Artificial Intelligence

Beyond Classical Search

Genetic algorithms

- A genetic algorithm is a variant of stochastic beam search in which successor states are generated by combining two parent states rather than by modifying a single state.
- A successor state is generated by combining two parent states
- Start with *k* randomly generated states (population)
- A state is represented as a string over a finite alphabet (often a string of 0s and 1s)
- Evaluation function (fitness function). Higher values for better states.
- Produce the next generation of states by selection, *crossover*, and *mutation* (*genetic operators*)

Biology Concepts

- Population
- Fitness
- Selection
- Crossover
- Mutation

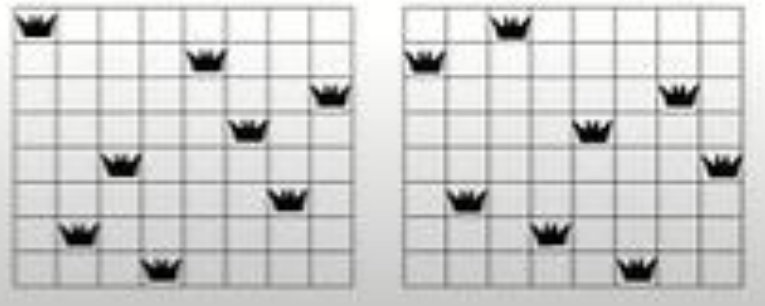
Biology Concepts: Population

Biology

- Collection of individuals.

Algorithm

- Collection of states.



Biology Concepts: Fitness

Biology

- More healthy, less prone to diseases.

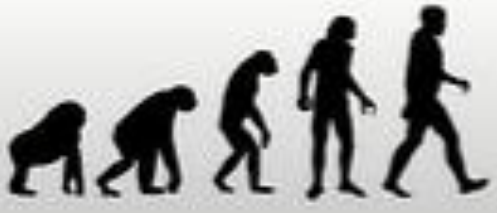
Algorithm

- Closest to the final solution.

Biology Concepts: Selection

Biology

- Selecting species that are the most biologically fit.



Algorithm

- Selecting states that are closest to the solution (Fittest).

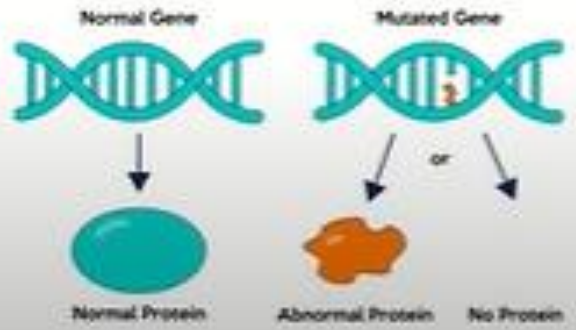
Biology Concepts: Crossover

Biology	Algorithm
generation	Interchanging values between selected states

Biology Concepts: Mutation

Biology

- Change or variation.



Algorithm

- Alteration.

Before Mutation

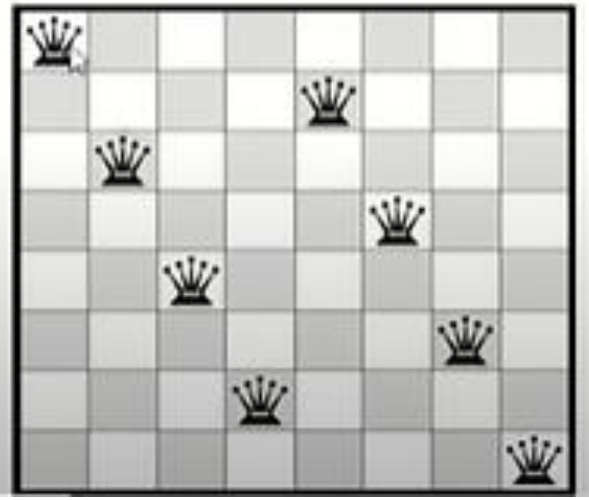
A5	1	1	1	0	0	0

After Mutation

A5	1	1	0	1	1	0

8- Queens Problem

- Arrange 8 queens on a standard chess board in such a way that no queen attacks each other.



