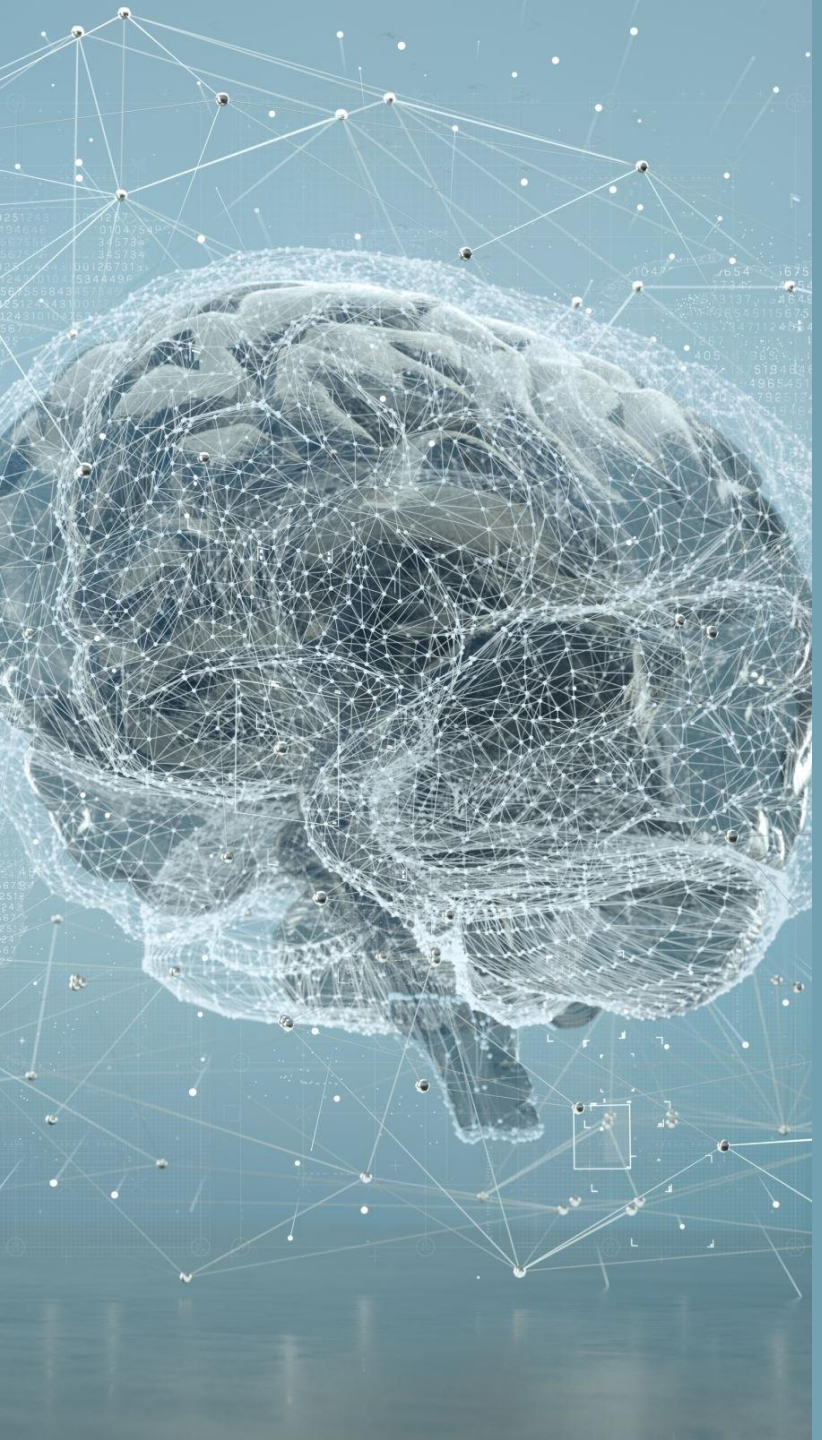


كلية الحاسبات والذكاء الاصطناعي

FACULTY OF COMPUTERS & ARTIFICIAL INTELLIGENCE

ARTIFICIAL INTELLIGENCE
LAB 09 – GENETIC ALGORITHM
PART (1) –THEORETICAL-

AI Course Team

Eng. Yousef Elbaroudy – Eng Sahar Mohammed

Eng. Reham Ibrahim – Eng. Mawada Ashraf

Eng. Eman Nasser

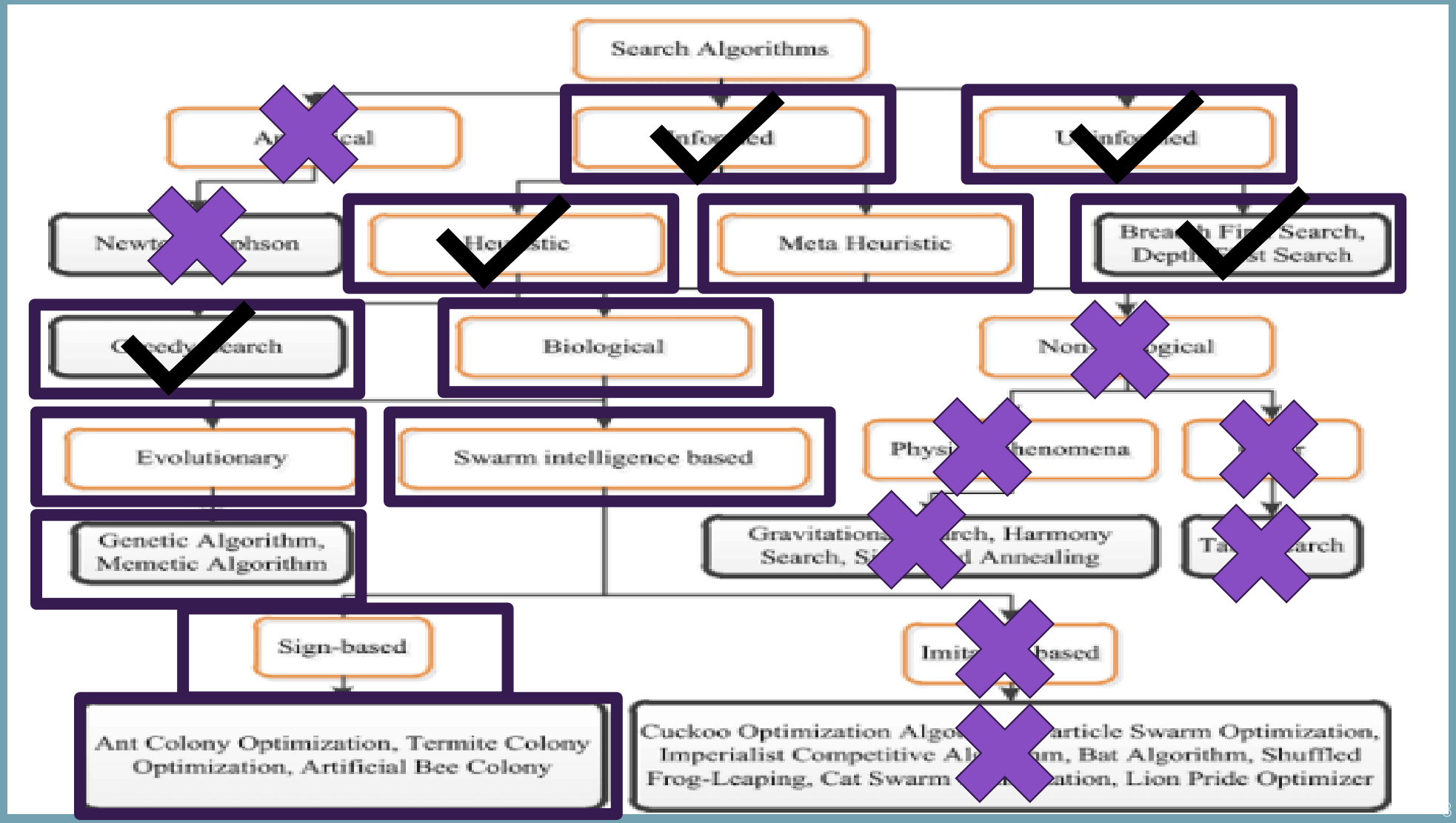
2025/2026

GUIDELINES

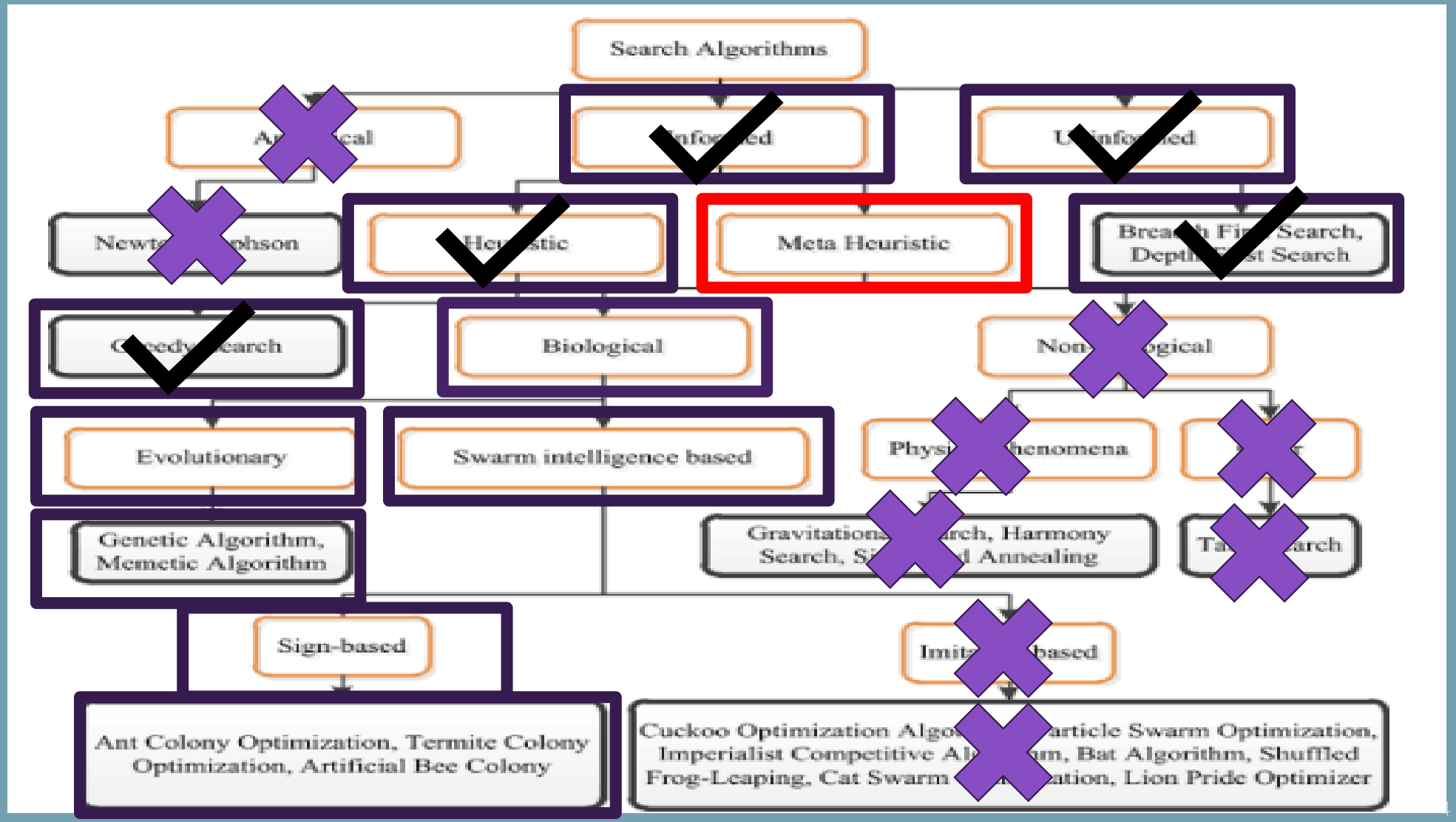
You don’t need to memorize what will be mentioned. Instead, try to understand

