

Evolutionary Computation (Genetic Algorithms)

Dr. Mohamed Mostafa

AI 511 Foundations of Artificial Intelligence

Lecture 7



Outline

- Main Motivation for Heuristic Techniques
- Evolutionary Algorithms: OVERVIEW
- Genetic Algorithms (GA) OVERVIEW
 - Genetic Algorithm Pseudocode
 - Genotype Representation
 - Initial Population
 - Parent Selection
 - Crossover Operation
 - Mutation Operation
- Examples

Resources for this
lecture

- **Chapter 3 (Structures & Strategies for State Space Search; sections 3.2, and 3.3) and Chapter 12 (Genetic algorithms and evolutionary approaches)** *from* **George F. Luger, "Artificial Intelligence: Structures and strategies for complex problem solving, " Sixth edition (2009), Pearson Education Limited..**

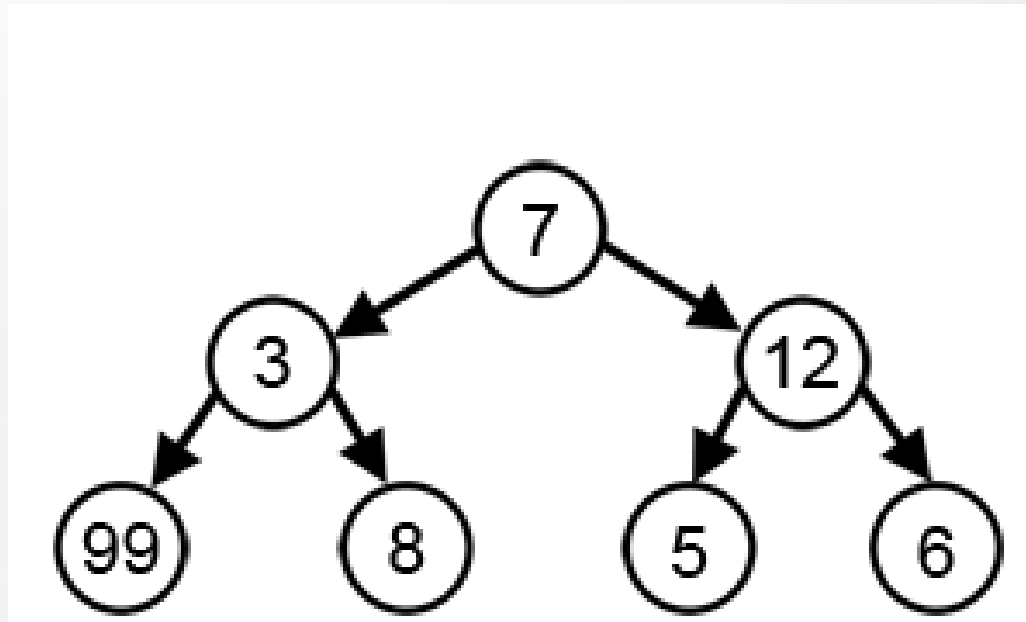
Main Motivation for Heuristic Techniques:

- **Efficiency:** Heuristic techniques aim to provide solutions to complex problems in a timely manner, especially when finding an optimal solution is computationally infeasible or too time-consuming.

Example: For instance, consider the problem of finding the shortest path between multiple points in a graph. The optimal solution to this problem can be obtained using algorithms like Dijkstra's algorithm or A* search, which guarantee finding the shortest path. However, these algorithms can be computationally expensive, especially for large graphs.

In contrast, the Greedy Algorithm provides a heuristic approach to solving this problem. It works by selecting the nearest neighbor at each step, without considering the global optimal solution.

Main Motivation for Heuristic Techniques:



Main Motivation for Heuristic Techniques:

- **Scalability:** Heuristic techniques are designed to handle large problem instances efficiently, making them suitable for real-world applications with vast amounts of data or complex problem spaces.

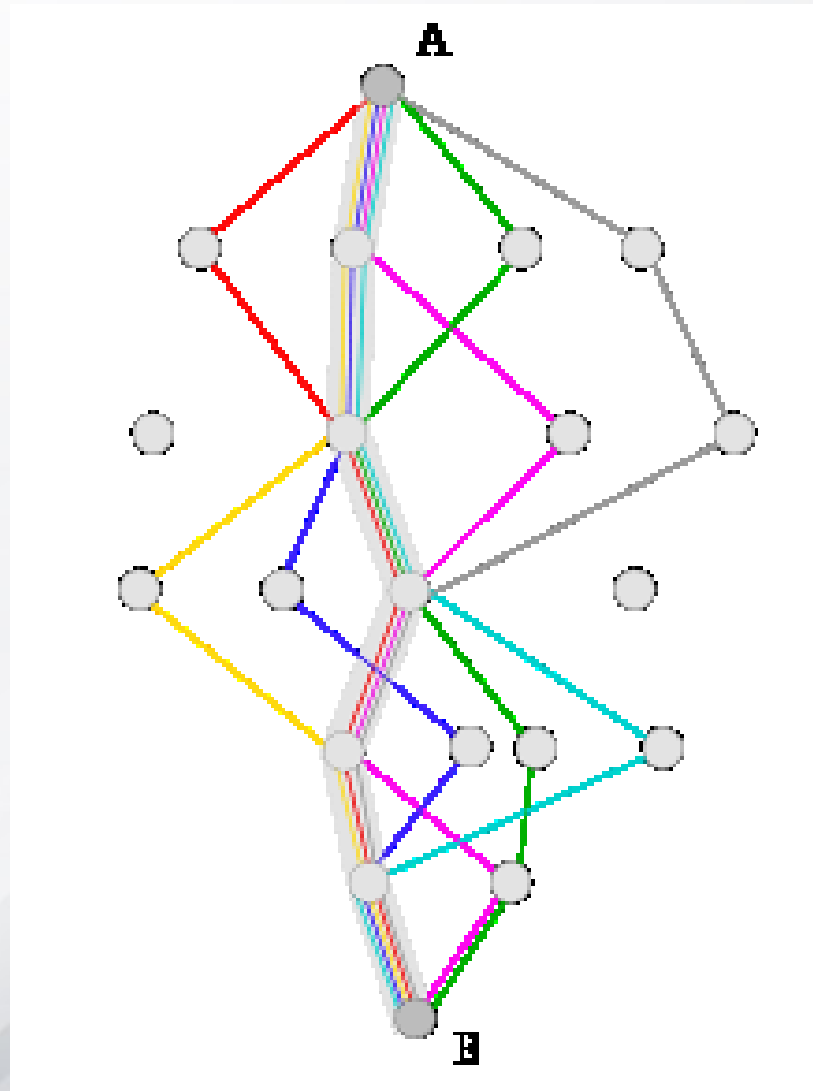
Example: Genetic algorithms are heuristic techniques used in optimization problems with a large search space, such as finding optimal solutions in scheduling, routing, or evolutionary biology.

Main Motivation for Heuristic Techniques:

- **Adaptability:** Heuristic techniques can adapt to different problem domains and problem instances, making them versatile tools for addressing a wide range of complex problems.

Example: Ant Colony Optimization (ACO) algorithms are heuristic techniques inspired by the foraging behavior of ants. ACO has been successfully applied to various optimization problems, including routing, scheduling, and telecommunications.

Main Motivation for Heuristic Techniques:



Main Motivation for Heuristic Techniques:

- **Resource Optimization:** Heuristic techniques help in minimizing computational resources such as time, memory, or energy consumption, while still providing reasonable solutions to complex problems.

Example: In the Knapsack Problem, you are given a set of items, each with a weight and a value, and a knapsack with a maximum weight capacity. The goal is to determine the combination of items to include in the knapsack that maximizes the total value while not exceeding the weight capacity.

