Metaheuristics
Evolutionary Algorithms

Evolutionary Algorithms

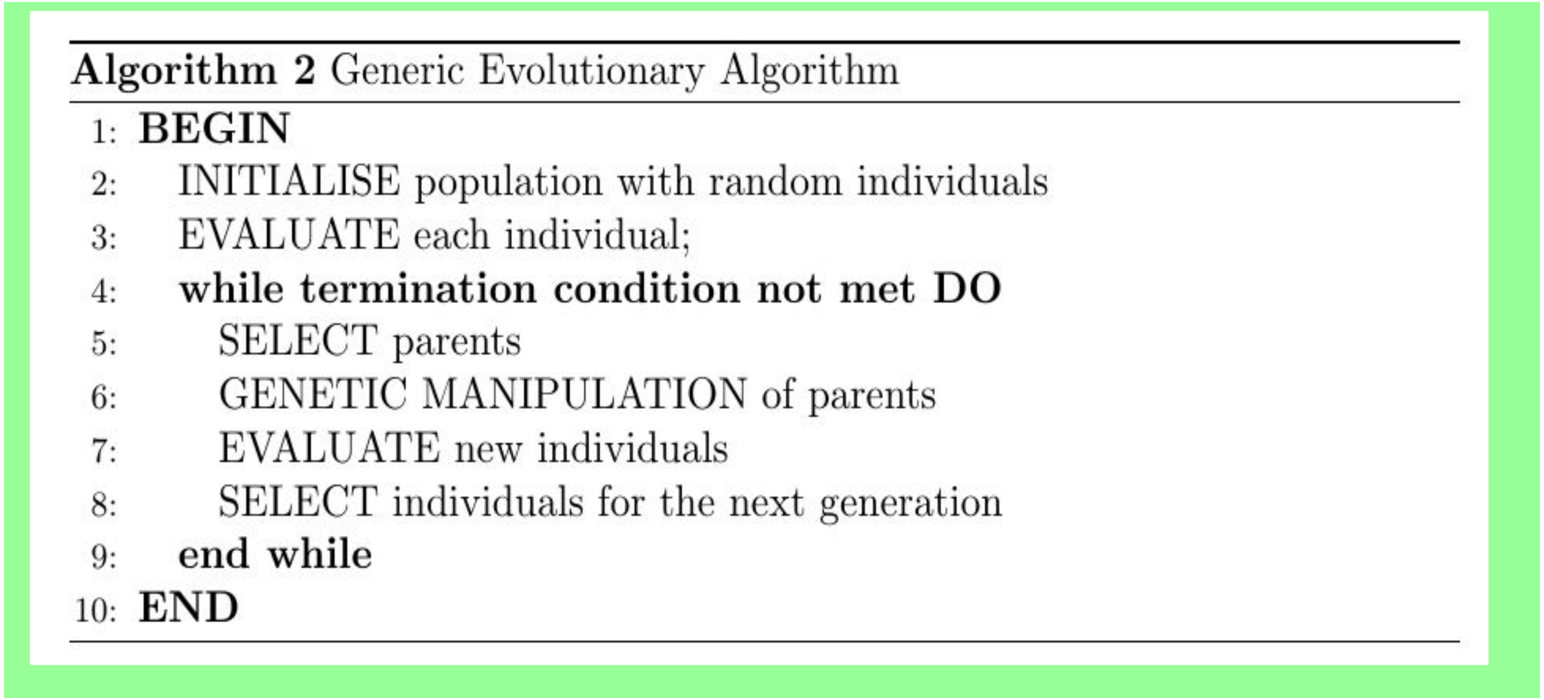
- Evolutionary Algorithms model Darwin’s principle of natural selection -> **survival of the fittest.**
- They operate on a collection of candidate solutions for some problem.
- Each candidate solution is represented as an individual and a collection of individuals is called a **population**.

