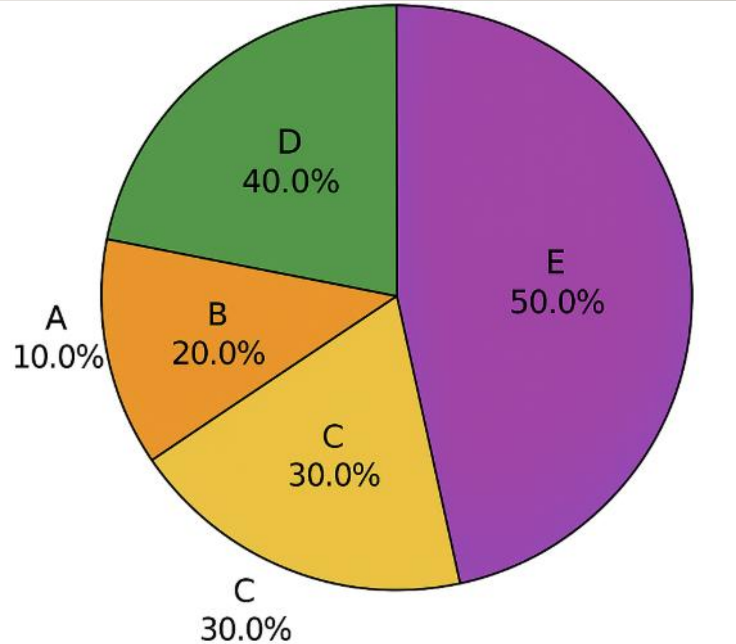
$$\text{Proportional fitness} = \frac{f_i}{\sum f}$$

- ➤ Used in **selection methods** like RWS and SUS.
- ➤ Represents **selection probability** (i.e. how big a slice on the wheel).

- **Key Difference**
  Raw fitness ranks quality.
  Proportional fitness determines **chance of being selected**

FITNESS PROPORTIONAL SELECTION METHODS

- Roulette Wheel Selection (RWS): multiple spins
- Selection chance proportional to fitness
- Risk: "Superhero" domination & loss of pressure
- Basis for other FPS methods

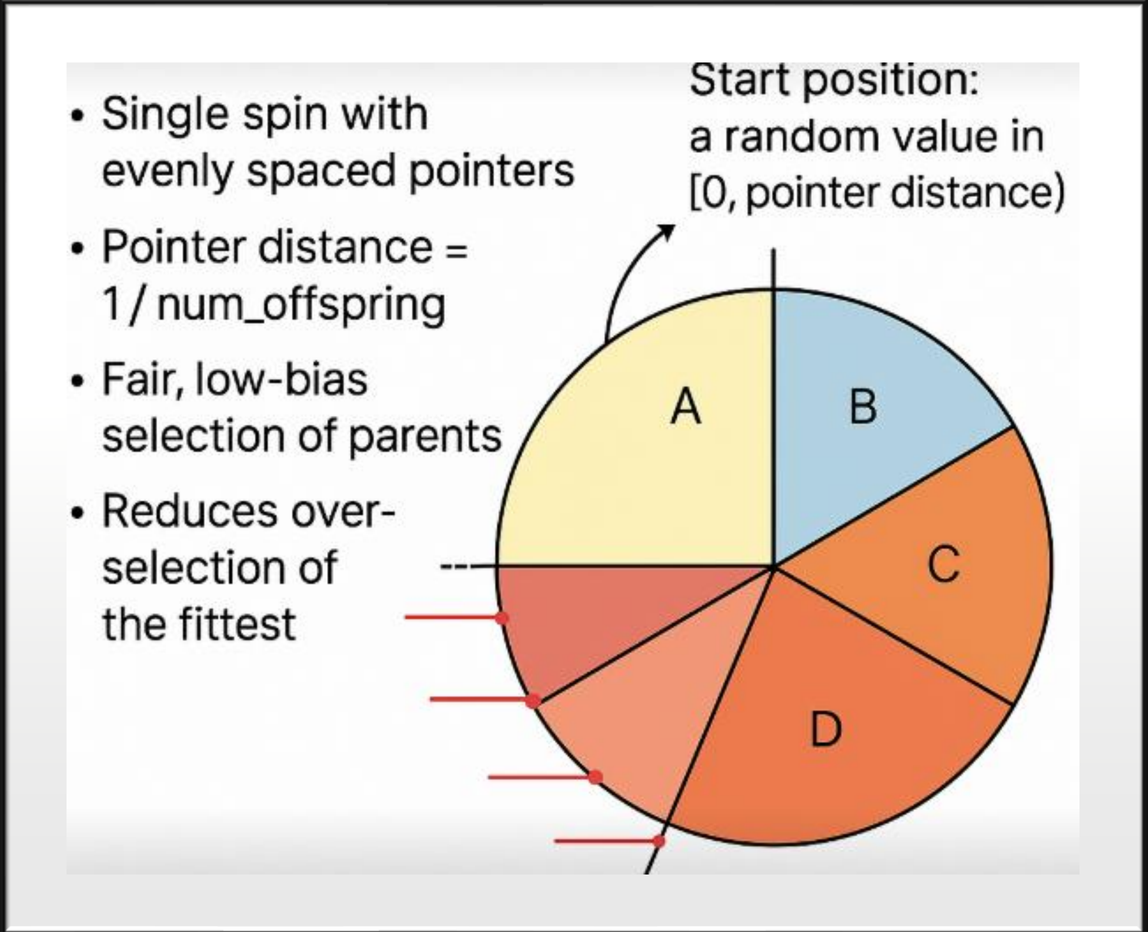


ROULETTE WHEEL SELECTION (RWS)