The Greedy Algorithm can be applied to solve this problem by selecting items one at a time, based on their value-to-weight ratio, and adding them to the knapsack until the weight capacity is reached or all items are considered.



Evolutionary Algorithms: OVERVIEW

- Evolutionary Algorithms (EA) are a collection of methods developed for solving combinatorial optimization problems.
- They apply Darwinian principles to automated problem-solving, mimicking the process of natural evolution.
- EA is a subset of Evolutionary Computation, which falls under the umbrella of Artificial Intelligence (AI) or Computational Intelligence.

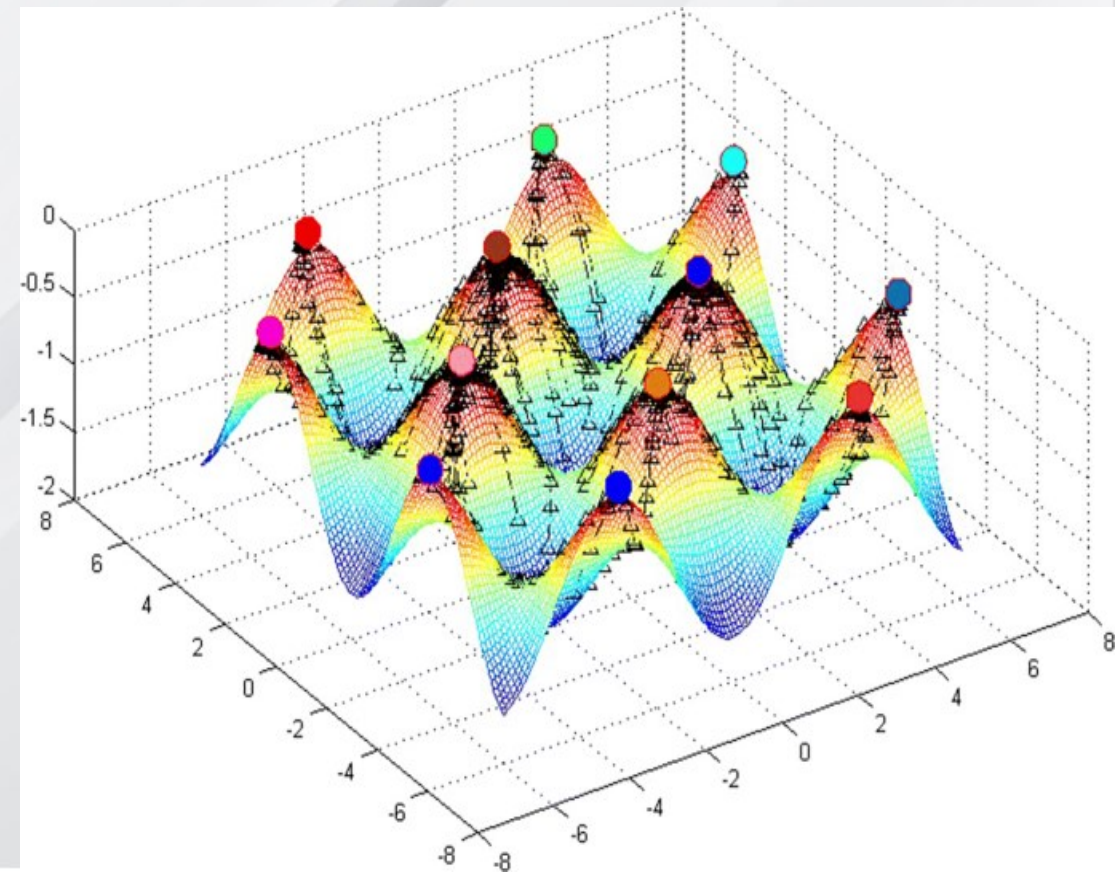
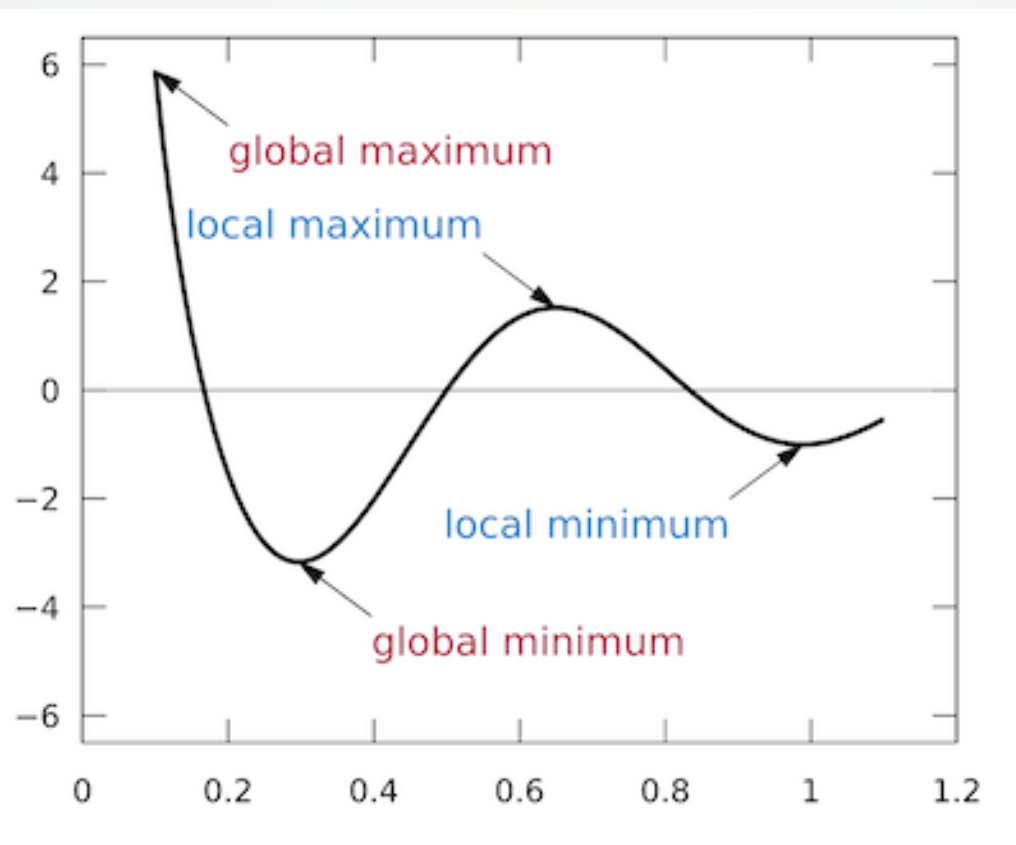
Evolutionary Algorithms: OVERVIEW

- Evolutionary Algorithms are metaheuristic optimization algorithms that rely on populations of candidate solutions inspired by natural evolution.
- Key components of Evolutionary Algorithms include:
 - A population or set of individuals representing candidate solutions to the optimization problem.
 - A problem-specific fitness function, which evaluates the quality of each individual solution (state).
 - Pairs of individuals are randomly selected for reproduction.
 - From each pair new offsprings (children) are generated using crossover operation.
 - A crossover point is chosen randomly from the positions in the string.
 - The offsprings are created by crossing over the parent strings at the crossover point.
 - Each child is subject to random mutation with a small independent probability.

Genetic Algorithms (GA) OVERVIEW

- The Genetic Algorithm (GA) is a search-based optimization approach inspired by genetic and natural selection principles (“survival of the fittest”).
- It is utilized to find optimal or near-optimal solutions to complex problems that are computationally challenging to solve using traditional methods.
- GA mimics the process of natural selection, employing mechanisms such as reproduction, mutation, and crossover to iteratively improve candidate solutions.
- GA is favored for its ability to efficiently explore large solution spaces and handle nonlinear, non-convex, and multi-modal optimization problems.