Each PowerPoint Presentation will be available as PDF file on Google Drive Folder of the Course

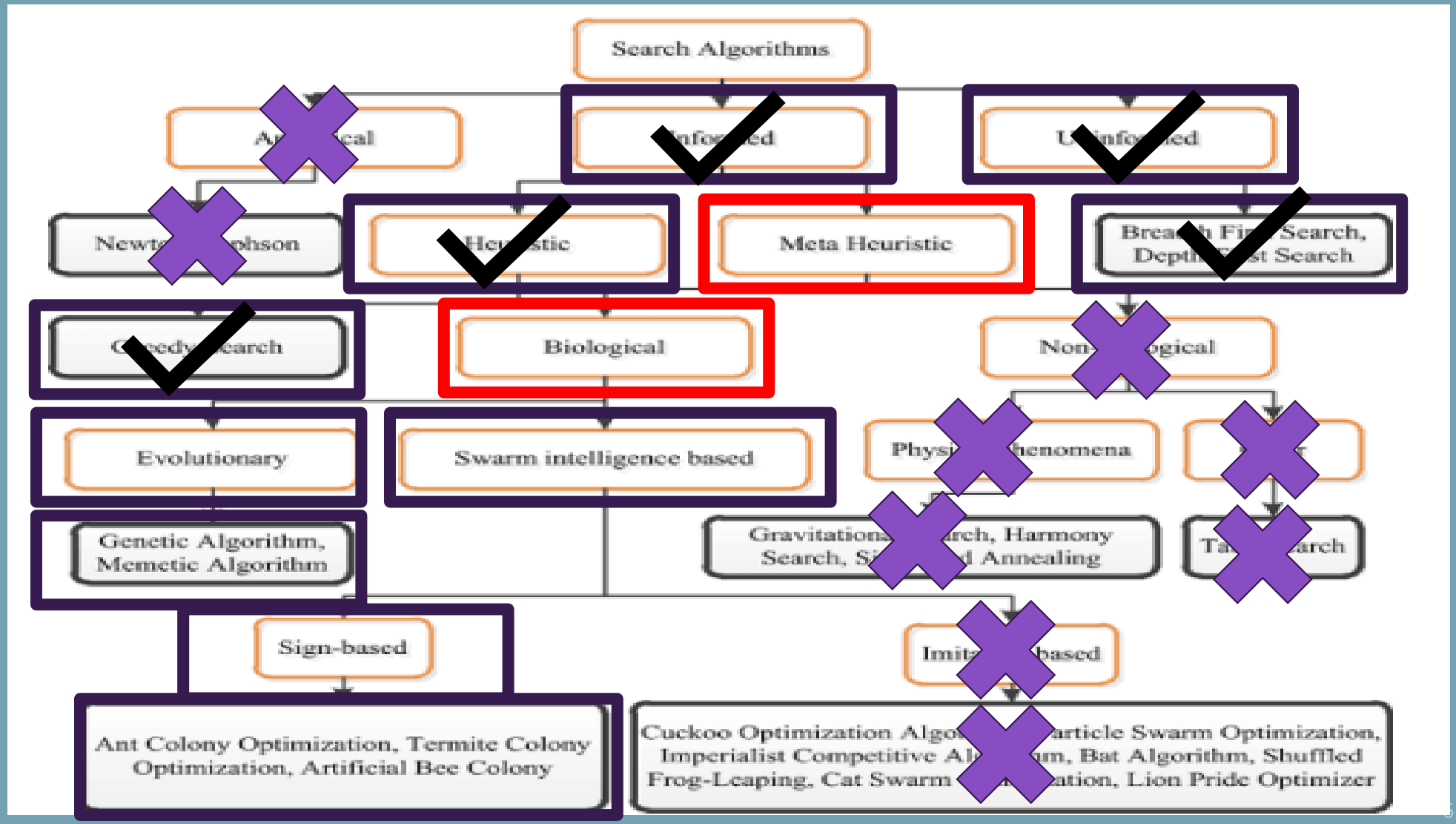
THE CLASSIFICATION OF SEARCHING ALGORITHMS



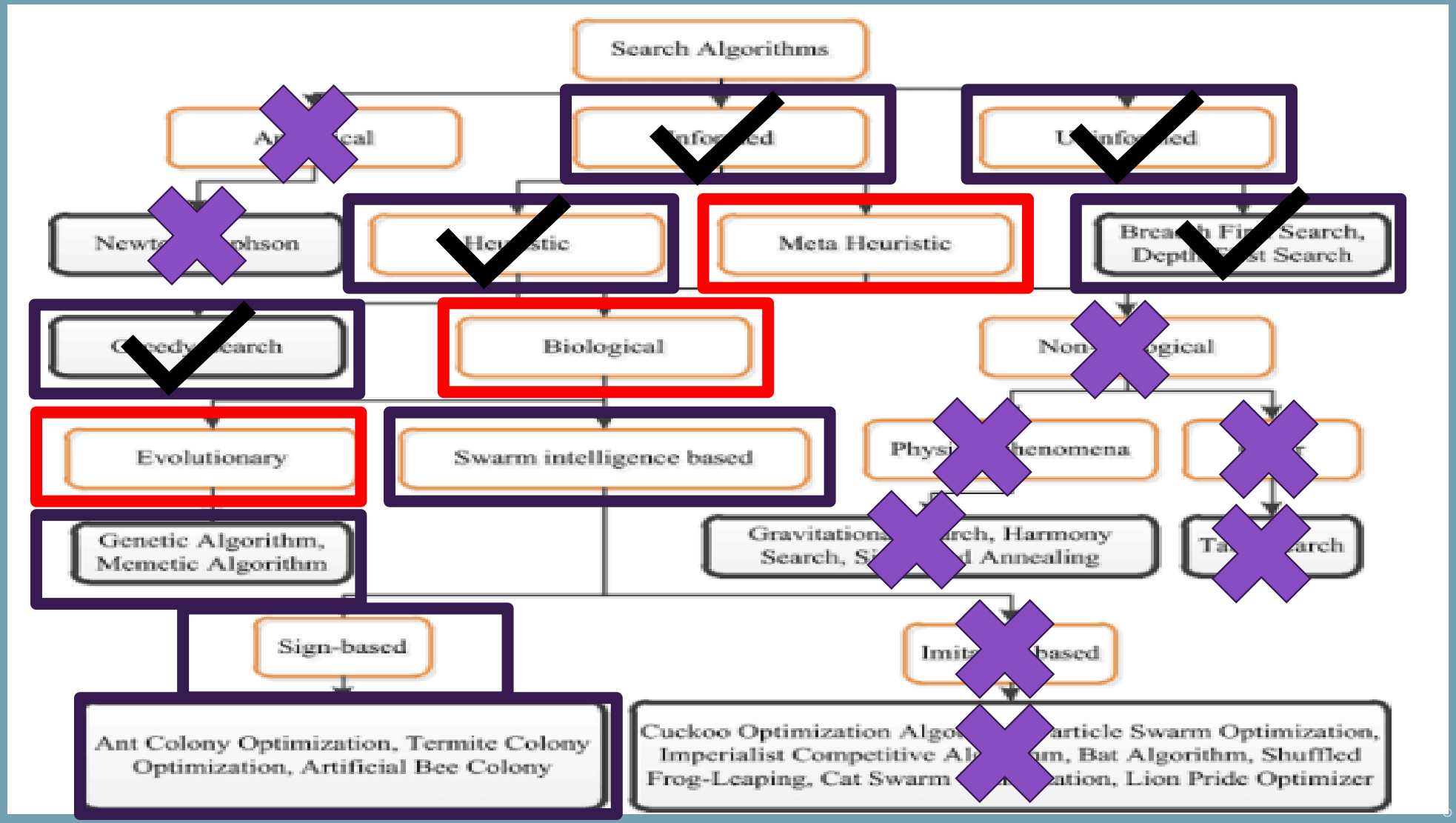
THE CLASSIFICATION OF SEARCHING ALGORITHMS



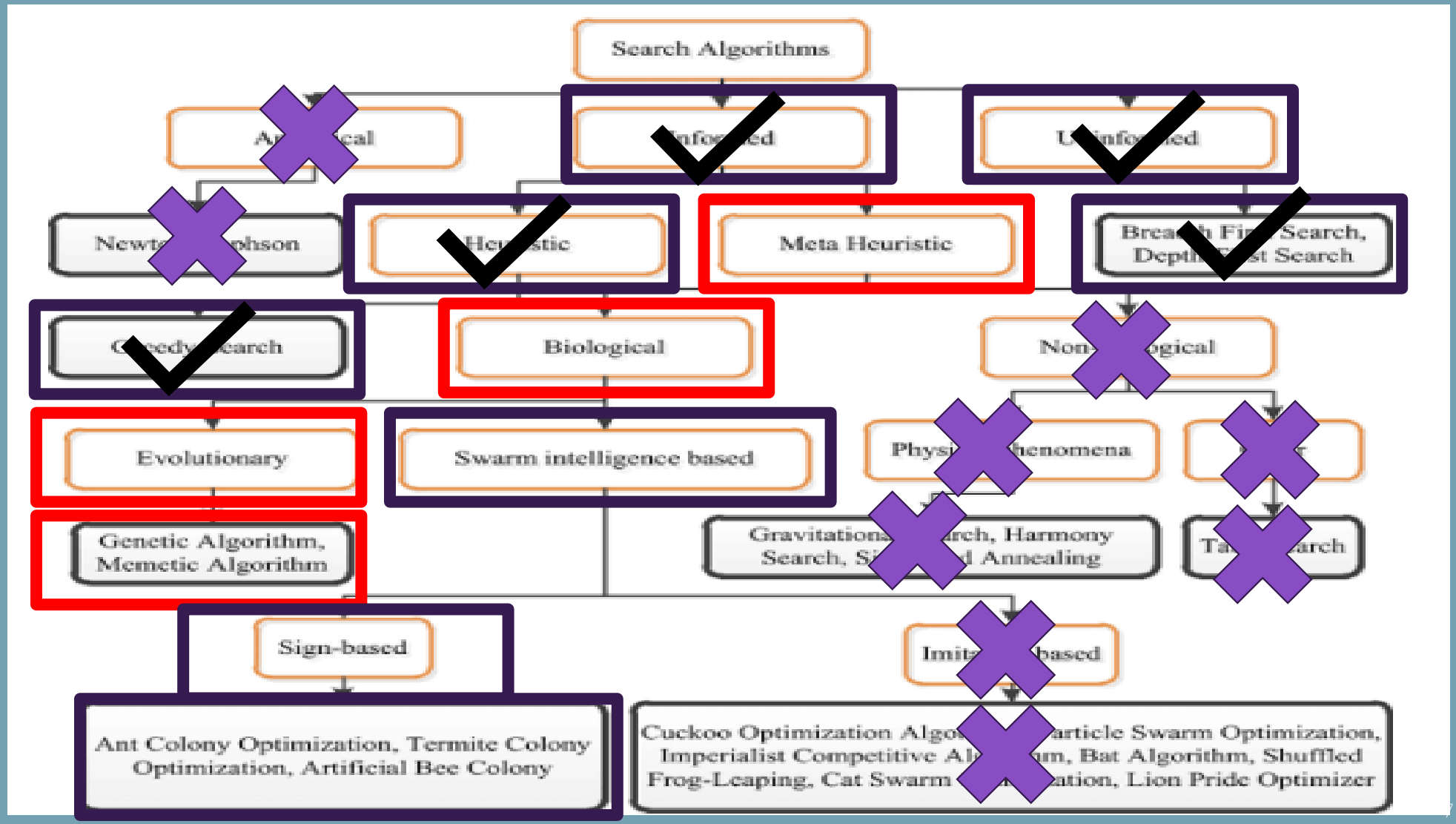
THE CLASSIFICATION OF SEARCHING ALGORITHMS



THE CLASSIFICATION OF SEARCHING ALGORITHMS



THE CLASSIFICATION OF SEARCHING ALGORITHMS



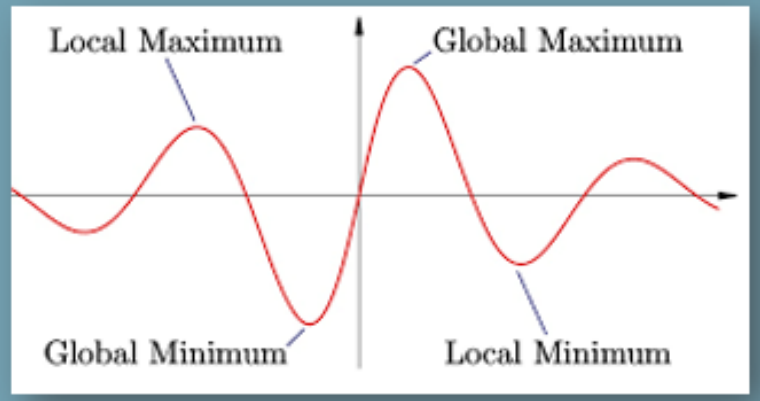
HOW TO REPAIR "GREEDY" ?

GREEDY SEARCH (طمع/جشع)

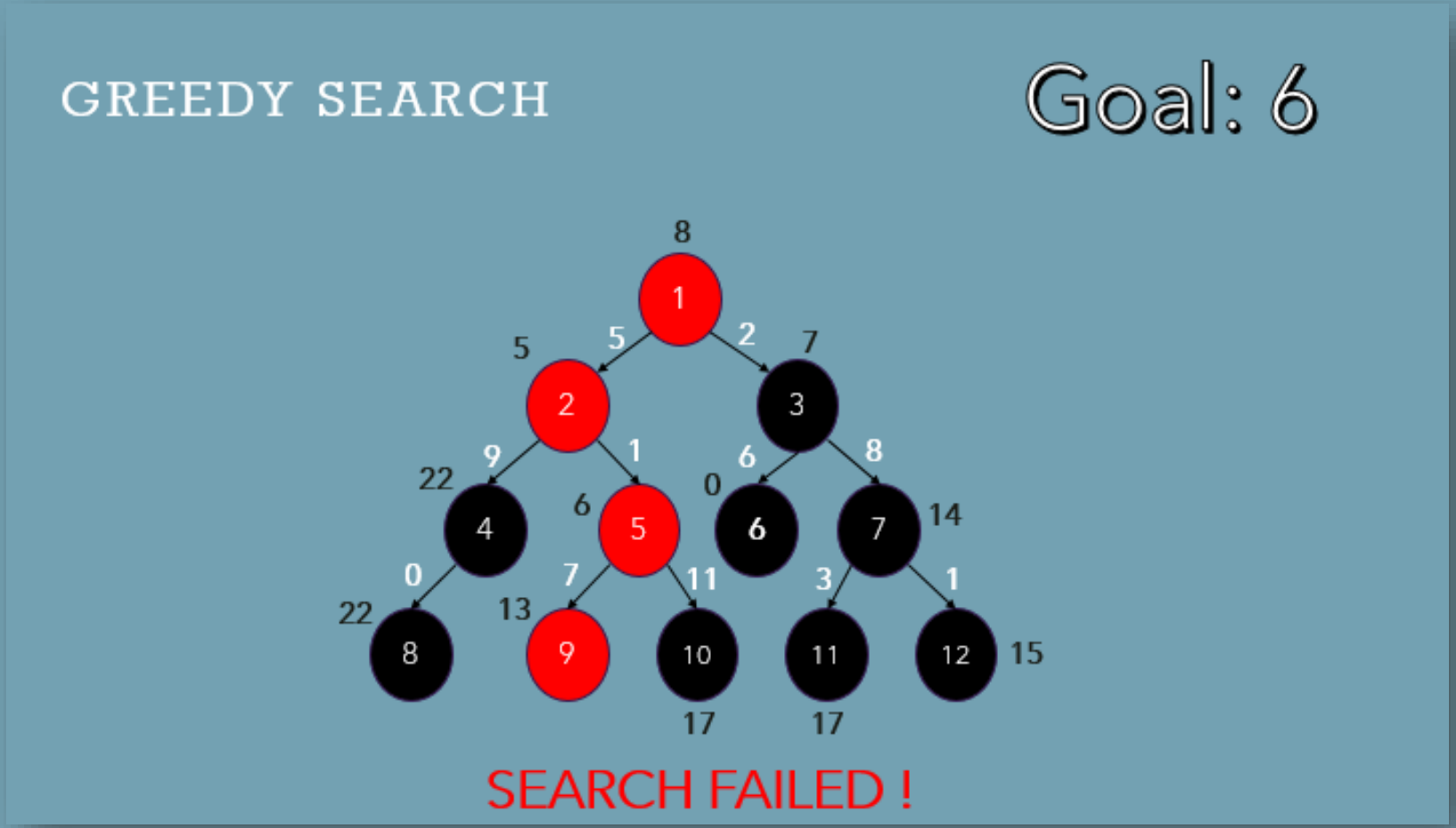
- **Greedy search** is a form of Best-First Search that uses a **heuristic** to always expand the node that seems **closest to the goal**, based **ONLY ON THAT ESTIMATE without looking for the others.** It is like the running toward the goal without looking for anything else.

- In Greedy search, it always looking for the best/closet to the goal, which it seems you are reaching to it, but it may not actually be close to the goal at all (Trap). Which it is called **Local Minima**

In other words, Greedy search looking for the minimum but not the goal



DON'T YOU REMEMBER ?



INTRODUCING META-HEURISTIC

KEY CONCEPTS OF META-HEURISTIC ALGORITHMS