Evolutionary Algorithms

- Each individual of the population is assigned a quality value known as a fitness which is evaluated by a function called a **fitness function**.

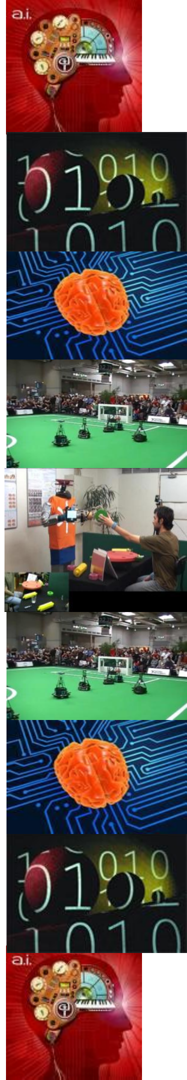
- Evolutionary algorithms are iterative and each iteration is called a generation.

Generic Evolutionary Algorithm



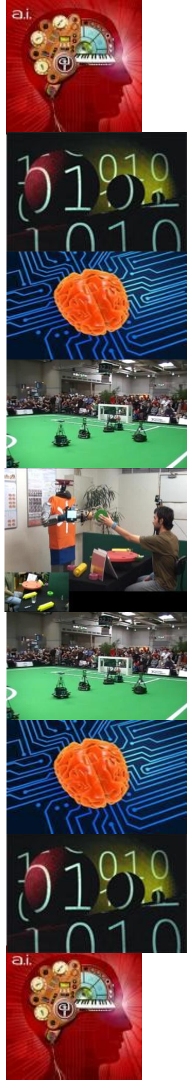
Algorithm 2 Generic Evolutionary Algorithm

```
1: BEGIN
2:    INITIALISE population with random individuals
3:    EVALUATE each individual;
4:    while termination condition not met DO
5:       SELECT parents
6:       GENETIC MANIPULATION of parents
7:       EVALUATE new individuals
8:       SELECT individuals for the next generation
9:    end while
10: END
```



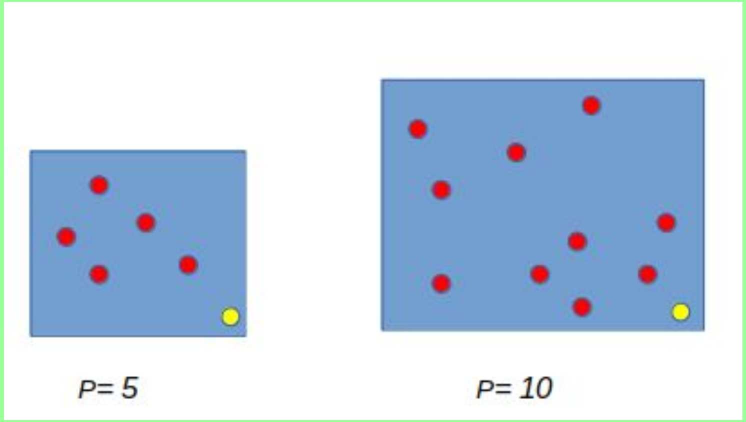
Key Stages in Generic EA

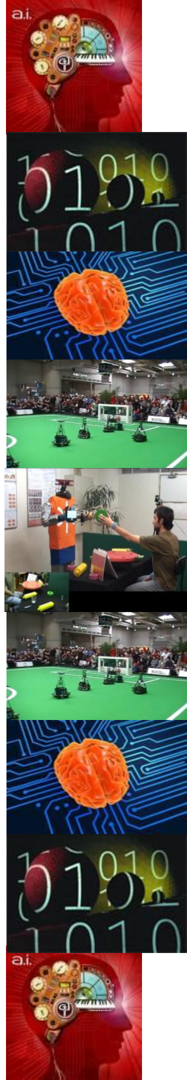
1. Initial population generation. (usually random)
2. Evaluation of every individual in the population.
3. Selection of parents
4. Application of genetic operators.
5. Evaluation of every individual in the population.
6. Population update (generational / steady state)
7. Check for the stopping criteria.