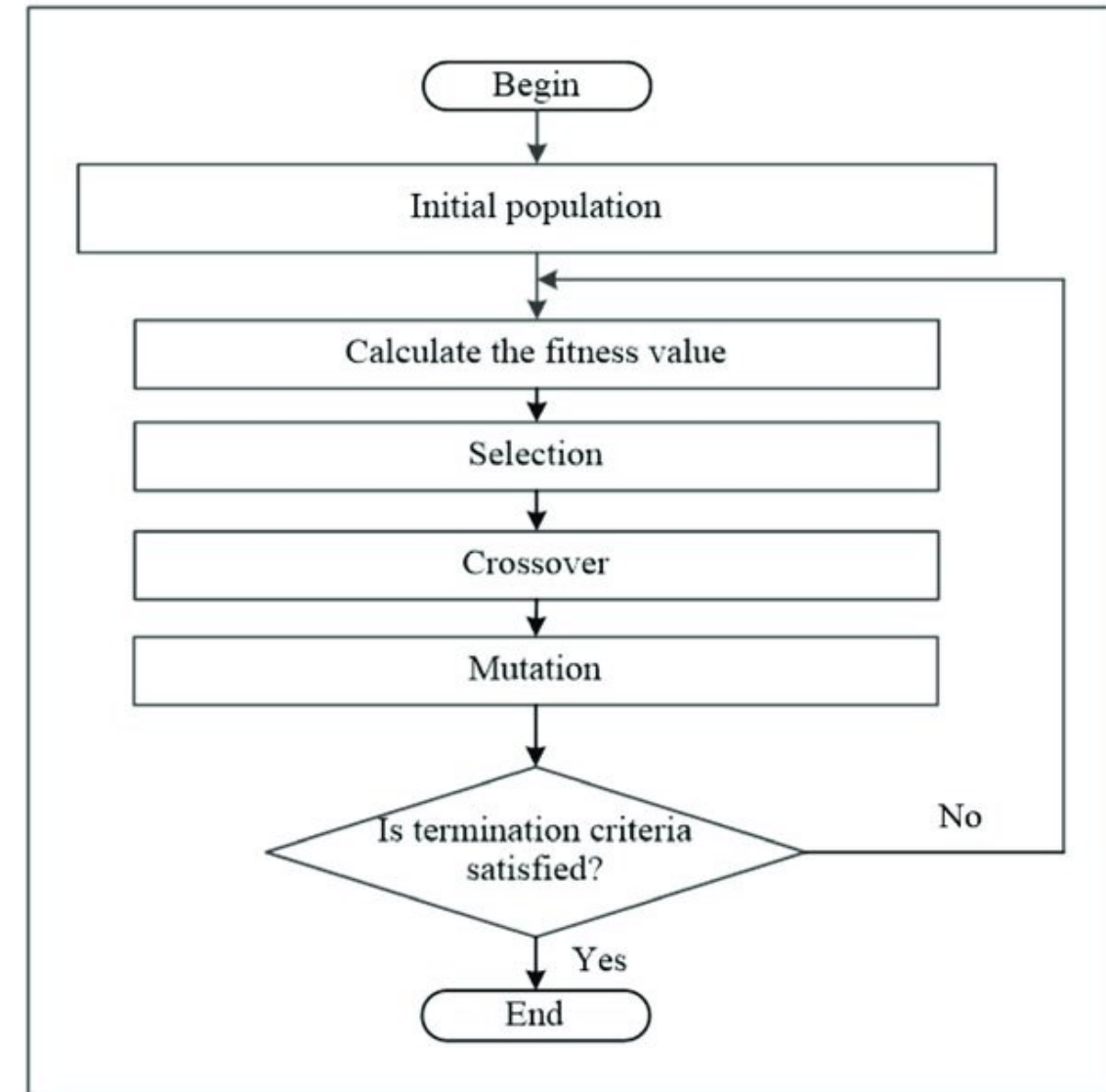
Genetic Algorithms (GA) OVERVIEW



Genetic Algorithms (GA) OVERVIEW

Steps in a Genetic Algorithm

- Initialize population
- Select parents by evaluating their fitness
- Crossover parents to reproduce
- Mutate the offsprings
- Evaluate the offsprings
- Merge offsprings with the main population and sort



Genetic Algorithm Pseudocode

Genetic Algorithm

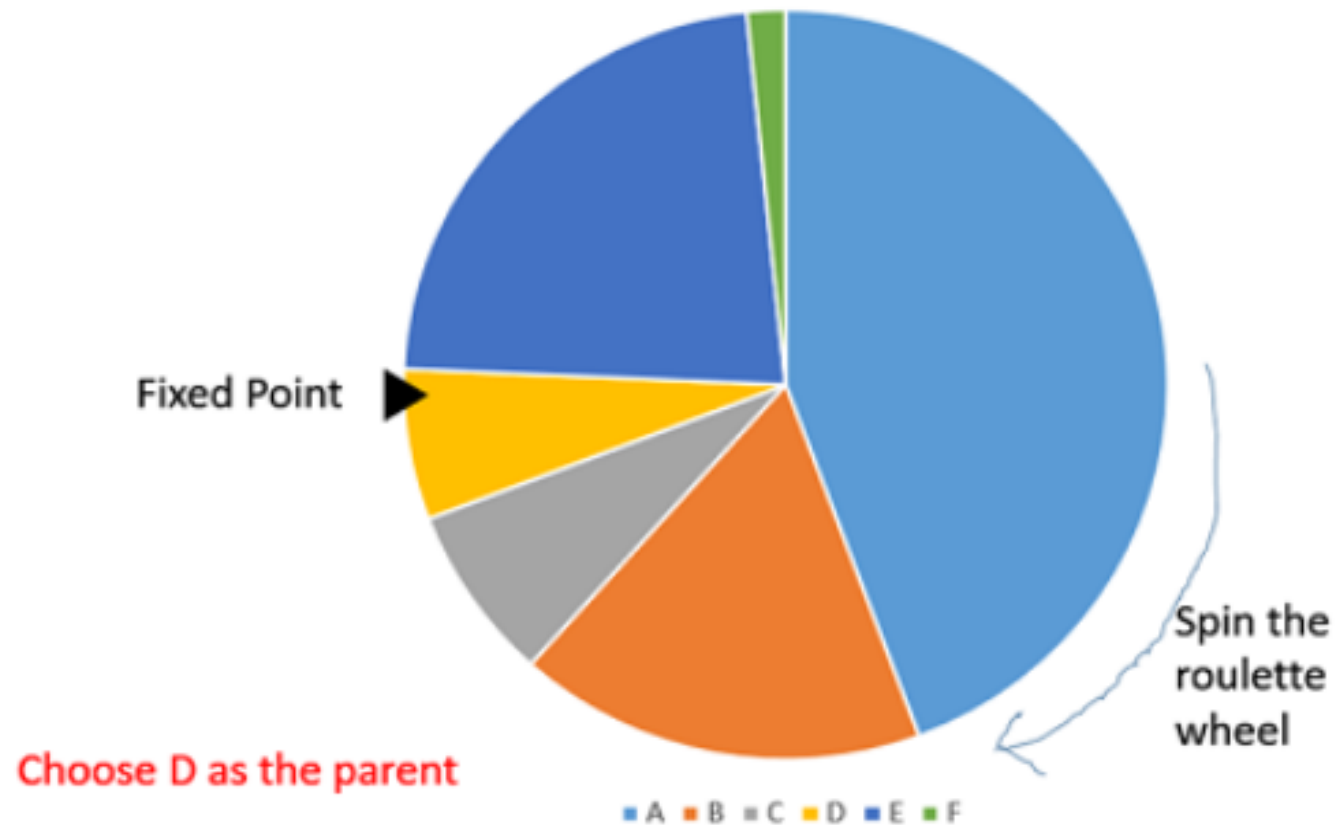
```

{
    Generate initial population  $P_0$ 
    Evaluate population  $P_0$ 
    While (stopping GA criteria not satisfied) Repeat
        {
            For 1 to (number of tournaments)
                {
                    Select  $N_t$  chromosomes for tournament
                    Find chromosome with lowest fitness
                    Remove chromosome with lowest fitness
                    Crossover (Create new chromosome)
                    Evaluate new chromosome
                }
            Mutation
            Evaluate (mutated chromosomes)
        }
    }

