Evolutionary algorithms № 2

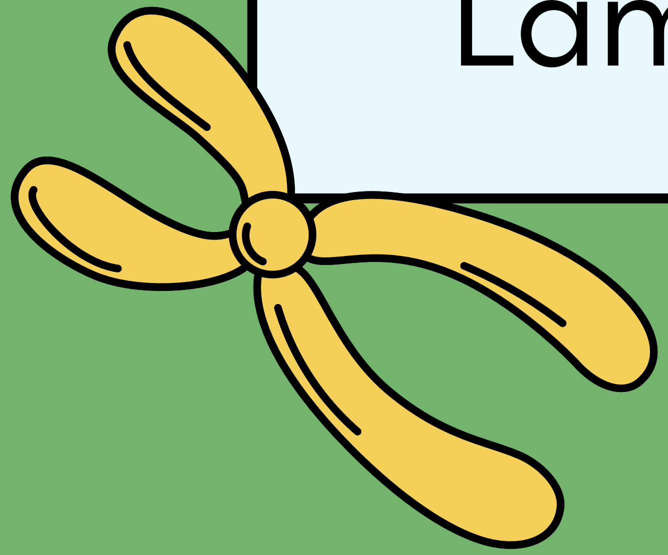
27.09.2024

Session Outline

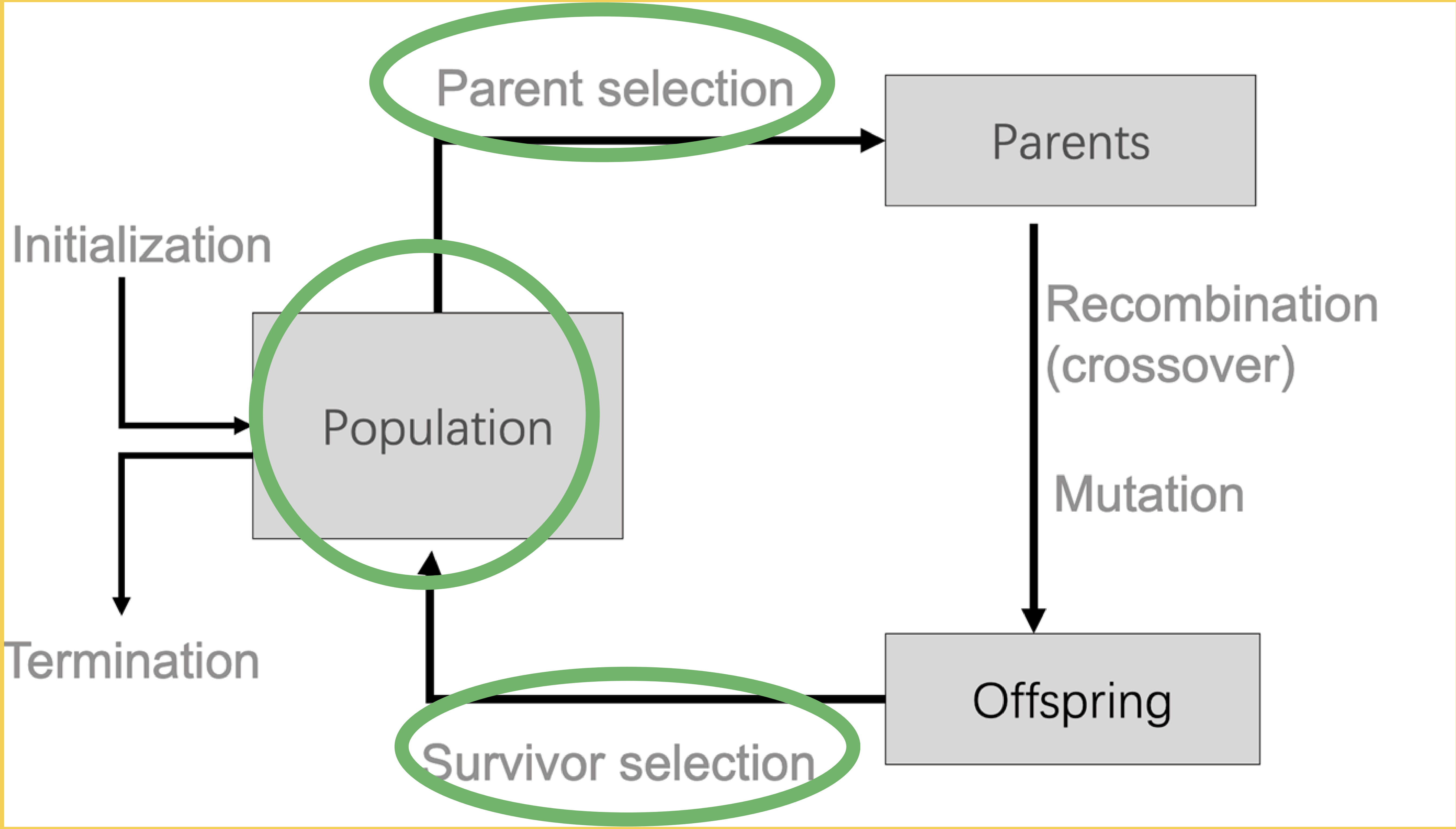
EA selection, merge theory

Code for EA

Lamarckian and Baldwinian



Where are we now



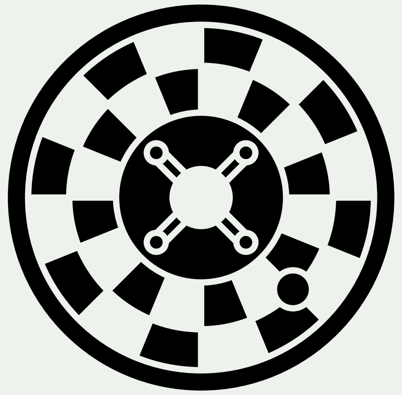
Brief repetition of previous week

	Mutation	Recombination
Binary	Bit flip	Single point, uniform, n-point
Integers	Creep mutation Random resetting	n-point, uniform
Real valued/float	Uniform non Uniform	Discrete Intermediate
Permutations	Swap, insert, scrumble inversion	PMX, Edge recombination, order crossover, cycle crossover

Genotype - set of genes or values
Phenotype - what could be developed based on the genotyoe

Locus: the position of a gene
Allele = 0 or 1 (What values a gene can have)
Gene: one element of the array
Genotype: a set of gene
valuesPhenotype: What could be built/developed based on the genotype

PARENT SELECTION



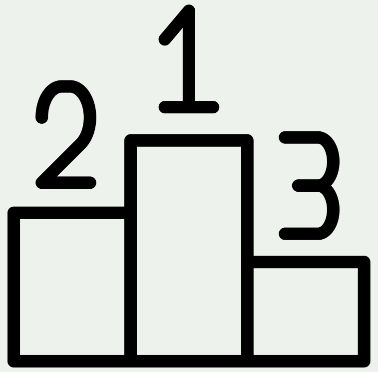
the process of choosing individuals from the current population to act as parents for the creation of the next generation.

Roulette Wheel Selection:

Individuals are selected with a probability proportional to their fitness.
Fitness values are normalised to create a probability distribution.

Tournament Selection:

Randomly select a subset of individuals (a tournament) and choose the fittest individual from the subset.