Population Initialisation

- Random - why ?.
- This defines the search space.
- There is a population size parameter that needs to be specified.





Individual Evaluation

- A **fitness function** is used to evaluate the suitability of an individual.
- It measures how good the individual is at solving the problem i.e meeting the objective.
- The function must provide a consistent measure of an attribute that guides the search to the solution.

Selection of Parents

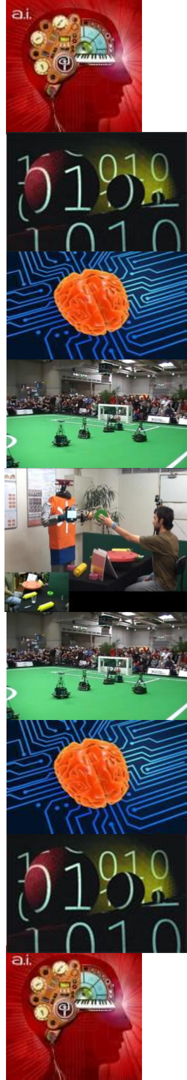
- Selection operators mimic natural selection and are used to choose parents which the genetic (reproduction) operators are applied to.
- These methods are biased towards the fitter individuals of the population.

Genetic Operators

- Genetic operators - Genetic operators are used to create the offspring of each generation.
- The simplest operator is reproduction which makes a copy of the parent.
- The crossover operator swaps components in copies of two parents to create two offspring.

Genetic Operators

- The mutation operator creates an offspring by making random changes to a copy of the chosen parent.



Stopping Criteria

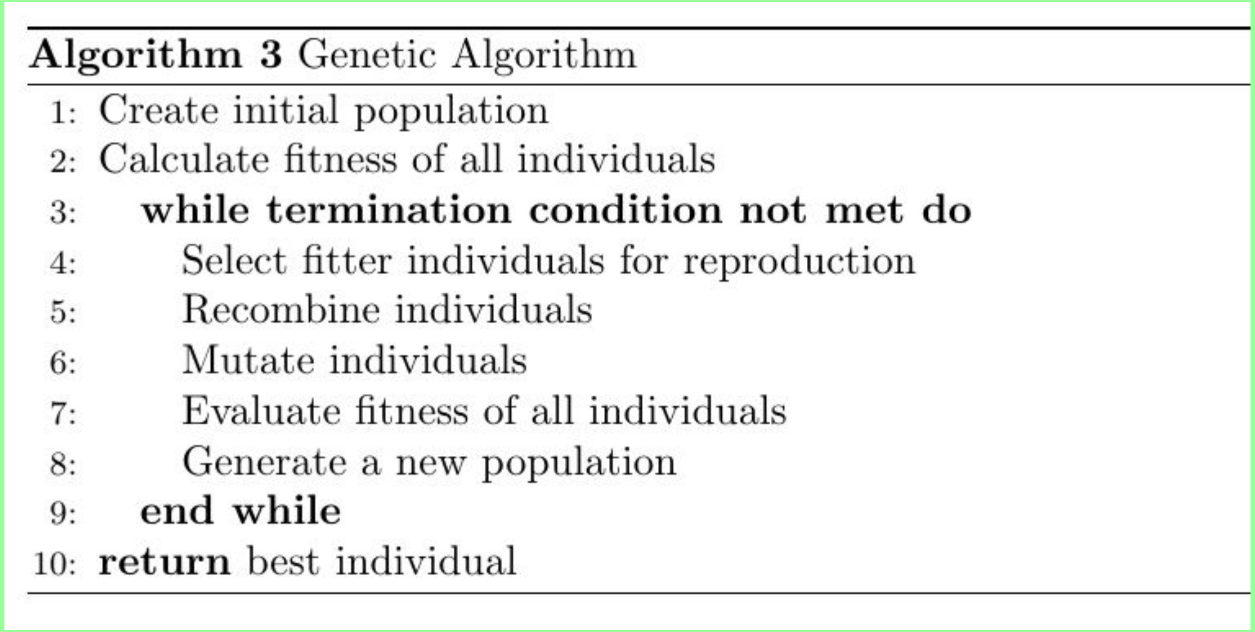
- Solution or near optimal solution found.
- No improvement - convergence.
- Number of generations achieved.