```

Parent Selection

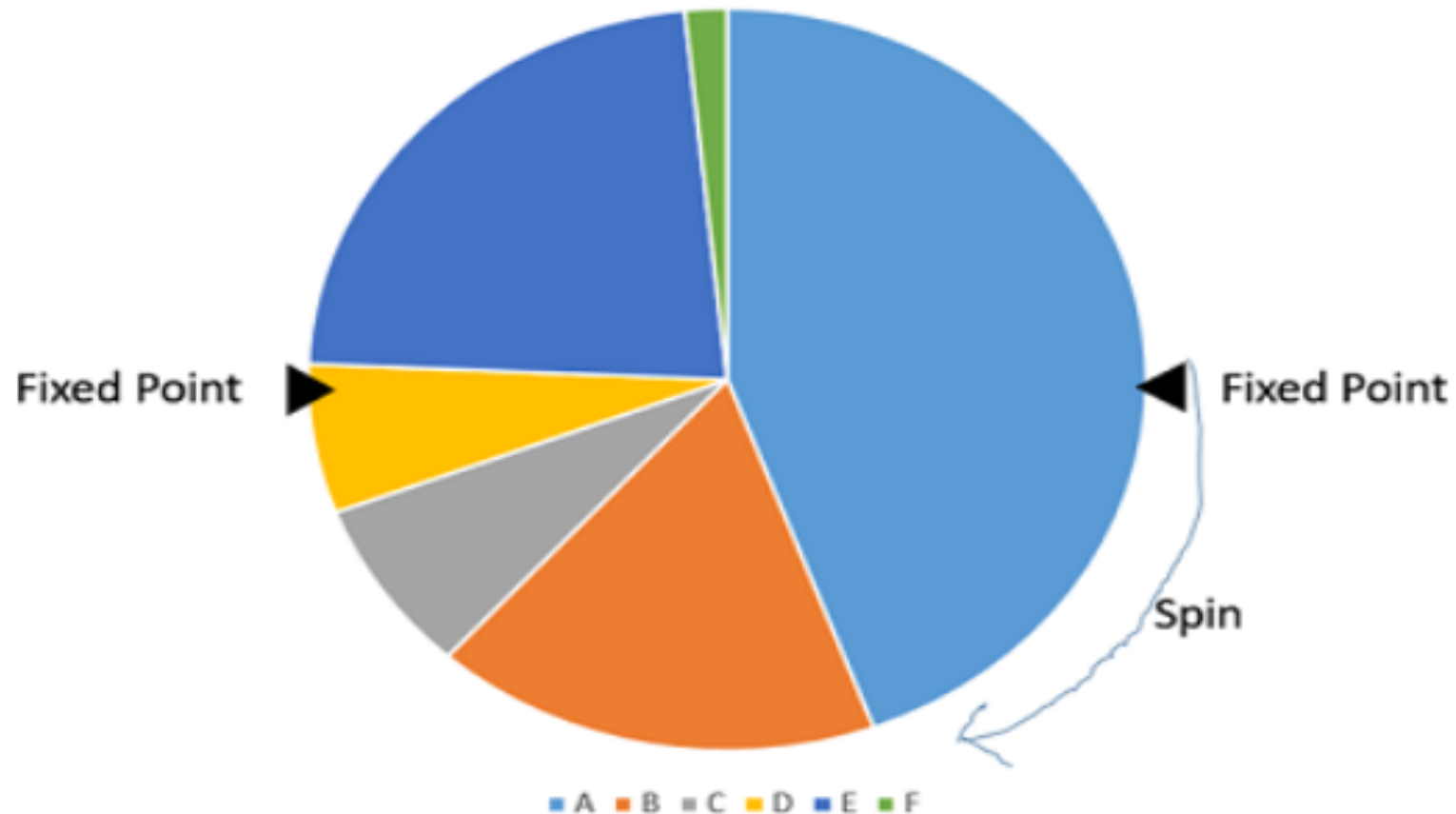
- Roulette Wheel Selection



Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

Parent Selection

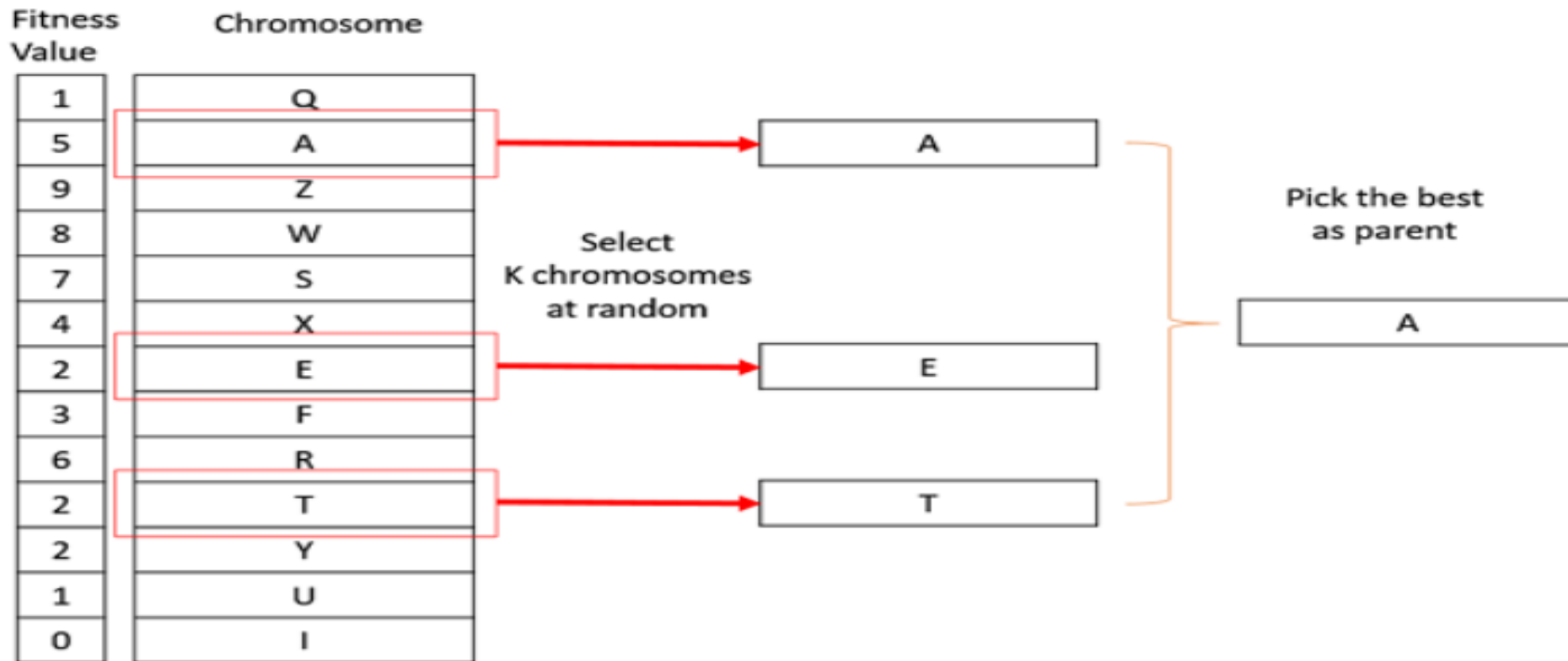
- Stochastic Universal Sampling (SUS)



Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

Parent Selection

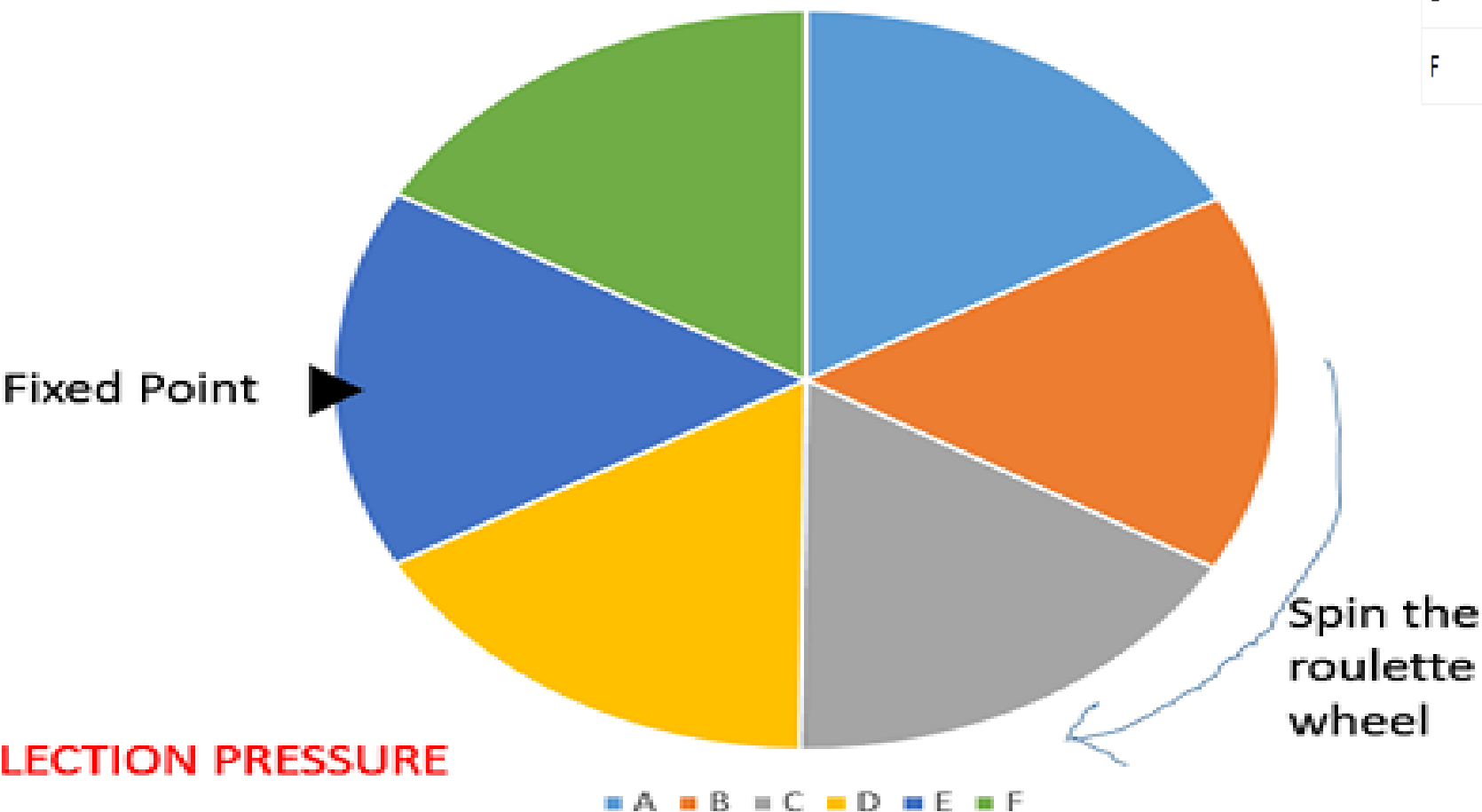
- Tournament Selection



Parent Selection

- Rank Selection

Chromosome	Fitness Value	Rank
A	8.1	1
B	8.0	4
C	8.05	2
D	7.95	6
E	8.02	3
F	7.99	5

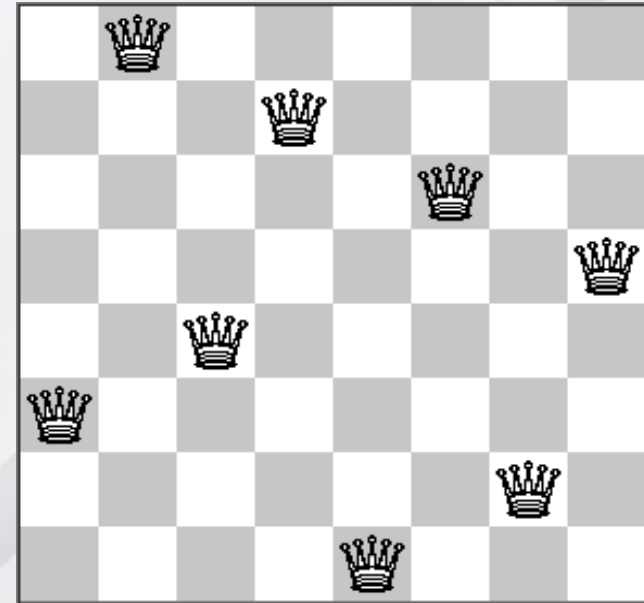


Chromosome	Fitness Value
A	8.1
B	8.0
C	8.05
D	7.95
E	8.02
F	7.99

Genotype Representation

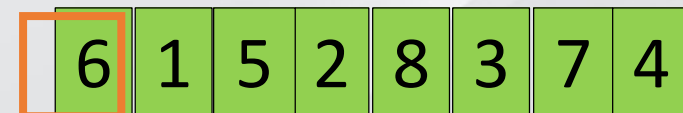
Phenotype

- **Fitness Function :**
the sum of the penalties of all queens. (Minimization)