Inspired from Artificial Intelligence rather than pure math, which means they are not based on algebraic models


There is no theoretical, which means lack of proofs


Capability of addressing both discrete and continues optimization problems


Efficiently explore the search space in order to find near-optimal feasible solutions


Provide no guarantee of global or local optima (BEST BENEFIT) which can lead to computational complexity


They balance exploration and exploitation, and incorporate randomness to avoid getting stuck in local solutions

- Meta-Heuristic: high-level, problem-independent strategies or frameworks designed to solve complex optimization problems.

EXPLORATION AND EXPLOITATION

Exploration

- Searching new, unexplored areas of the solution space.
- Discover diverse solutions and avoid local optima.
- **Randomness**, **diversification**, and wide search space coverage.
- May explore unnecessary regions, increasing computational cost.

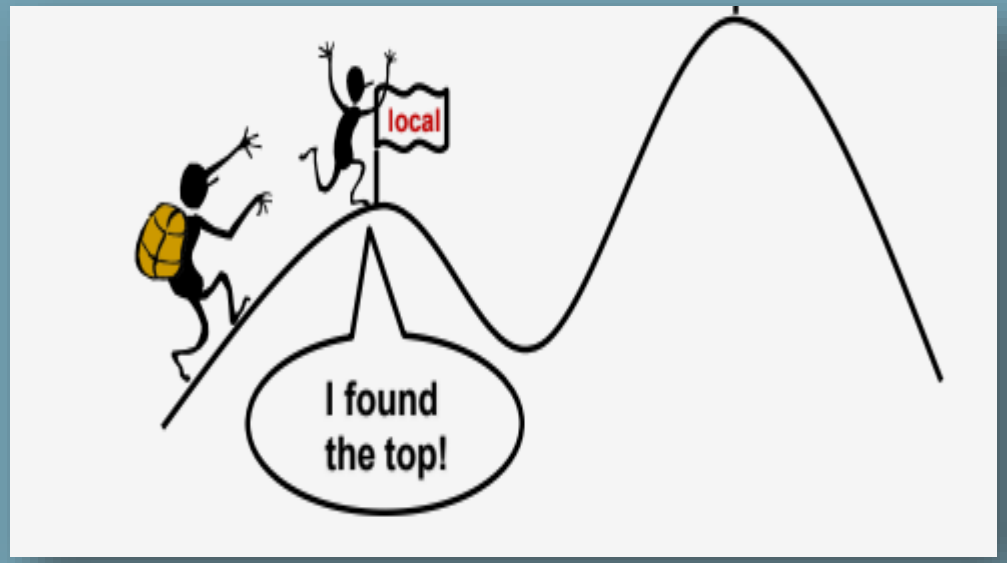
Exploitation

- Refining known good solutions to improve performance.
- Optimize the best-known solutions for higher quality results.
- Focused, intensification, and local optimization.
- May get trapped in local optima and miss the global best solution.