Repeated until a sufficient number of parents are chosen.



Rank-Based Selection:

Individuals are ranked based on their fitness.
Selection is then based on the rank rather than absolute fitness.

PARENT SELECTION

Fitness-Proportionate Selection (FPS)

• Probability for individual i to be selected for mating in a population size μ with FPS is

$$P_{FPS}(i) = f_i \Big/ \sum_{j=1}^{\mu} f_j$$

+

- faster convergence
- Intuitive
- Gives chance to weak individuals

−

- Exploitation > Exploration
- Premature convergence
- Loss of selection pressure#
- Loss of diversity

Tournament Selection

Idea for a procedure using only local fitness information:

- Pick k members at random then select the best of these
- Repeat to select more individuals

Probability of selecting i will depend on:

- Rank of i
- Size of sample k
- higher k increases selection pressure

- Picking without replacement increases selection pressure
- • Whether fittest contestant always wins (deterministic) or this happens with probability p

+

- Maintains Diversity
- easy-to-implement
- randomness can help avoid premature convergence and explore a diverse set of solutions

−

- Biased Towards Better Individuals
- Less Pressure for Diversity
- Biased Towards Better Individuals
- Lack of Global Information

SURVIVOR SELECTION

The process determines which individuals from the combined set of parents and offspring will make up the next generation.