Genotype

gene



chromosome

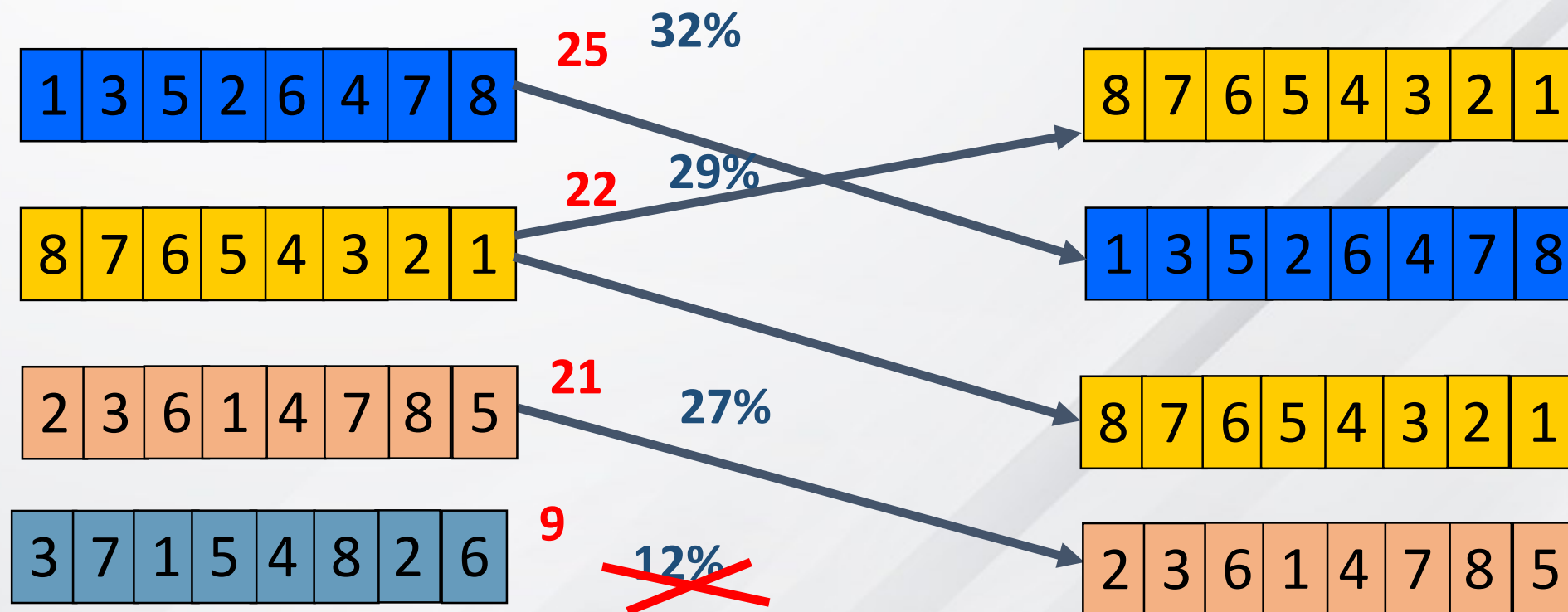
Initial Population

1	3	5	2	6	4	7	8	25
8	7	6	5	4	3	2	1	22
2	3	6	1	4	7	8	5	21
3	7	1	5	4	8	2	6	9

The **initial population** has 4 states.

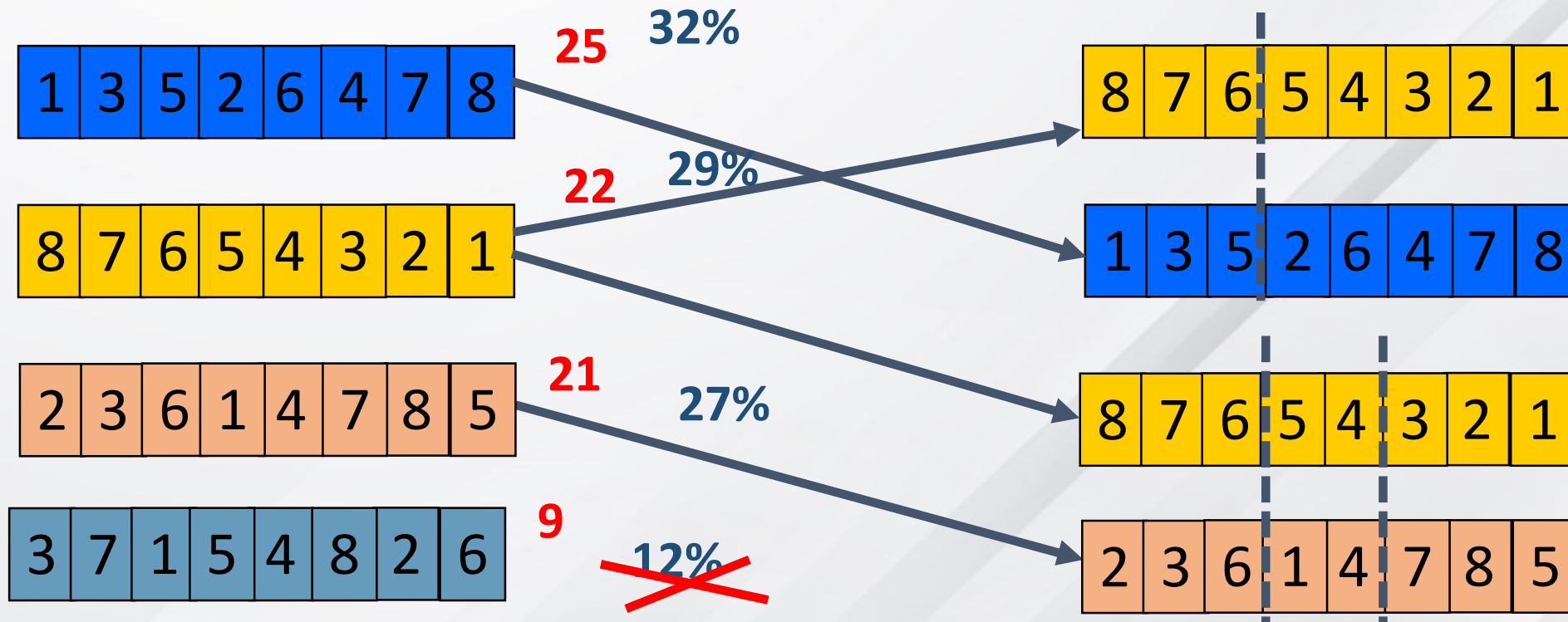
- A fitness function should return higher values for better states, so, for the 8-queens problem we use the number of non-attacking pairs of queens, which has a value of 28 for a solution.
- The values of the four states are 25, 22, 21, and 9.

Parent Selection



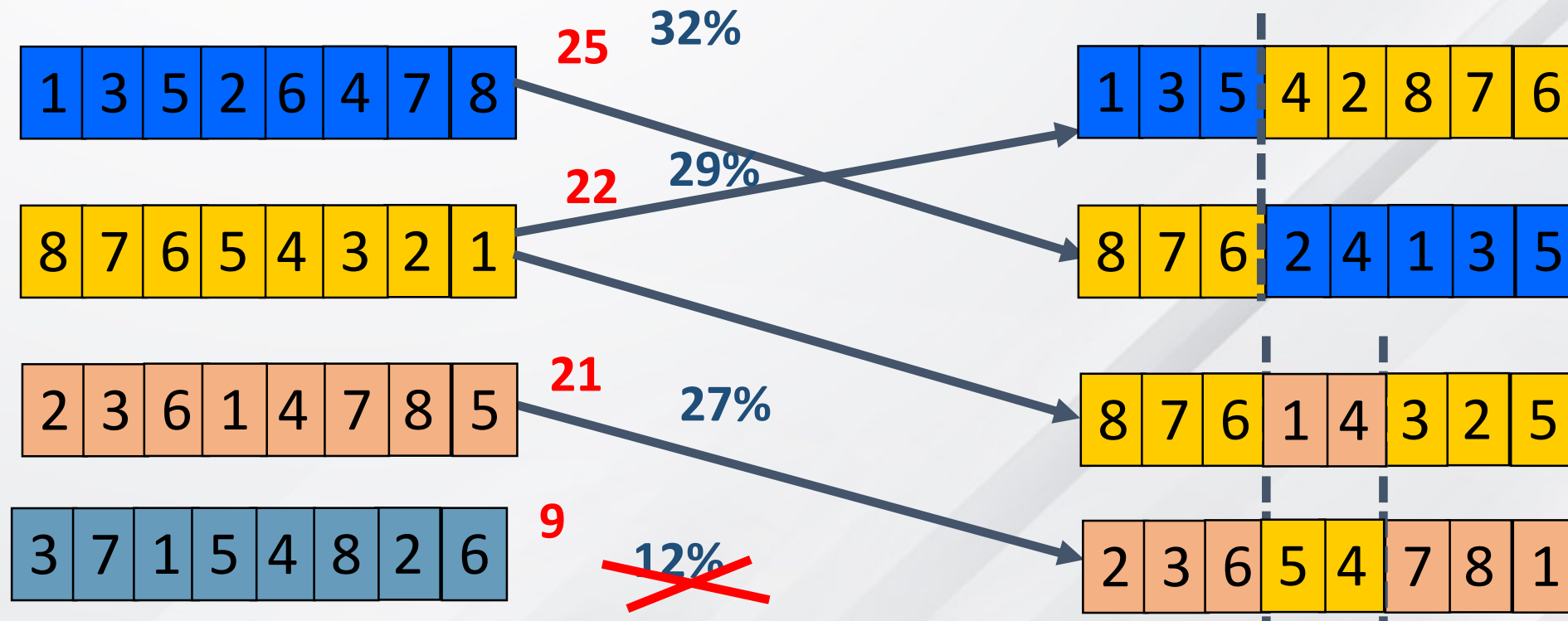
The **probability of being chosen for reproducing** is directly proportional to the fitness score. • Two pairs are selected at random for reproduction, in accordance with the probabilities. • Notice that one individual is selected twice and one not at all.