GENETIC ALGORITHM

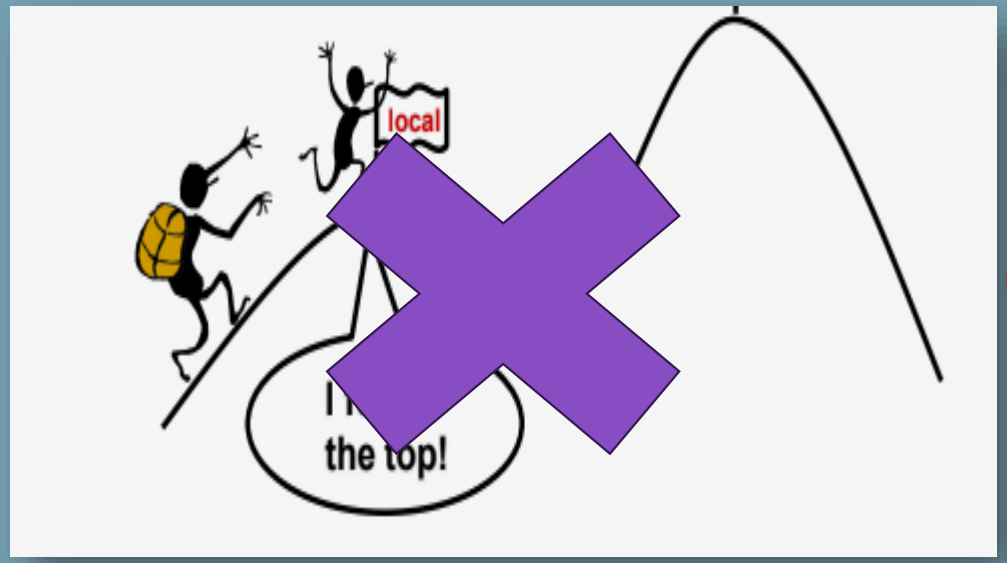
GENETIC ALGORITHM

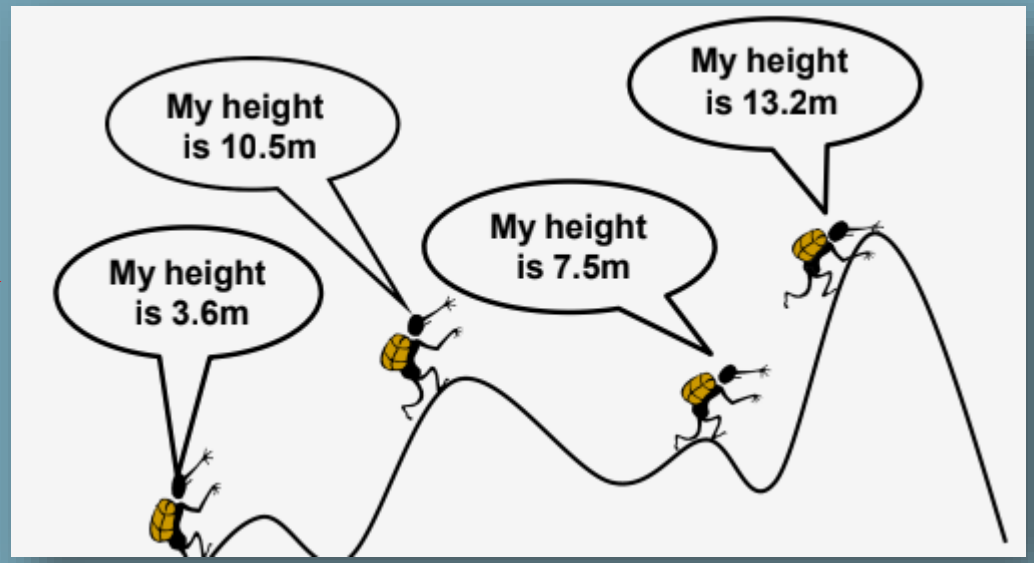
Genetic Algorithm (GA) is a method for solving optimization and search problems by mimicking the process of natural evolution. It's especially useful when the search space is huge, complex, or poorly understood.



GENETIC ALGORITHM

Genetic Algorithm (GA) is a method for solving optimization and search problems by mimicking the process of natural evolution. It's especially useful when the search space is huge, complex, or poorly understood.





GENETIC ALGORITHM

Genetic Algorithm (GA) is a method for solving optimization and search problems by mimicking the process of natural evolution. It's especially useful when the search space is huge, complex, or poorly understood.

The Genetic Algorithm (Mainly all meta-heuristic algorithms) uses multiple climbers in parallel to find the global optimum (a population-based algorithm).

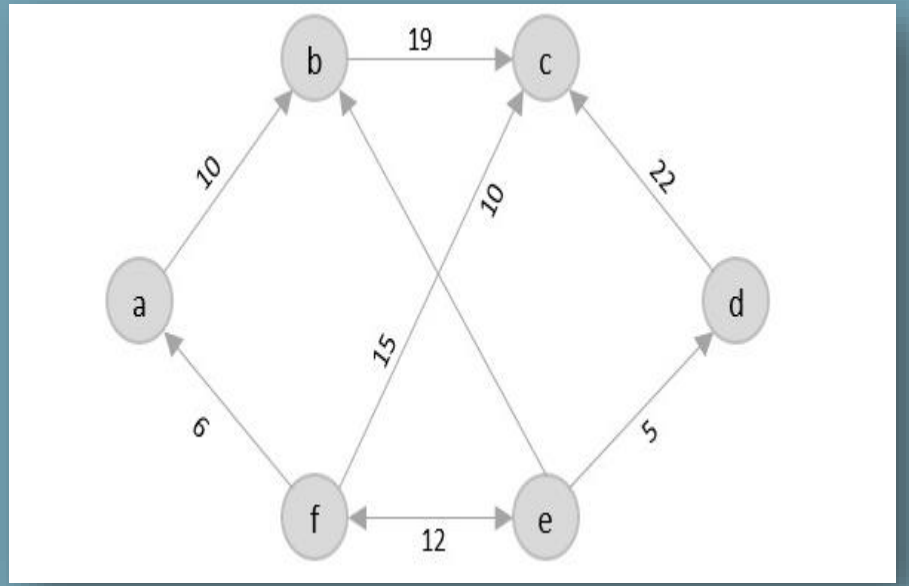
GA's are useful for solving multidimensional problems containing many local maxima (or minima) in the solution space.

WHAT KIND OF ENVIRONMENTS ?

COMMON PROBLEMS TO SOLVE - 1

Traveling Salesmen Problem

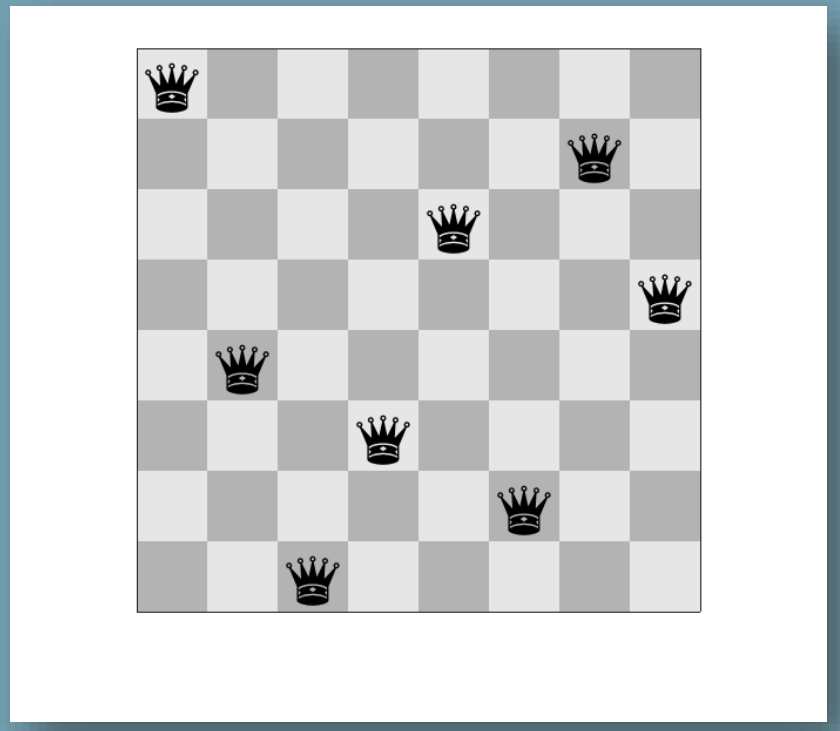
Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?



COMMON PROBLEMS TO SOLVE - 2

8-Queen Problem

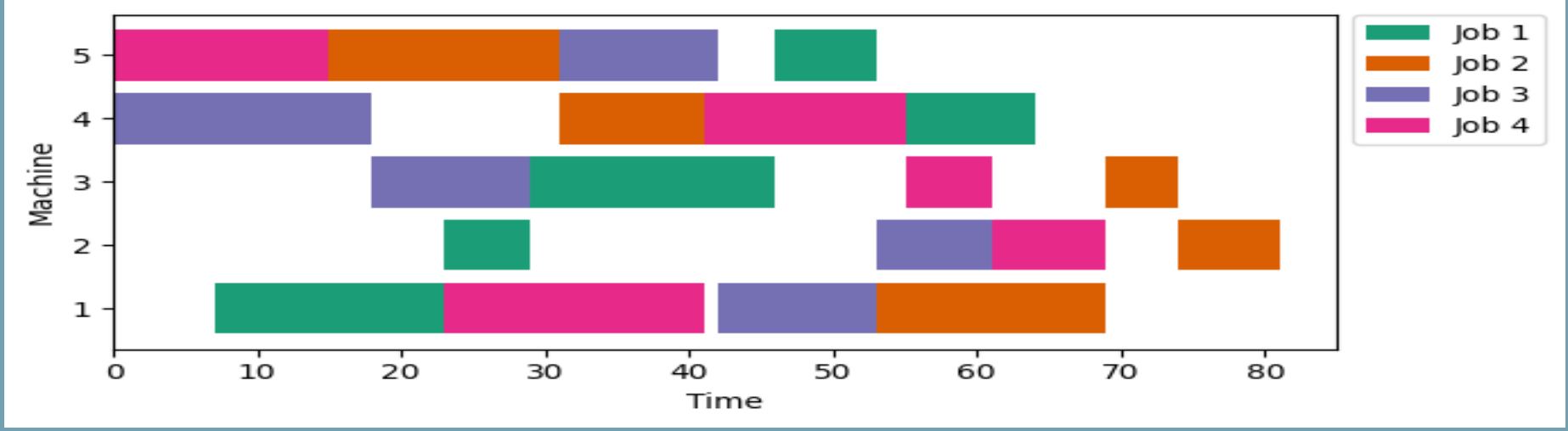
placing eight chess queens on an 8×8 chessboard so that no two queens threaten each other