- **Basic Idea:**
  Each individual's chance of being selected is proportional to its (possibly scaled) fitness value. Imagine a roulette wheel where each slice's size corresponds to the individual's fitness.
- **Selection Process:**
  - **Compute Total Fitness:** Sum the (scaled) fitness values of all individuals.
  - **Assign Probabilities:** Each individual gets a probability equal to its fitness divided by the total fitness.
  - **Spin the Wheel:** Generate a random number between 0 and 1. The individual whose cumulative probability exceeds this random number is selected.
  - **Repeat:** For each selection, the process repeats, potentially with replacement.

SUS ALGORITHM EXPLANATION

- Compute cumulative, normalized fitness
- Calculate pointer_distance = 1/num_offspring
- Each pointer: start + i × pointer_distance
- Selects individuals corresponding to pointers
- Provides equitable parent selection for crossover



FITNESS SCALING METHODS

- Adjust raw fitness to control selection pressure
- Techniques: Linear, exponential, Boltzmann, Sigma
- Prevents dominance of super-fit individuals
- Encourages early exploration and later exploitation
- Tailor scaling to dynamic population fitness

LINEAR FITNESS SCALING

- A general linear scaling transformation is defined as:

$$f'(x) = a \cdot f(x) + b$$

where:

- f(x) is the original fitness.
- f′(x) is the scaled fitness.
- a and b are constants chosen to adjust the range of fitness values.

Example Adjustments:

- **Enhance Differences:** If differences between fitness values are too small, choose $a>1$ to amplify them.
- **Dampen Differences:** If one individual's fitness is too dominant, choose $0<a<1$ or adjust b to reduce the gap.

FITNESS RANKING IN EVOLUTIONARY ALGORITHMS

- Replaces raw fitness with rank-based values (best = rank 1)
- Reduces sensitivity to outliers and noise
- Supports controlled selection pressure (linear or exponential)
- Encourages diversity by limiting dominance
- Ideal for tournament and stochastic selection methods
- Prevents premature convergence in deceptive landscapes

I. LINEAR RANKING

- **Idea**: Assigns fitness in a **linearly decreasing/increasing** manner based on rank.
- The best individual gets a fitness of max, the worst gets min.
- Formula typically involves a **rank-based interpolation** between min and max.
- **Example**:
  If max = 2 and min = 1, and you have 5 individuals, their assigned values would be like:

```
Rank 1: 2.0
Rank 2: 1.75
Rank 3: 1.5
Rank 4: 1.25
Rank 5: 1.0
```

2. EXPONENTIAL RANKING

- **Idea**: Assigns exponentially decreasing fitness values.
- The best individual gets the highest exponential value; each successive rank gets a fitness reduced by a constant ratio.

Example:

Using a decay factor (e.g., $\beta < 1$):

Rank 1: $\beta^0 = 1$ Rank 2: β^1 Rank 3: β^2 ... Rank n: $\beta^{(n-1)}$

- Normalized to ensure sum of all fitness values is 1 (for probabilistic selection).

ALTERNATIVE PARENT SELECTION STRATEGIES

- Tournament selection: Best of k random picks
- Threshold selection: Uniform chance for top individuals
- Ranking methods: Based on ordered fitness
- Trade-off: Higher selection pressure vs. diversity
- Selection method chosen per problem need

SURVIVOR SELECTION & POPULATION UPDATE

- Strategies: Age-based, fitness-based, elitism
- SGA: Full replacement vs. SSGA: Steady-state update
- Preserve best solutions while exploring new ones
- Replace similar or weak individuals
- Balances quality and diversity over generations

REPLACEMENT STRATEGIES

1. REPLACE THE MOST SIMILAR GENE
2. REPLACE THE WORST GENES

REPLACE THE MOST SIMILAR GENE

- The “**De Jong Strategy”**:
- Replace Most Similar Genes:
  - Newborns (newly created offspring) replace the most similar individuals in the population.
  - This approach ensures that new genes are integrated into the population while maintaining genetic diversity.
  - It helps prevent the population from becoming too homogeneous by replacing individuals that are genetically similar to the newborns.

REPLACE THE WORST GENES

- **The Derating Method:**
- Reduce the Fitness of Similar Genes:
  - This method involves reducing the fitness of individuals that are genetically similar to each other.
  - By lowering the fitness of similar genes, the algorithm encourages diversity, preventing similar individuals from dominating the population.
  - This promotes a more varied gene pool and helps explore different regions of the search space.

CORE AGING CONCEPT

- **Individual Class:**

Age Increment: Increase each individual's age by 1 every generation.