ELITISM

Elitism involves preserving a certain number of the best individuals (based on fitness) from the current generation and allowing them to directly pass on to the next generation without undergoing genetic operations. Elitism helps maintain high-performing solutions across generations, contributing to convergence.

PRESERVE DIVERSITY

- **Explicit approaches**
- Make similar individuals compete for resources (fitness)
- Make similar individuals compete with each other for survival

Implicit approaches:
- Impose an equivalent of geographical separation
- Impose an equivalent of speciation

Fitness Sharing

- Restricts the number of individuals within a given niche by "sharing" their fitness
- Need to set the size of the niche σshare in either genotype or phenotype space
- run EA as normal but after each generation set

$$f'(i) = \frac{f(i)}{\sum_{j=1}^{\mu} sh(d(i,j))} \qquad sh(d) = \begin{cases} 1 - d/\sigma & d \le \sigma \\ 0 & otherwise \end{cases}$$

Crowding

New individuals replace similar individuals
- Randomly shuffle and pair parents, produce 2 offspring
- Each offspring competes with their nearest parent for survival (using a distance measure)
- Result: Even distribution among niches

Automatic Speciation

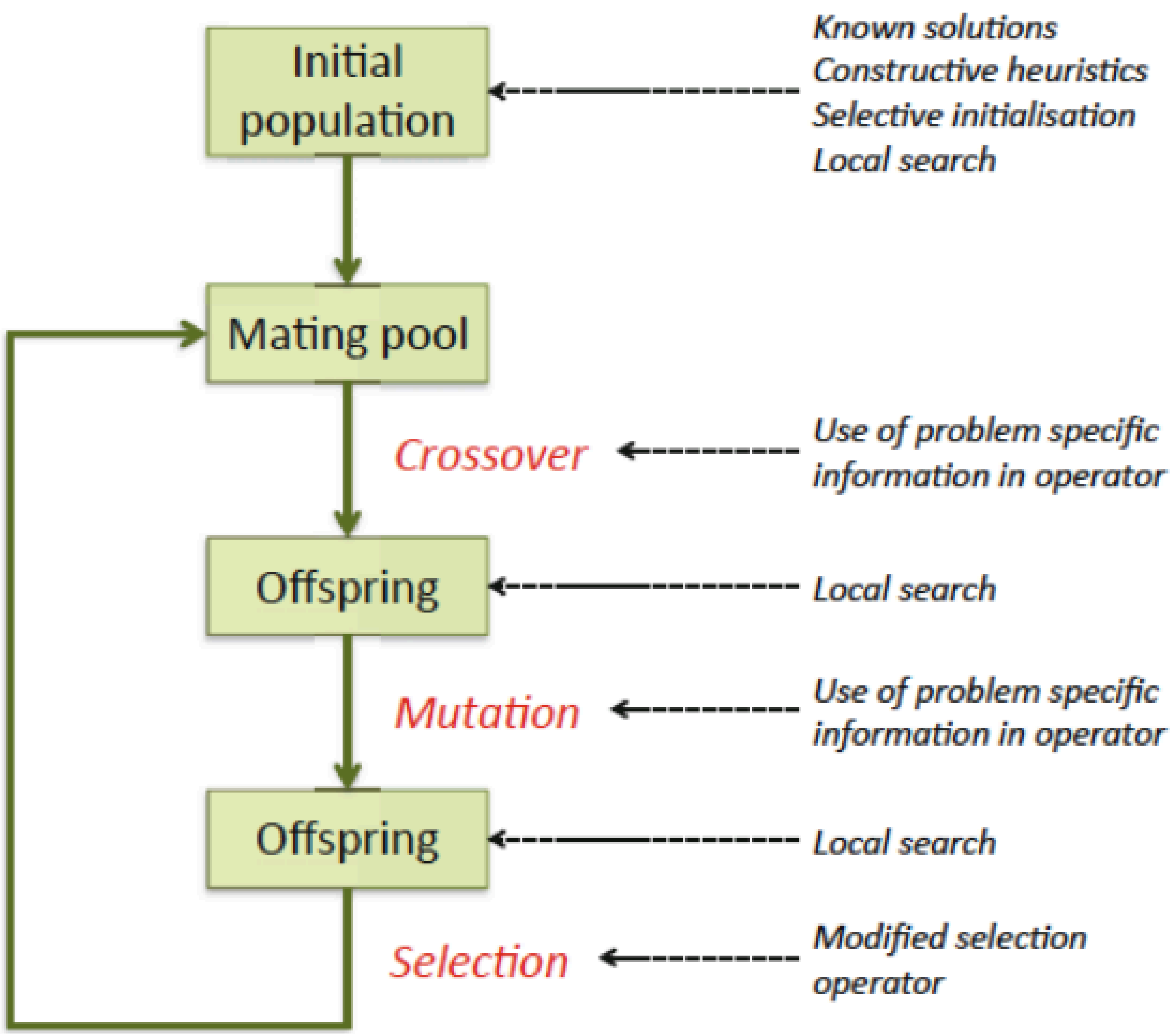
Either only mate with genotypically / phenotypically similar members or
- Add species-tags to genotype
- initially randomly set
- when selecting partner for recombination, only pick members with a good match