COMMON PROBLEMS TO SOLVE - 3

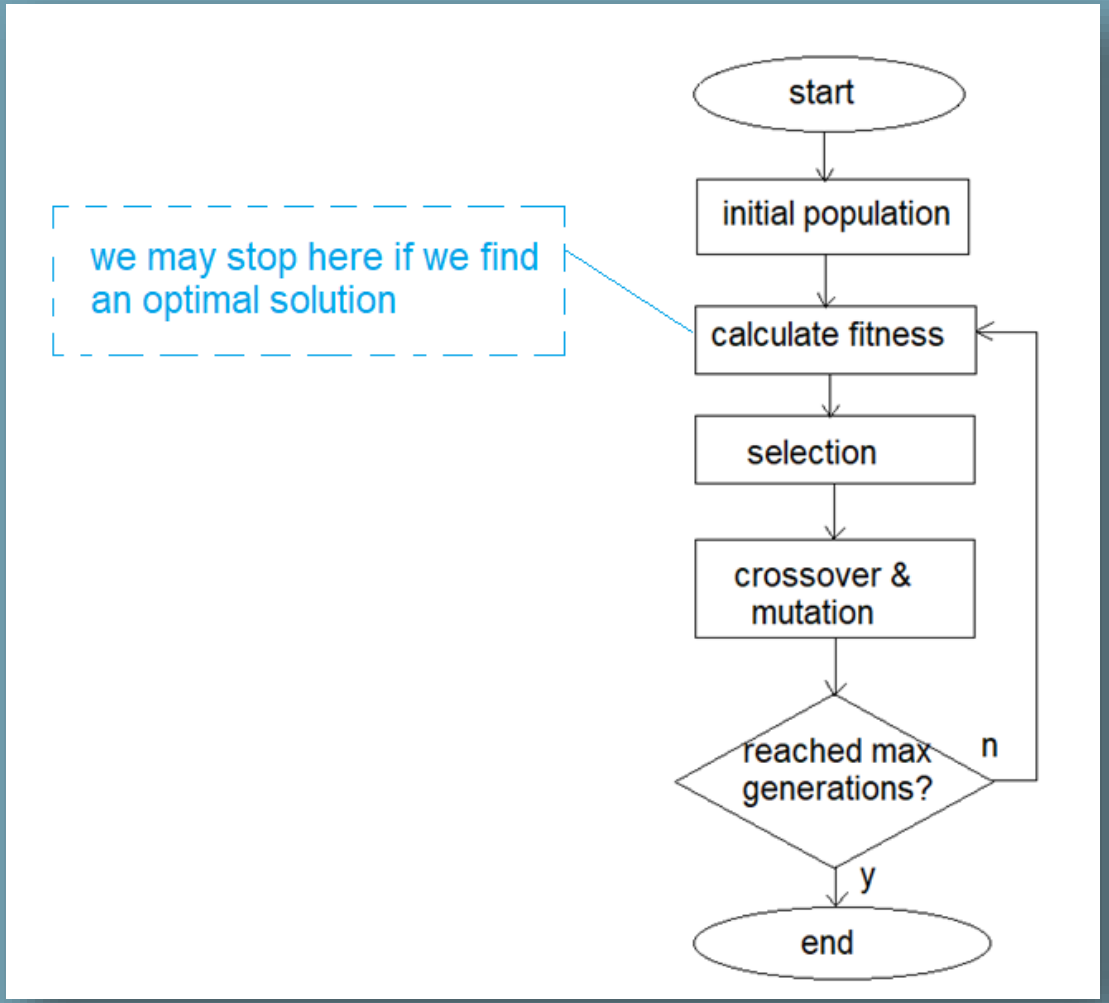
Scheduling Problem

Varying processing times, which need to be scheduled on m slots with varying time intervals, while trying to minimize the makespan – the total length of the schedule (that is, when all the jobs have finished processing).



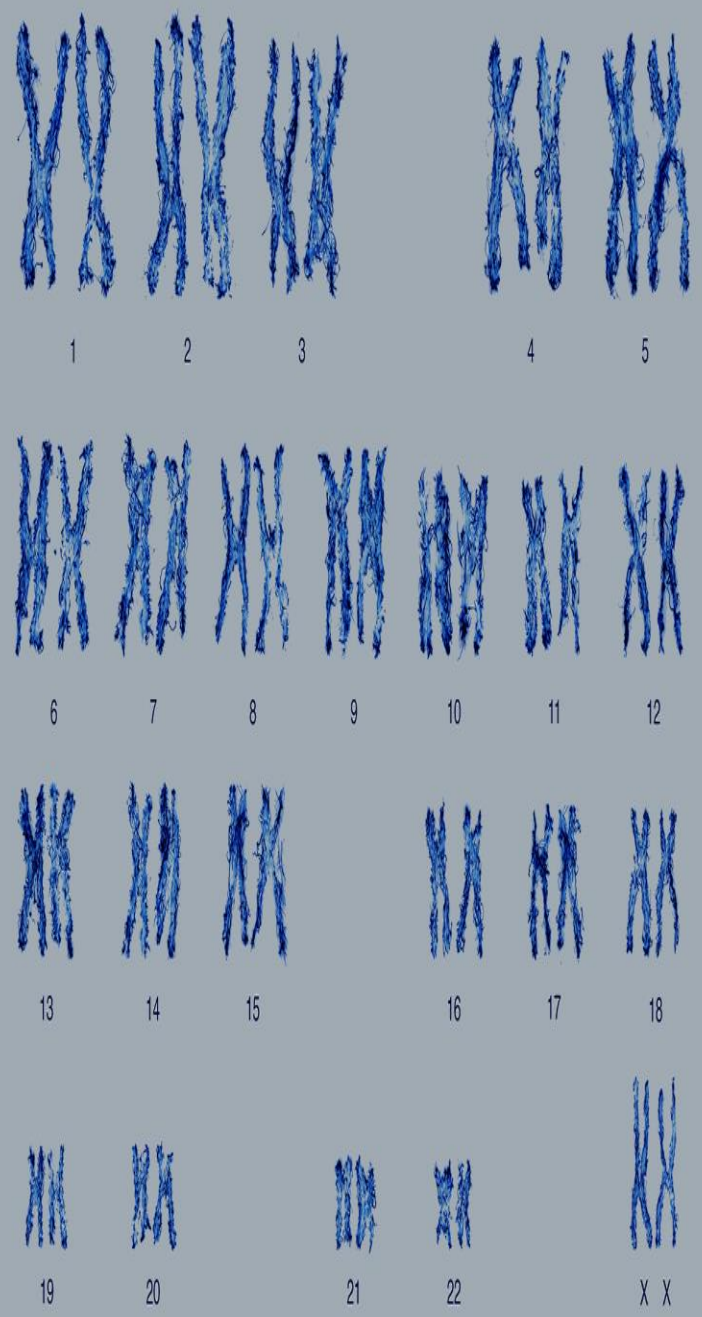
STEPS OF THE ALGORITHM

GENETIC ALGORITHM STEPS



(1) INITIAL POPULATION

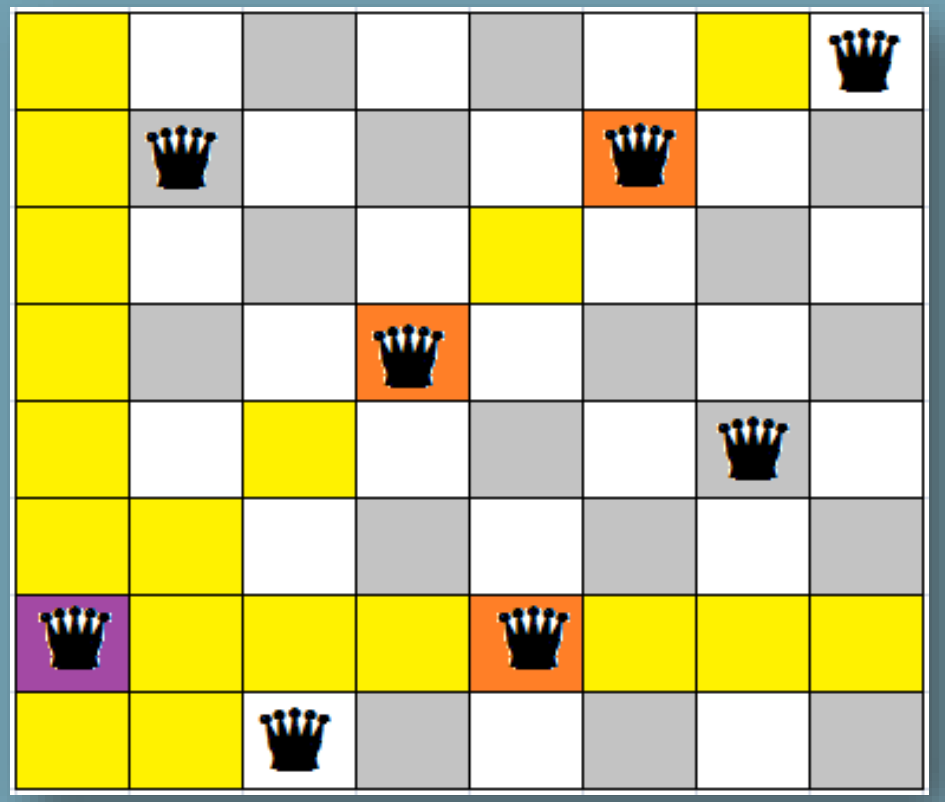
- We randomly generate a set of possible solutions (called individuals or chromosomes).
- You must know **how to represent (encode)** the problem correctly in order to implement the algorithm efficiently before even proceeding to the algorithm.
- It can be determined the number of individuals (possible searchers) you want.



(2) CALCULATE FITNESS (HEURISTIC)

- We evaluate HOW "GOOD" EACH INDIVIDUAL IS using a **fitness function** based on criteria or some penalty.
- Our target is to MAXIMIZE FITNESS.
- It can be any heuristic, e.g. in sudoku it can be the number of misplaced, or in 8-Queen problem it can be the number of queens that threatening each other.

NOTE ! Remember the rule you will make to calculate the fitness !
(If it is about how many correct = matching)
(if it is about how many mistakes = penalty)



Here, Fitness = 14

WHO IS THE BEST ?