Genetic Algorithm

- Genetic algorithms were introduced by John Holland of Michigan University.

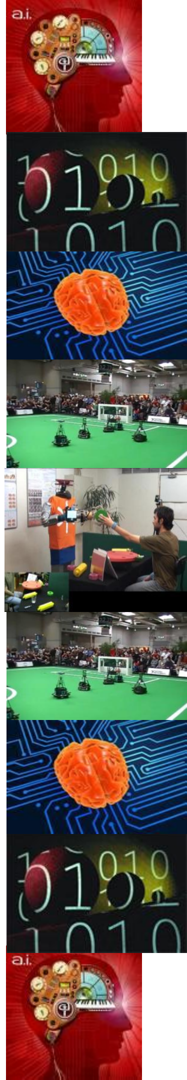
- Initially, each element of the population, i.e. each chromosome was represented as a fixed length binary string, e.g. 1001100.
- Other representations have been proposed

Genetic Algorithm

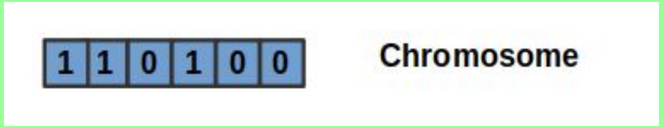


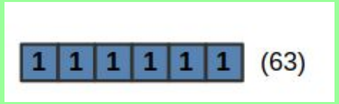
Algorithm 3 Genetic Algorithm

```
 1: Create initial population
 2: Calculate fitness of all individuals
 3:     while termination condition not met do
 4:         Select fitter individuals for reproduction
 5:         Recombine individuals
 6:         Mutate individuals
 7:         Evaluate fitness of all individuals
 8:         Generate a new population
 9:     end while
10: return best individual
```