Max Age Threshold: Replace individuals that exceed the defined maximum age.

Immunity: Individuals below the threshold remain protected.

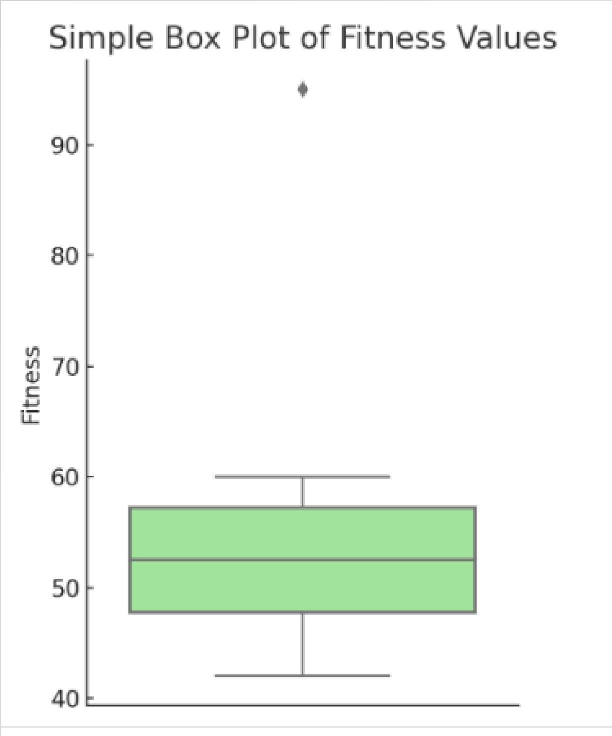
- **Elitism:** Assign higher age thresholds to elite individuals.

```
class Individual:
    def __init__(self, genes):
        self.genes = genes
        self.age = 0
```

AGE EXTENSIONS & VARIATIONS

- **Random Initialization:** Start with random ages.
- **Fitness-Based Thresholds:** Set personal age limits based on fitness.
- **Reproduction Condition:** Allow mating only when a threshold age is reached.
- **Dynamic Threshold:** Adjust max age as evolution progresses.

GENERATION FITNESS BOX PLOT




```
[42, 45, 47, 50, 52, 53, 55, 58, 60, 95]
```

- **Median (middle line)** ≈ 52.5
- **Box (IQR)** spans from Q1 (≈ 47) to Q3 (≈ 58)
- **Whiskers** extend to the typical range (excluding extreme values)
- **Dot above the box** marks `95` as an **outlier**

TOP-AVERAGE SELECTION PROBABILITY RATIO

1. **Step 1: Compute Individual Selection Probabilities**

- For many evolutionary algorithms, selection probabilities are computed based on fitness. For example, using fitness-proportional (roulette wheel) selection, the probability pi for individual i is: where fi is the fitness of individual i and N is the total number of individuals.

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$$

TOP-AVERAGE SELECTION PROBABILITY RATIO

Step 2: Identify the Top Individuals

- **Definition:**
  Choose a subset of individuals defined as "top" (e.g., the best 10% or the best individual).
- **Computation:**
  Compute the average selection probability for this top subset:

$$p_{\text{top}} = \frac{1}{M} \sum_{i \in \text{top}} p_i$$

- where M is the number of top individuals.

TOP-AVERAGE SELECTION PROBABILITY RATIO

Step 3: Compute the Average Selection Probability for the Entire Population

- **Computation:**
  Since the probabilities pi sum to 1 over the entire population:

$$p_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} p_i = \frac{1}{N}$$

- (for fitness-proportional selection) or computed directly if a different selection mechanism is used.

TOP-AVERAGE SELECTION PROBABILITY RATIO

Step 4: Form the Ratio

- **Formula:**
  The Top-Average Selection Probability Ratio is then given by: $$\text{Ratio} = \frac{p_{\text{top}}}{p_{\text{avg}}}$$

- A higher ratio indicates that top individuals are disproportionately favored compared to the average. Implying stronger selection pressure

BIN PACKING HEURISTICS

- First Fit (FF): Place each item in the first bin that can accommodate it.
- Best Fit (BF): Place each item in the bin that will leave the least remaining capacity.
- Worst Fit (WF): Place each item in the bin that will leave the most remaining capacity.
- Next Fit (NF): Place each item in the current bin until it is full, then move to the next bin.
- First Fit Decreasing (FFD): Sort items in decreasing order and apply First Fit.
- Best Fit Decreasing (BFD): Sort items in decreasing order and apply Best Fit.
- These heuristics offer different strategies to balance packing efficiency and computational complexity.
- The choice of heuristic depends on the specific requirements and constraints of the bin packing problem at hand.

THE AI IQ TEST OF GENERAL ARTIFICIAL INTELLIGENCE

- **To pass the test, the AI must:**
  - Learn from a limited number of examples
  - Abstract and reason
  - Handle problems it has never SEEN before
- Based on the ARC dataset (Abstraction and Reasoning Corpus), where the example is given as:
  Input → Image transformation → Output
- GivenA set of examples, and by generalizing the example, the AI must generate a **fitting output image**.