"Island" Model

Periodic migration of individual solutions between populations
- Run multiple populations in parallel
- After a (usually fixed) number of generations (an Epoch), exchange individuals with neighbours
- Repeat until ending criteria met

HYBRIDISATION

integration of different optimization techniques, algorithms, or problem-solving approaches within an evolutionary algorithm



Memetic

A Memetic Algorithm (MA) is a **hybrid evolutionary algorithm** that combines principles from evolutionary algorithms with local search techniques.

The main objectives of memetic algorithms are to combine the global exploration capabilities of evolutionary algorithms with the local exploitation abilities of local search methods. By leveraging the strengths of both exploration and exploitation, memetic algorithms aim to efficiently navigate complex solution spaces, find high-quality solutions, and overcome challenges such as premature convergence.

LOCAL SEARCH

Local search is a optimization technique employed to fine-tune or improve individual solutions in the **neighborhood of the current candidate**. Unlike global search operators such as mutation and crossover that explore the entire solution space, **local search focuses on exploiting the local structure around a given solution**.

it is often applied as a post-processing step to individuals in the population, aiming to refine or optimize them further. It is particularly useful when the initial individuals generated by the evolutionary operators are close to optimal solutions but may still have room for improvement.

LAMARKIAN AND BALDWINIAN

Lamarckian Evolution:

Lamarck proposed the theory of the inheritance of acquired traits, suggesting that individuals could acquire new traits during their lifetime and pass them on to their offspring. In the context of evolutionary algorithms, Lamarckian evolution implies that changes made to an individual during its lifetime can be directly passed on to its descendants.

Lamarckian Evolution in EA:

allowing individuals to adapt and improve their solutions during their lifetime, and **these improvements would be directly reflected in the genetic material passed to the next generation.**

Baldwinian Evolution:

learned behaviors or adaptations acquired during an individual's lifetime could influence the direction of evolution. However, unlike Lamarck, Baldwin **did not propose direct inheritance of acquired traits**. Instead, he suggested that **learning could guide the exploration of the solution space and affect the evolutionary trajectory.**

Baldwinian Evolution in EA:

Baldwinian approach involves using learning or adaptation mechanisms to guide the evolutionary process. Individuals may learn during their lifetime, and this acquired knowledge influences their chances of survival and reproduction. However, the **acquired knowledge itself is not directly passed on to offspring**.

MULTIOBJECTIVE AND HOW TO

Multiobjective Optimization Problems (MOPs) involve optimizing multiple conflicting objectives simultaneously. In evolutionary algorithms (EAs), addressing multiobjective optimization is particularly important because it allows for the exploration of trade-offs and the generation of a set of solutions known as the Pareto front. The Pareto front represents the best compromise solutions where no solution is superior to another in all objectives.

In a multiobjective optimization problem, the goal is not to find a single optimal solution but to discover a set of solutions that represent the trade-offs between conflicting objectives. These solutions collectively form the Pareto front.

Simple solution; Weighted sum – incorporated in fitness function

Pareto front and pareto optimality

Pareto Front:

The Pareto front is a set of solutions in multiobjective optimization where no solution is better than another in all objectives.
It represents the trade-offs between conflicting objectives, showcasing solutions that achieve the best compromise.

Pareto Optimality:

A solution is Pareto optimal if it belongs to the Pareto front; there is no other solution that dominates it.
Dominance implies being better than another solution in at least one objective without being worse in any other.

Finds the set of non-dominated solutions

Solution x dominates solution y, ($x \preccurlyeq y$), if:
x is better than y in at least one objective,
x is not worse than y in all other objectives