Assume you are grading the midterm for 3 students. Then, you've obtained the number of mistakes for each student as the following:

	Student #1	Student #2	Student #3
No. of mistakes	4	7	1

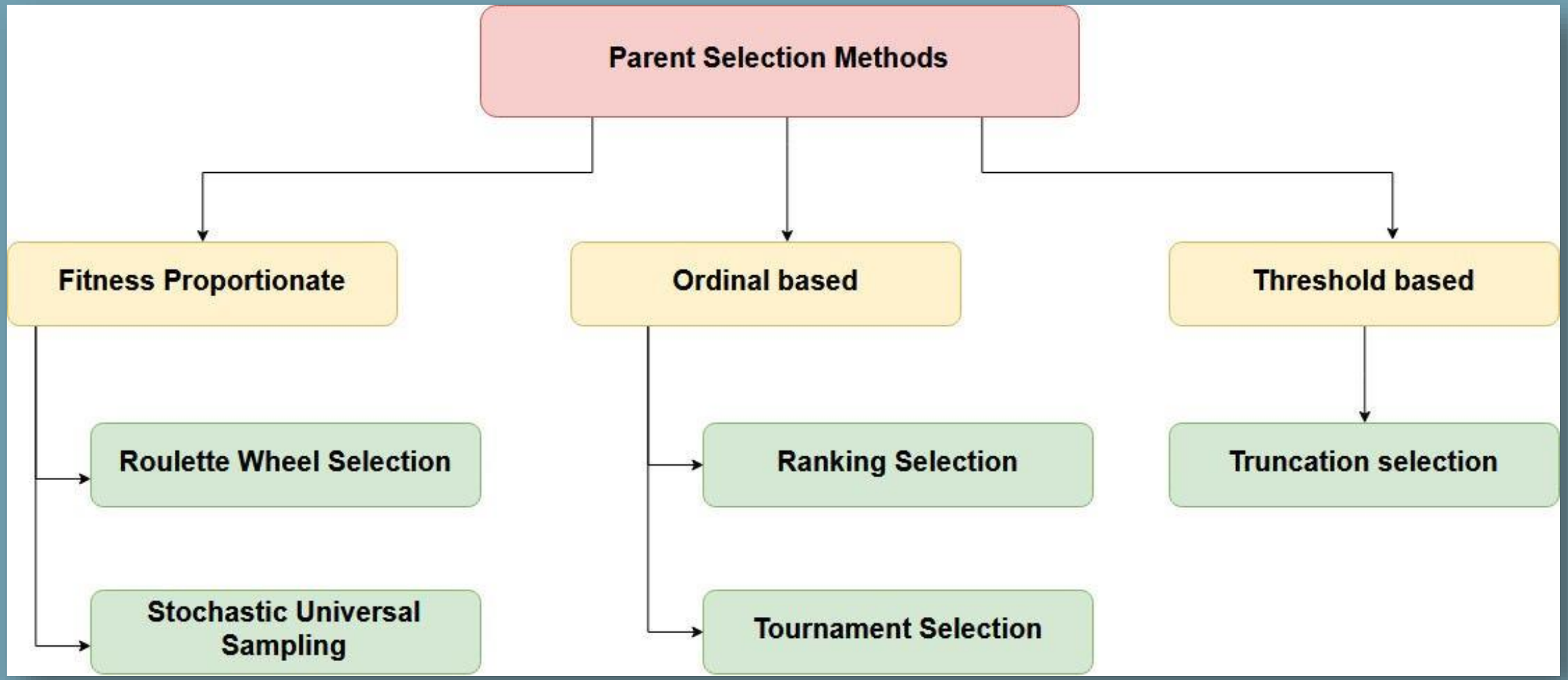
(3) SELECTION

- Choose the best individuals –in terms of CRITERIA- to "reproduce" based on their fitness. Better solutions have a higher chance of being selected.

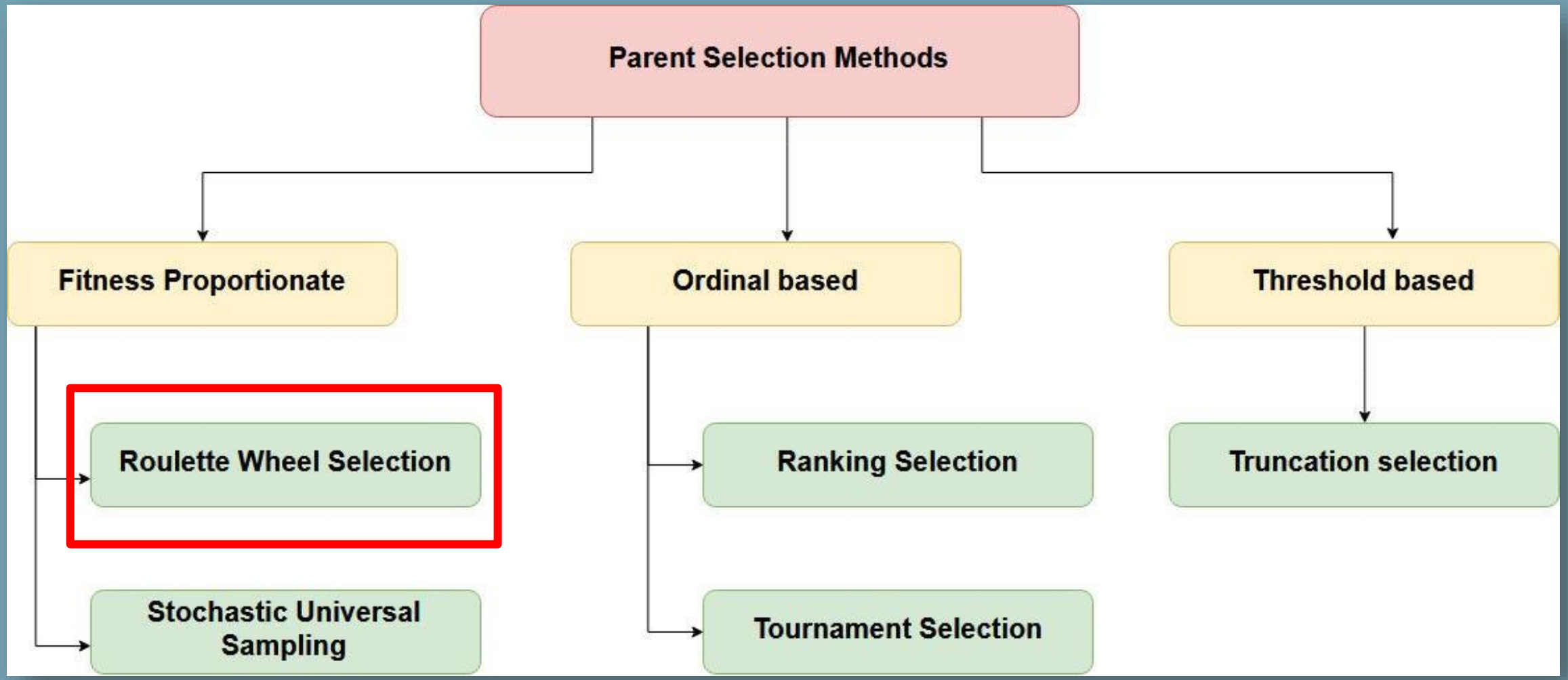
(3) SELECTION

- Choose the best individuals –in terms of CRITERIA- to "**reproduce**" based on their fitness. **Better solutions have a higher chance of being selected**.
- This could be considered as **Exploitation-Exploration tradeoff**, as there you are biased toward good solution

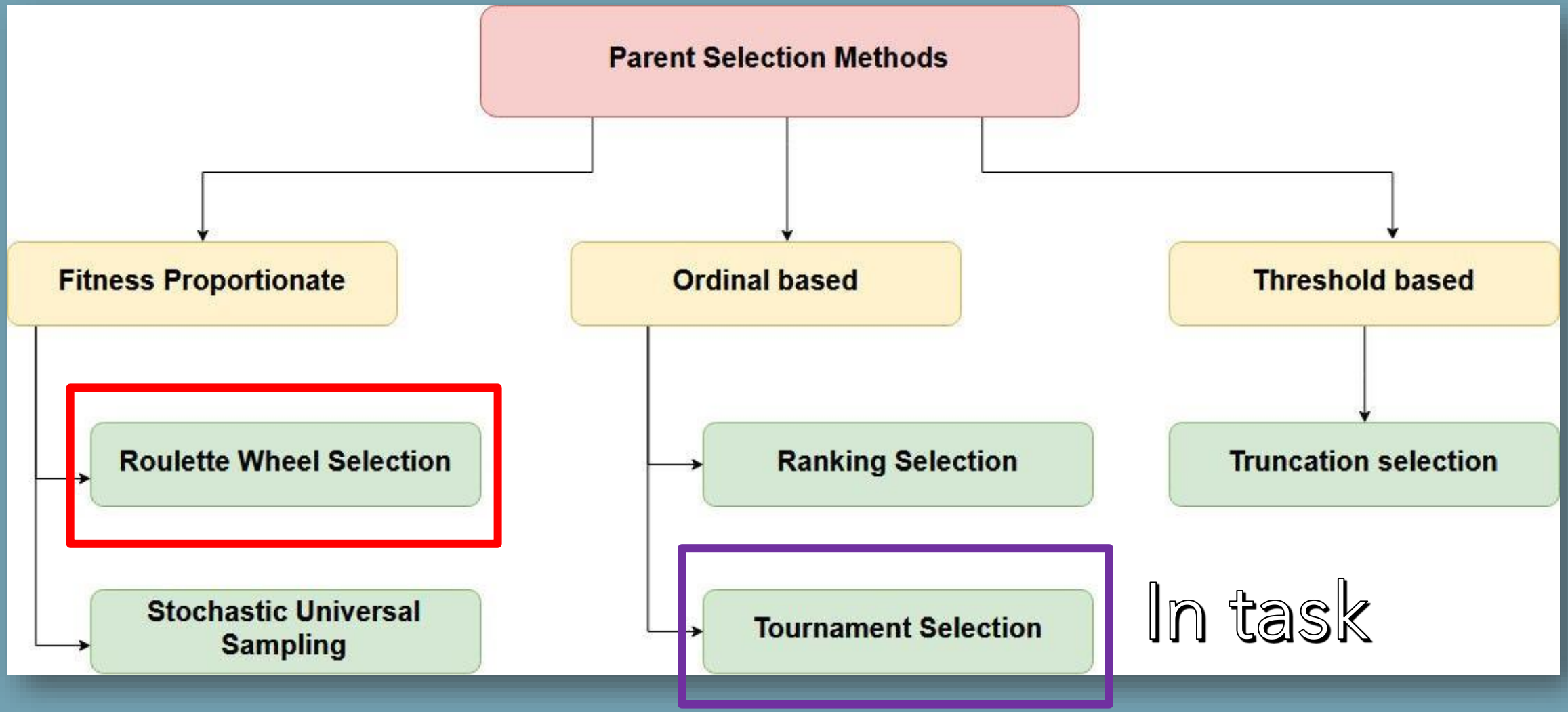
(3) SELECTION


Parent Selection Methods
Fitness Proportionate
Ordinal based
Threshold based
Roulette Wheel Selection
Stochastic Universal Sampling
Ranking Selection
Tournament Selection
Truncation selection