Solving 8-Queens using Genetic Algorithm

- Step 1: Representing individuals.
- Step 2: Generating an initial Population.
- Step 3: Applying a Fitness Function.
- Step 4: Selecting parents for mating in accordance to their fitness.
- Step 5: Crossover of parents to produce new generation.
- Step 6: Mutation of new generation to bring diversity.
- Step 7: Repeat until solution is reached.

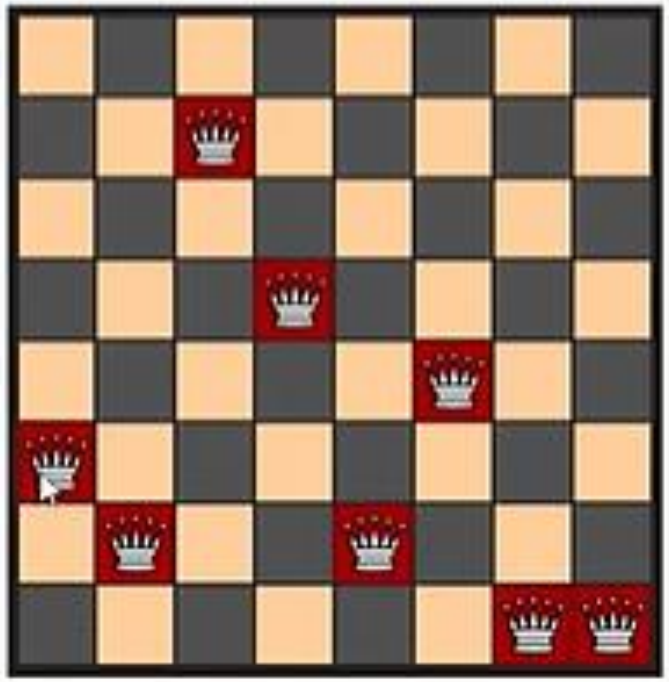
Step 1: Representing Individuals/States

- Formulate an appropriate method to represent individuals of a population.
- Array.
- Index: Column.
- Value: Row.

1 - 8

3	2	7	5	2	4	1	1

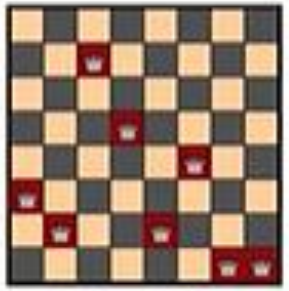
9



Step 2: Generate Initial Population

- Generate random arrangements of 8 queens on a standard chess board.