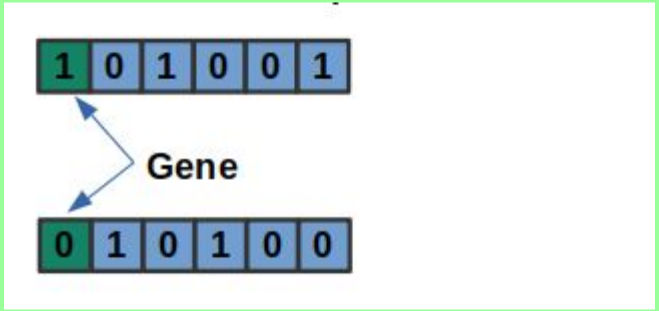


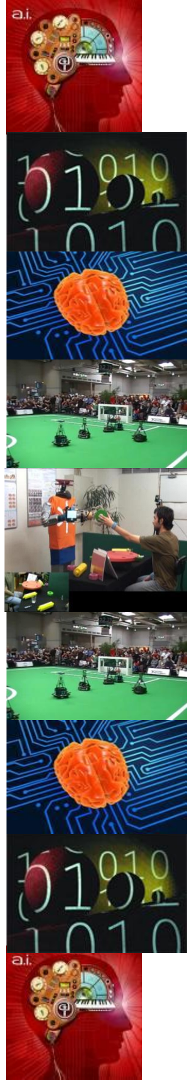
GA- Components



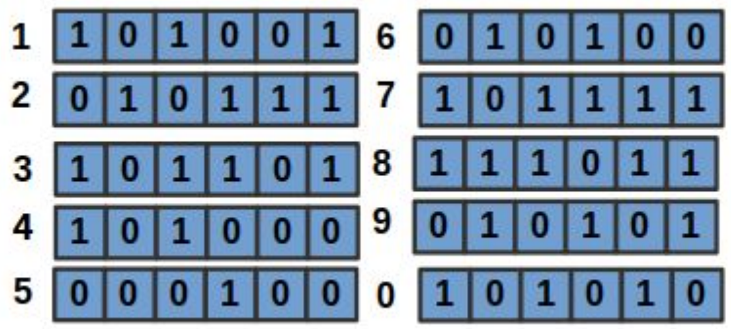


Target





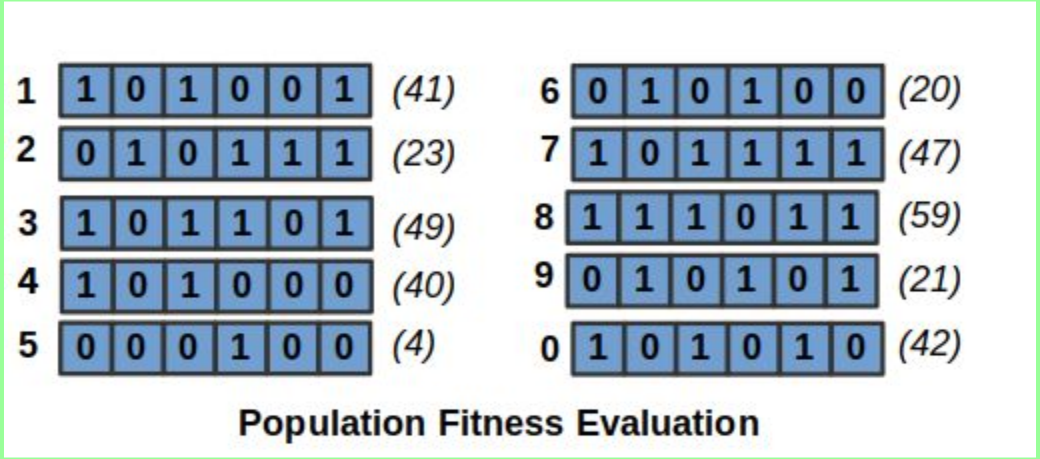
Initial Population Generation



1	1	0	1	0	0	1	6	0	1	0	1	0	0
2	0	1	0	1	1	1	7	1	0	1	1	1	1
3	1	0	1	1	0	1	8	1	1	1	0	1	1
4	1	0	1	0	0	0	9	0	1	0	1	0	1
5	0	0	0	1	0	0	0	1	0	1	0	1	0

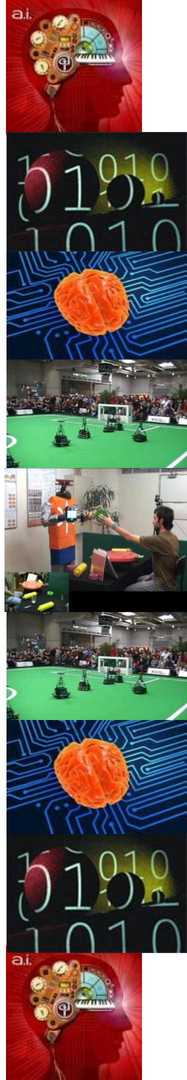
Population

Fitness Evaluation



1	1	0	1	0	0	1	*(41)*
2	0	1	0	1	1	1	*(23)*
3	1	0	1	1	0	1	*(49)*
4	1	0	1	0	0	0	*(40)*
5	0	0	0	1	0	0	*(4)*
6	0	1	0	1	0	0	*(20)*
7	1	0	1	1	1	1	*(47)*
8	1	1	1	0	1	1	*(59)*
9	0	1	0	1	0	1	*(21)*
0	1	0	1	0	1	0	*(42)*