(3) SELECTION

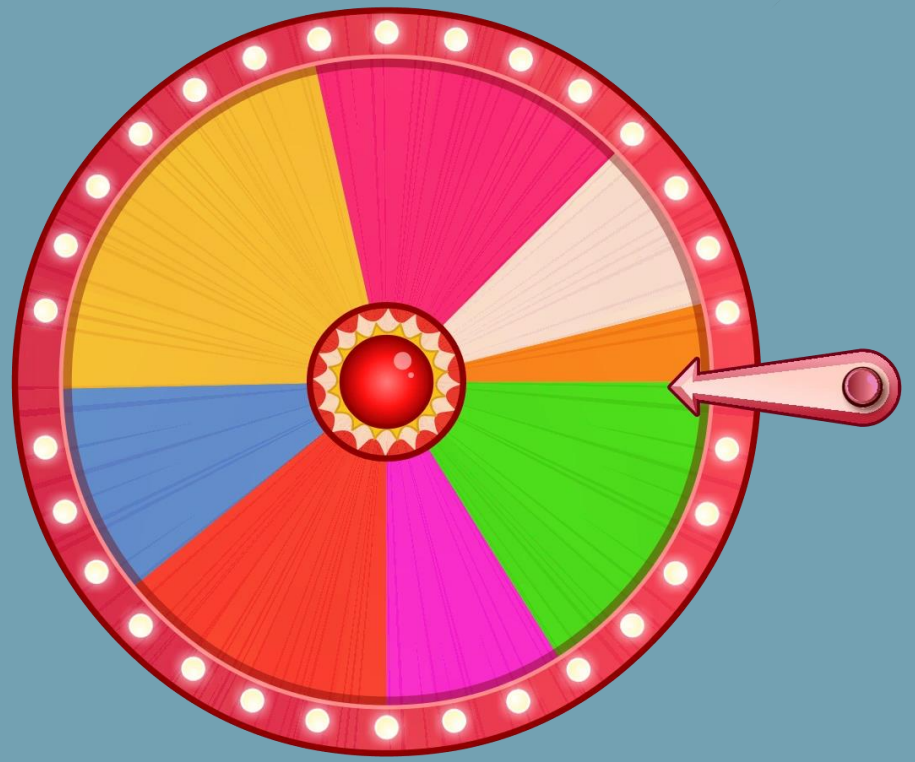

Parent Selection Methods
Fitness Proportionate
Ordinal based
Threshold based
Roulette Wheel Selection
Stochastic Universal Sampling
Ranking Selection
Tournament Selection
Truncation selection

(3) SELECTION


Parent Selection Methods
Fitness Proportionate
Ordinal based
Threshold based
Roulette Wheel Selection
Ranking Selection
Truncation selection
Stochastic Universal Sampling
Tournament Selection
In task

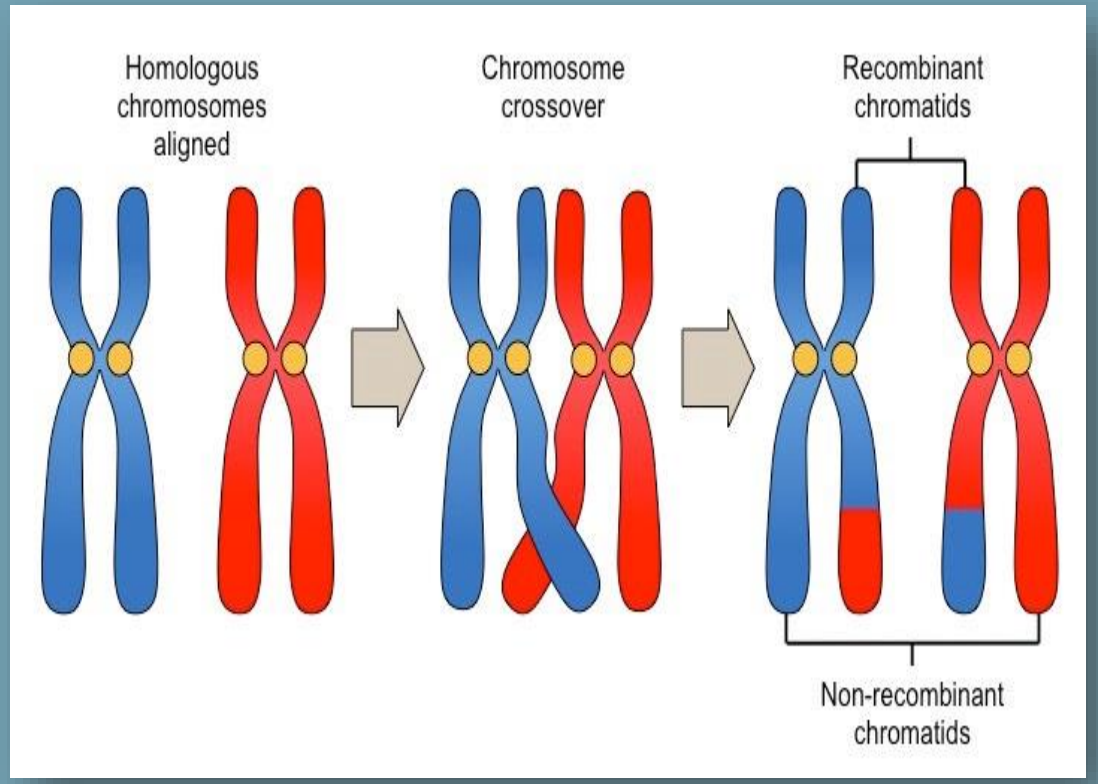
(3) SELECTION (ROULETTE WHEEL)

- Based on fitnesses we've collected, we need to put them on the wheel to select from.
- As you know, each sector on a –CIRCLE- is determined by its fraction OR PROBABILITY
- Convert each fitness into a probability (with keeping the best fitness with a greater probability) as the following (FITNESS VALUE/SUM OF ALL VALUES)
- Now, cumulate the probabilities until you reach to 1 and put them on the wheel. Finally, Spin the wheel !
- Based on the number of population you've chosen; you will select same number of times from the wheel WITH REPLACEMENT



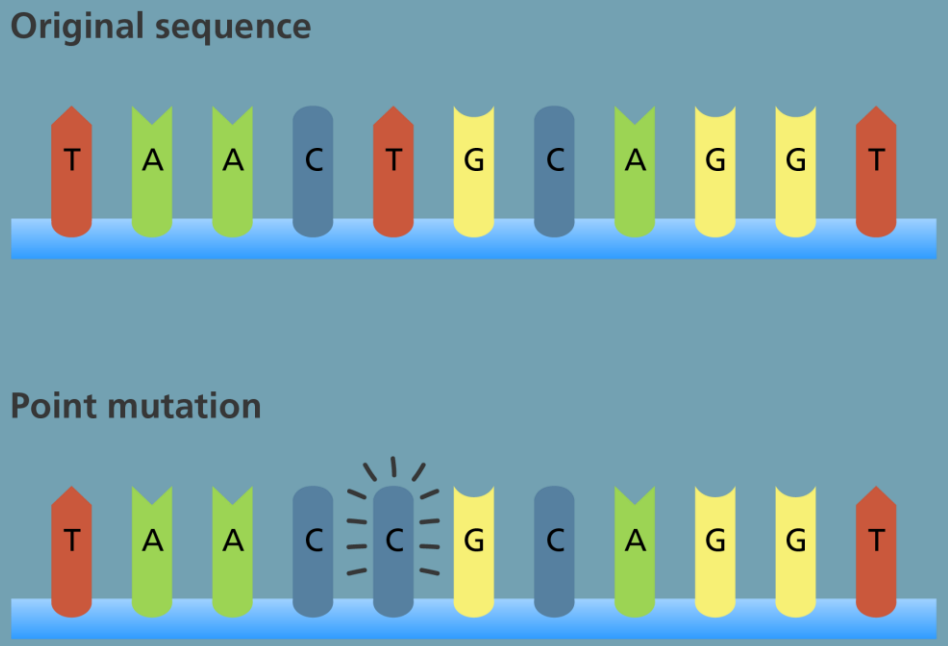
(4) CROSSOVER (EVOLUTIONARY PROCESS)

- Each two parents participate to produce new two childs **(offsprings)** by merging some parts of both parents
- That can introduce both exploration and exploitation, as the new childs can be very close to the goal, at the same time they aren't discovered before
- Basically, when two parents (individuals) have great fitness, they will produce childs with strong fitness too.
- There can be too many **combinations.**



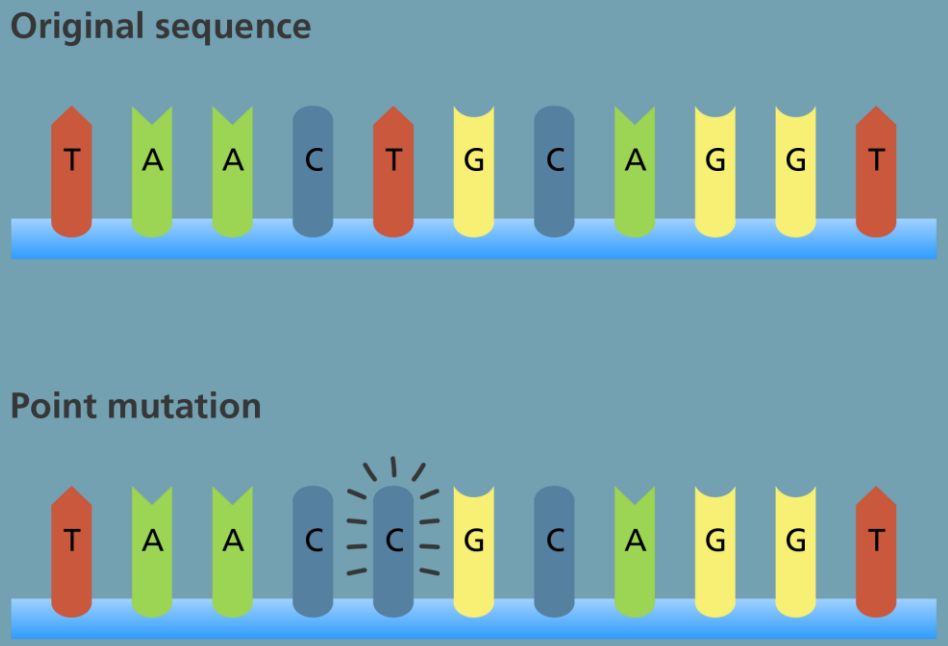
(5) MUTATION (طفرة جينية)

- Random change in genes, it RARELY happens.
- In the algorithm, each child may face mutation, so one if its values can be changed.
- This change can be substitution (modifying) or swapping for nueclutide.