A



3	2	7	5	2	4	1	1

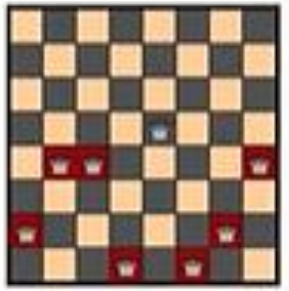
B

2	4	7	4	8	5	5	2

C

3	2	5	4	3	2	1	3

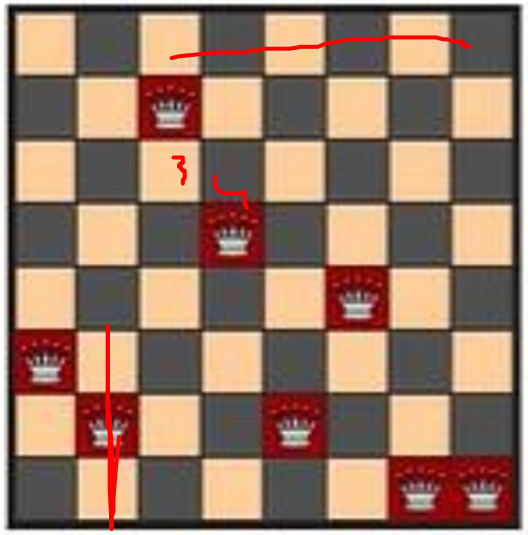
D



2	4	4	1	5	1	2	4

Step 3: Apply Fitness Function

Individual



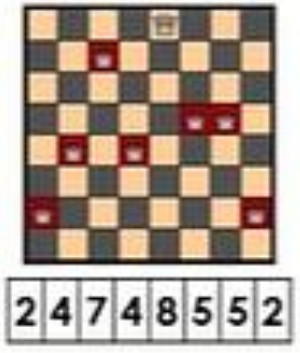
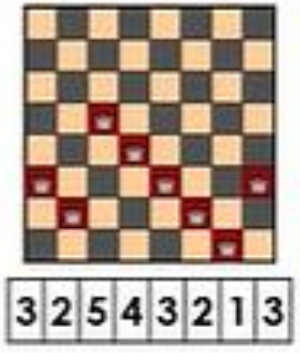
3	2	7	5	2	4	1	1

Fitness = No. of non attacking pairs

- Queen 1: 6
- Queen 2: 5
- Queen 3: 4
- Queen 4: 3
- Queen 5: 3
- Queen 6: 2
- Queen 7: 0
- Queen 8: 0

Total 23

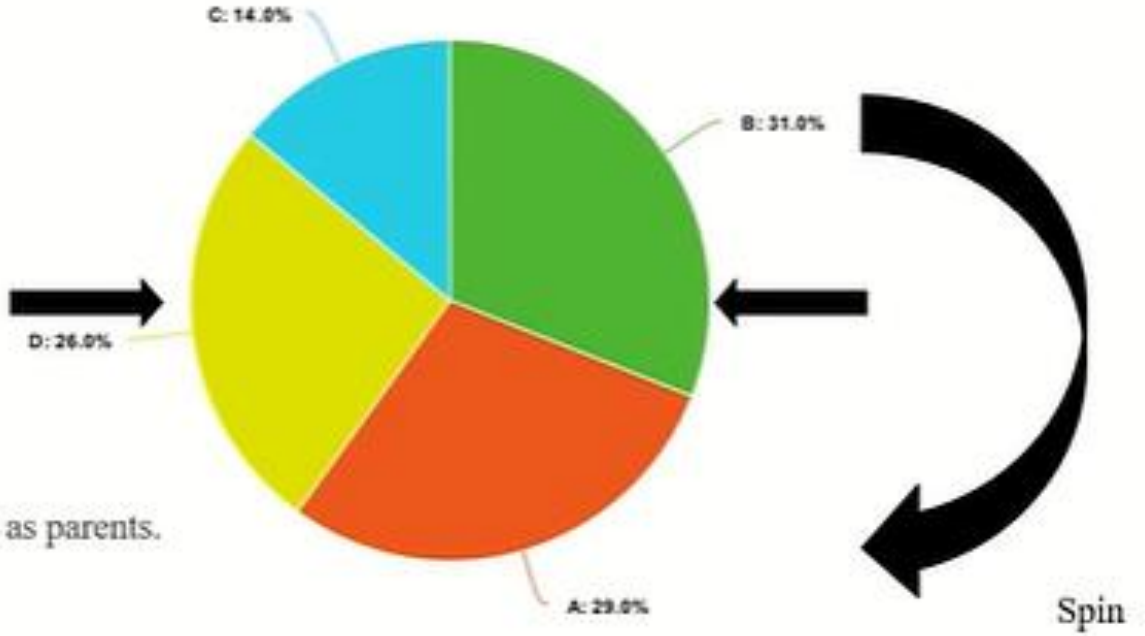
Step 3: Apply Fitness Function (contd.)