Crossover Operation



The **probability of being chosen for reproducing** is directly proportional to the fitness score. • Two pairs are selected at random for reproduction, in accordance with the probabilities. • Notice that one individual is selected twice and one not at all.

Crossover Operation



The **probability of being chosen for reproducing** is directly proportional to the fitness score. • Two pairs are selected at random for reproduction, in accordance with the probabilities. • Notice that one individual is selected twice and one not at all.

Mutation Operation



One digit was **mutated** in the first, third, and fourth offspring.

- In the 8-queens problem, this corresponds to choosing a queen at random and moving it to a random square in its column.

Genotype Representation

Phenotype

O		X
X	X	O
O		



Genotype

1	0	2	2	2	1	1	0
---	---	---	---	---	---	---	---

Crossover or Mutation

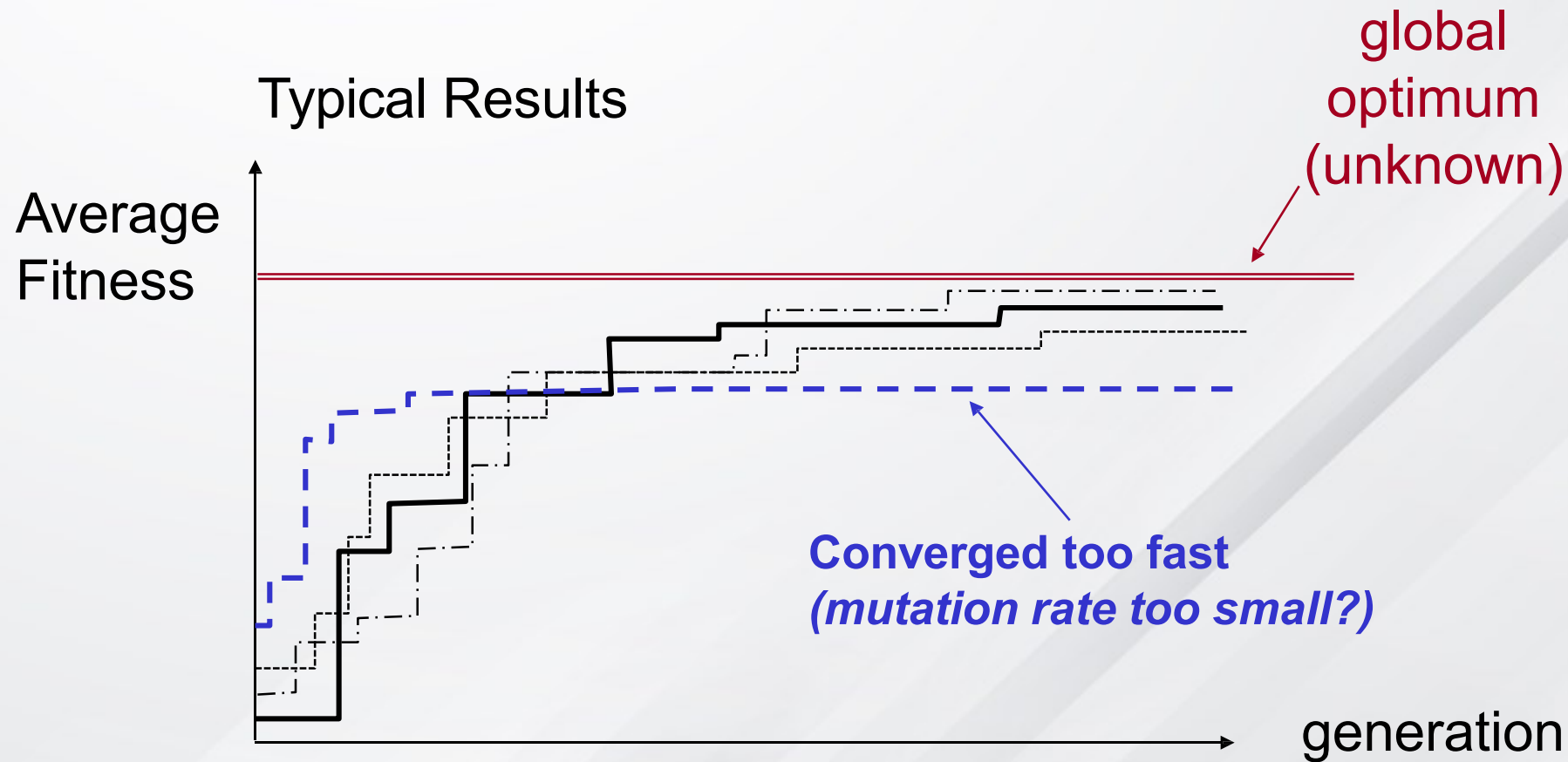
- **Exploration:**

- Discovering promising areas in the search space, i.e. gaining information on the problem
- Crossover is “usually” explorative, it makes a big jump to an area somewhere “in between” two (parent) areas.

- **Exploitation:**

- Optimizing within a promising area, i.e. using information
- Mutation is “usually” exploitative, it creates random small diversions, thereby staying near (in the area of) the parent

GA Convergence



Average performance of individuals in a population is expected to increase, as good individuals are preserved and bred and less fit individuals die out.