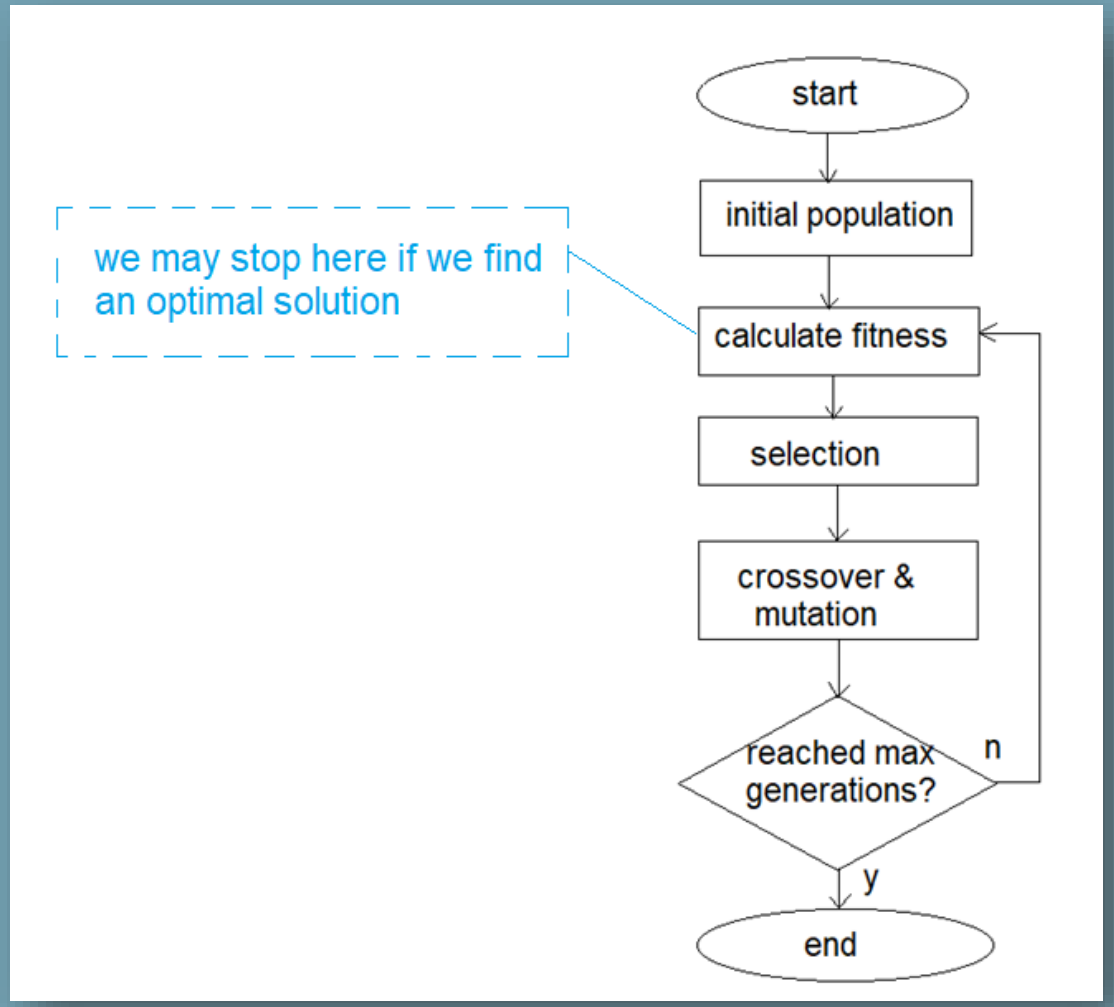
(5) MUTATION (طفرة جينية)

- Random change in genes, it RARELY happens.
- In the algorithm, each child may face mutation, so one if its values can be changed.
- This change can be substitution (modifying) or swapping for nueclutide.
- THIS IS THE REAL EXPLORATION WHICH INTRODUCE UNEXPLORED SOLUTION

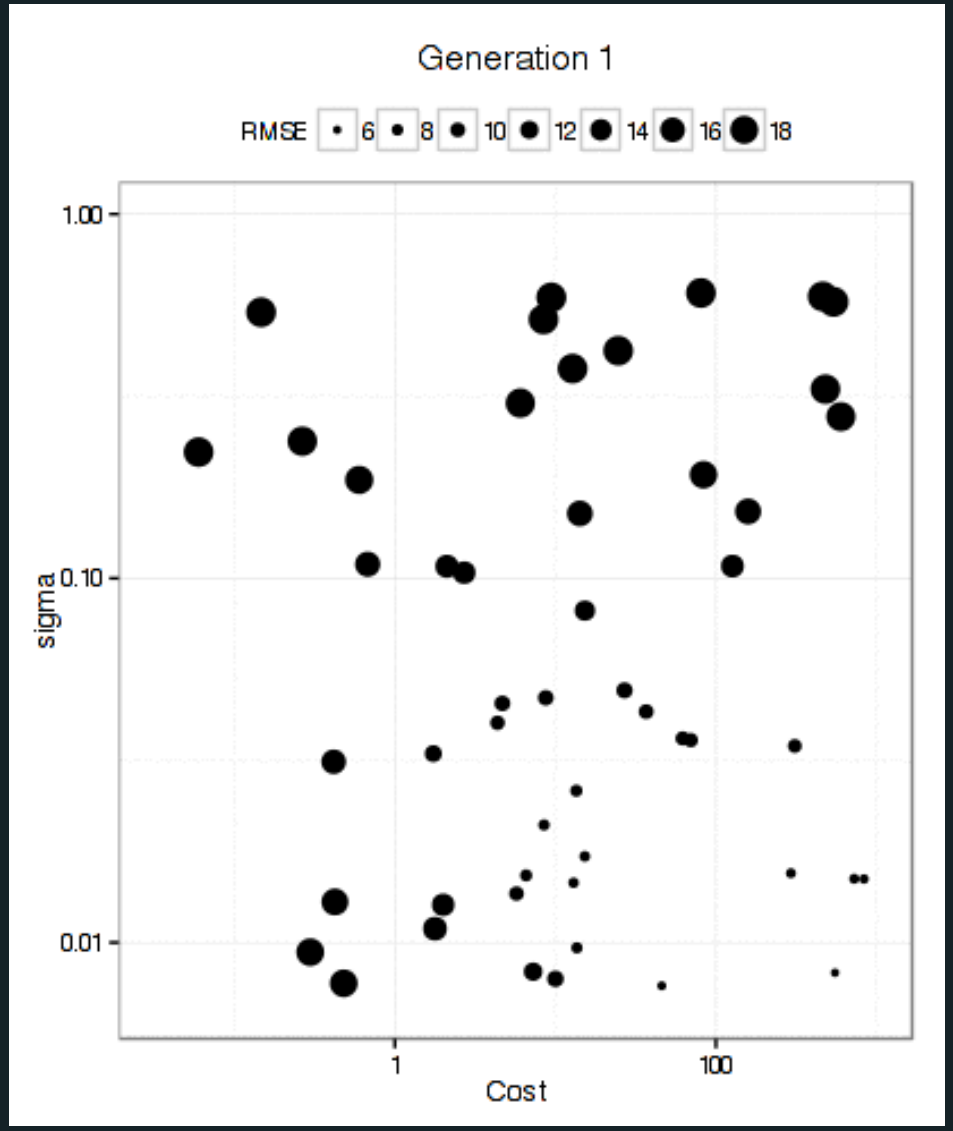


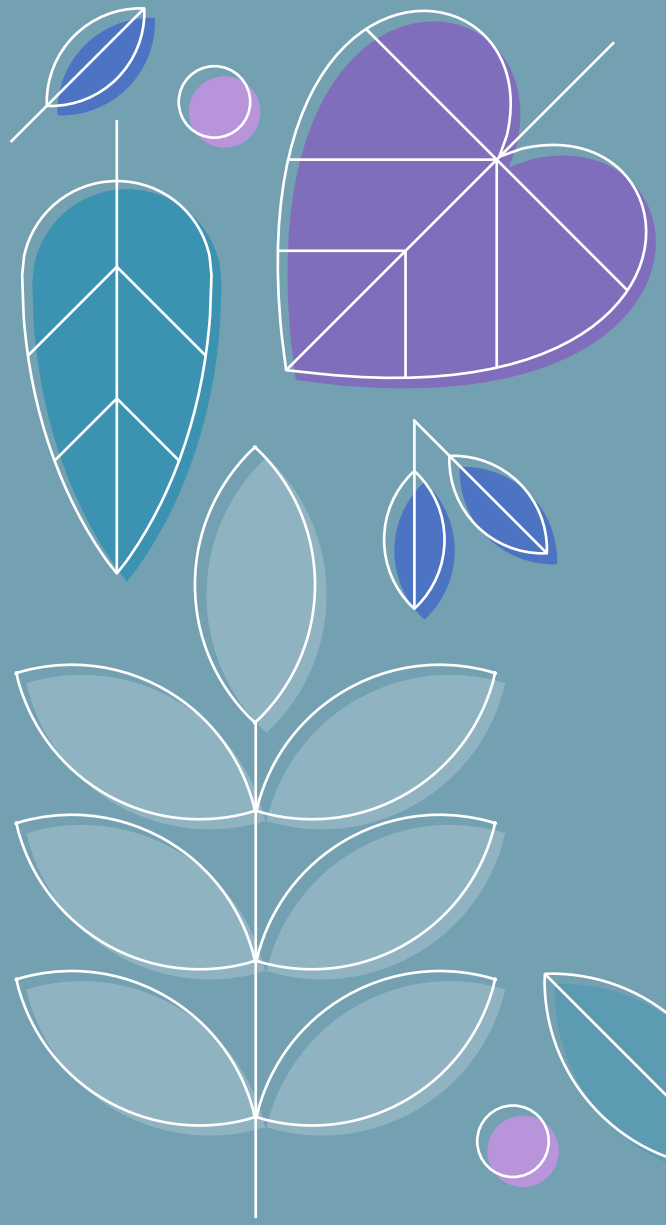
(6) REPEAT (STOPPING CRITERIA)

- Repeat from the step **“Calculate fitness”**
- Each iteration called **“Generation”** which can be a large number to guarantee a solution.
- You whether stop when reaching the maximum generations, or reaching the best solution you could ever obtain.



PERFORMANCE





NEXT LAB: GENETIC ALGORITHM PART (2)

Thank you