	A	B	C	D
Individuals	 3 2 7 5 2 4 1 1	 2 4 7 4 8 5 5 2	 3 2 5 4 3 2 1 3	 2 4 4 1 5 1 2 4
Fitness	23	24	11	20
Fitness %	29%	31%	14%	26%

Total:78

Step 4: Selection

Proportionate selection or roulette wheel selection gives individuals **with higher fitness values** a higher chance of being selected as parents, mimicking the concept of "survival of the fittest" in the natural evolutionary process.

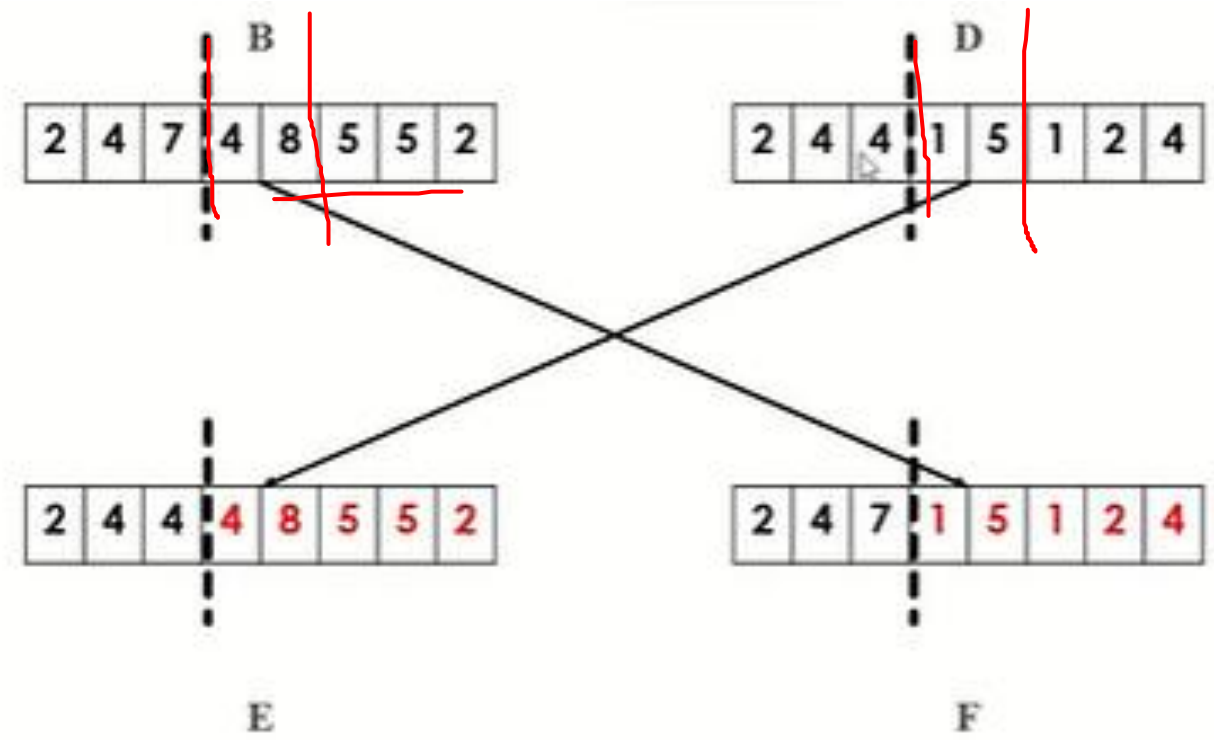


- There are various methods of selection.
- Roulette Wheel, Tournament, Rank, etc.
- Stochastic Universal Sampling (SUS).
- Population is divided on a wheel according to their respective percentages of fitness and two fixed points are placed.
- Wheel is spun and those individuals are selected at which the fixed points are pointing when the wheel stops.