GA Example

- Find the maximum of a function $f(x)=x^2$, where the value of x range from 0 to 31

Step 1: Encoding to Genotype

0 →

0	0	0	0	0
---	---	---	---	---

31 →

1	1	1	1	1
---	---	---	---	---

GA Example

- Find the maximum of a function $f(x)=x^2$, where the value of x range from 0 to 31

Step 2: Determine the population size

$$N = 4$$

GA Example

- Find the maximum of a function $f(x)=x^2$, where the value of x range from 0 to 31

Step 3: Randomly choose initial populations

15, 25, 3, 17

GA Example

- Find the maximum of a function $f(x)=x^2$, where the value of x range from 0 to 31

Step 4: Select best parents for the next offsprings

No	Initial	encoded	$f(x)$	Probability count	
1	15	01111	$15*15=225$		
2	25	11001	$25*25= 625$		
3	3	00011	$3*3= 9$		
4	17	10001	$17*17= 289$		
Total			$=1148$		

GA Example

- Find the maximum of a function $f(x)=x^2$, where the value of x range from 0 to 31

Step 4: Select best parents for the next offsprings

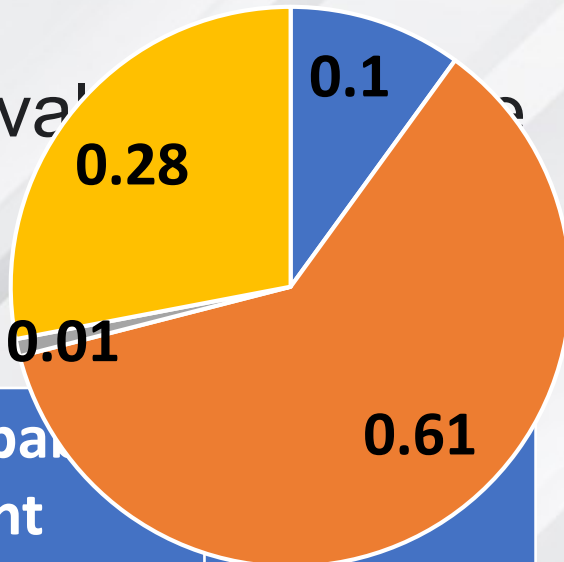
No	Initial	encoded	$f(x)$	Probability Count	
1	15	01111	$15*15=225$	0.2	
2	25	11001	$25*25= 625$	0.54	
3	3	00011	$3*3= 9$	0.01	
4	17	10001	$17*17= 289$	0.25	
Total			$=1148$	1	

Prob Count

GA Example

- Find the maximum of a function $f(x)=x^2$, where the value of x is from 0 to 31

Step 4: Select best parents for the next offsprings



No	Initial	encoded	f(x)	Probal count	
1	15	01111	15*15=225	0.2	1
2	25	11001	25*25= 625	0.54	2
3	3	00011	3*3= 9	0.01	3
4	17	10001	17*17= 289	0.25	4
Total			=1148	1	

GA Example

- Find the maximum of a function $f(x)=x^2$, where the value of x range from 0 to 31

Step 4: Select best parents for the next offsprings

No	Initial	encoded	f(x)	Probability Count	Expected Count
1	15	01111	$15*15=225$	0.2	$225/287=0.8$
2	25	11001	$25*25= 625$	0.54	2.2
3	3	00011	$3*3= 9$	0.01	0.03
4	17	10001	$17*17= 289$	0.25	1.0
Total			=1148	1	4
Average			287		

GA Example

- Find the maximum of a function $f(x)=x^2$, where the value of x range from 0 to 31

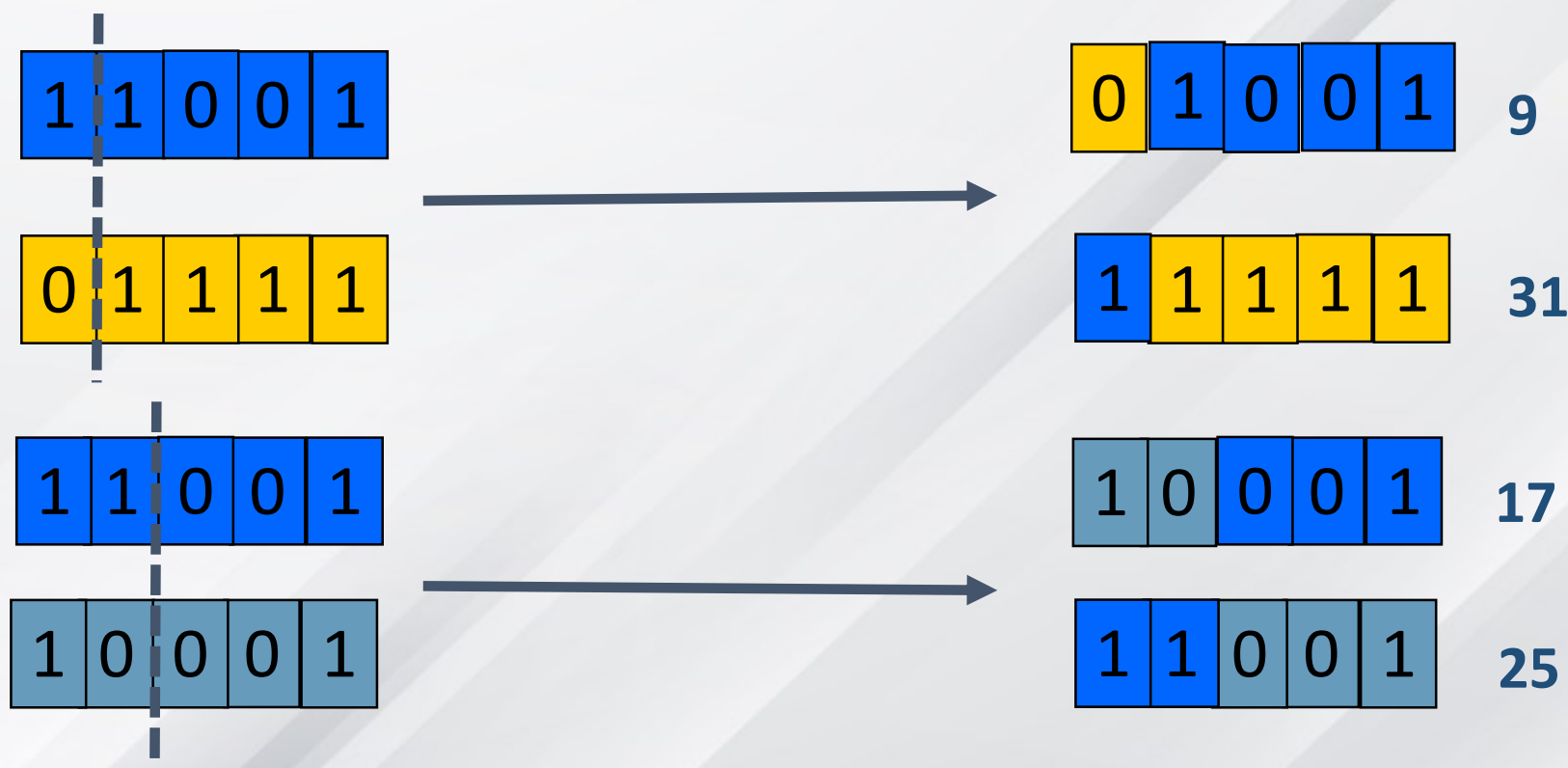
Step 4: Select best parents for the next offsprings

No	Initial	encoded	$f(x)$	Probability Count	Expected Count	Selection count
1	15	01111	$15*15=225$	0.2	$225/287=0.8$	1
2	25	11001	$25*25= 625$	0.54	2.2	2
3	3	00011	$3*3= 9$	0.01	0.03	0
4	17	10001	$17*17= 289$	0.25	1.0	1
Total			=1148	1	4	
Average			256			

GA Example

- Find the maximum of a function $f(x)=x^2$, where the value of x range from 0 to 31

Step 5: Crossover and Mutation for the Generating next offsprings



GA Example

- Find the maximum of a function $f(x)=x^2$, where the value of x range from 0 to 31

Step 5: Evaluating offsprings

No	Initial	encoded	$f(x)$	Probability count	Expected Count
1	9	01001	$9*9=81$		
2	31	11111	961		
3	25	11001	625		
4	17	10001	$17*17= 289$		
Total			=1956		

Additional Resources

- The Solving of the TSP using GA

https://www.researchgate.net/publication/235902844_An_Improved_Genetic_Algorithm_to_Solve_the_Traveling_Salesman_Problem



مجموعة جامعة
الجميع الذكية
للتعليم القابضة



Thank You