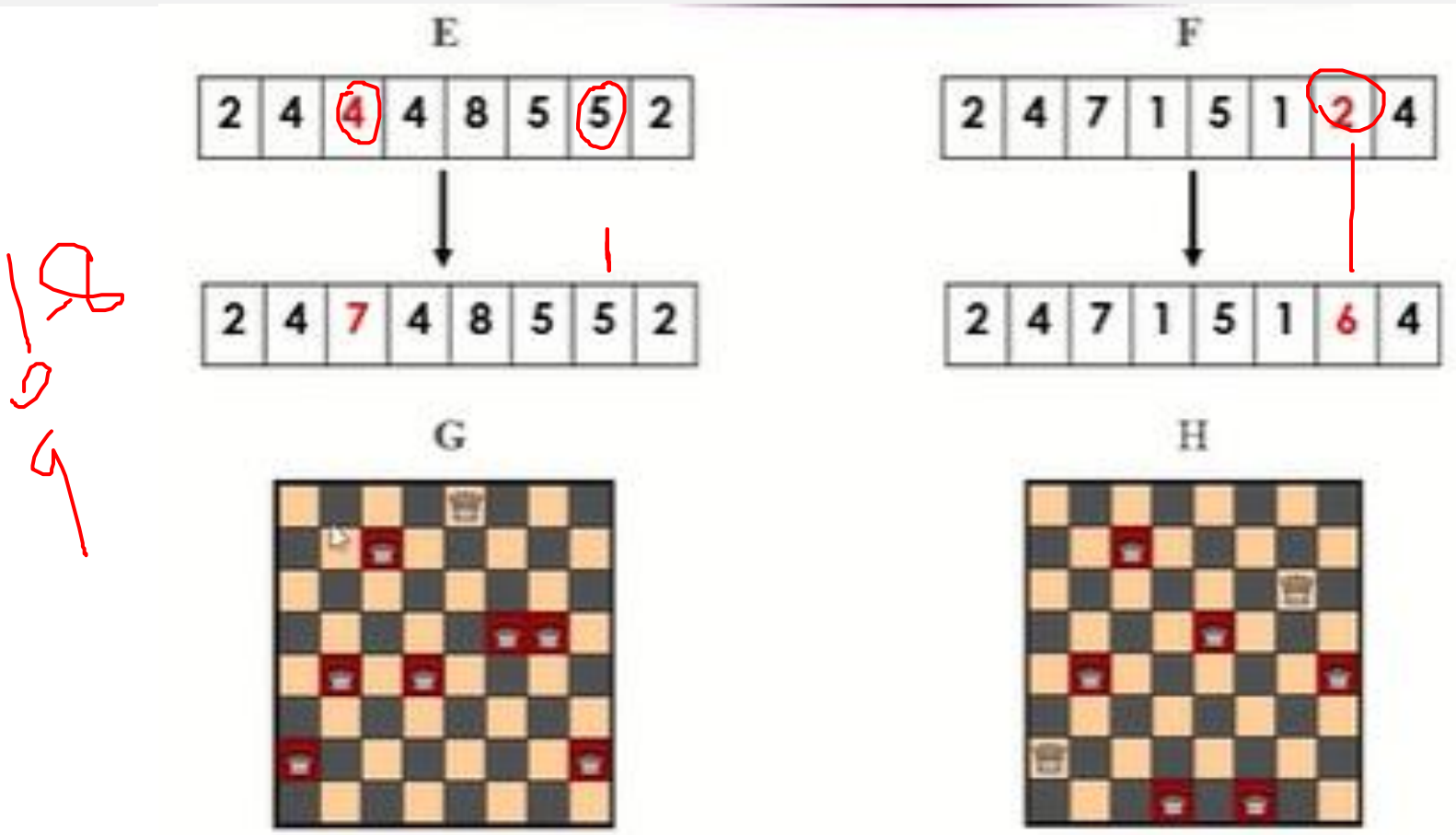
- B and D are selected as parents.

The key difference between SUS and traditional roulette wheel selection is that **SUS selects multiple parents in a single pass, rather than selecting one parent at a time.**

Step 5: Crossover



Step 6: Mutation



Step 7: Repeat

- All steps are repeated until best solution is reached.
- Best solution = Highest fitness score (28 in this case).