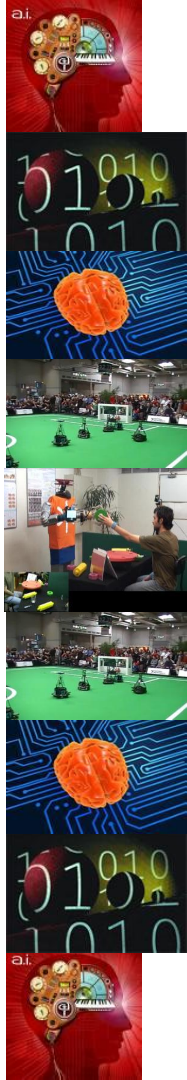
Population Fitness Evaluation



Selection

Use **Tournament Selection** of size **4**

i.e randomly select 4 individuals from the population and select the fittest to be **Parent 1** repeat the process for **Parent 2**



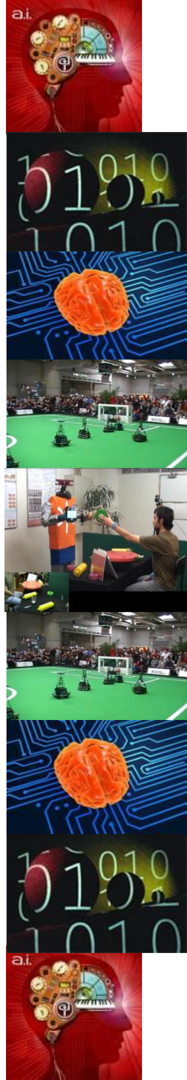
Selection

- Tournament Selection size is problem dependent.

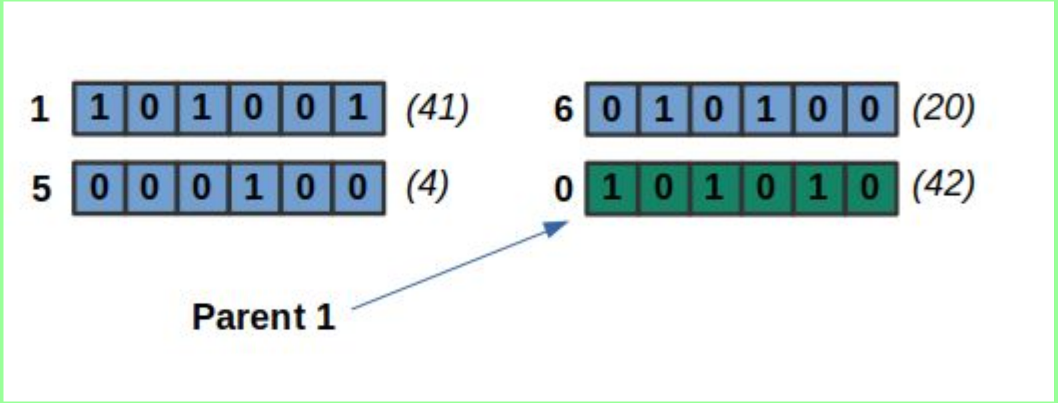
- What is the effect a size of

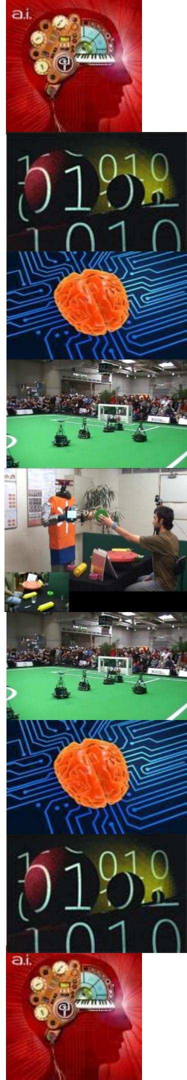
  i) 1   ?

  ii) equal to population size ?

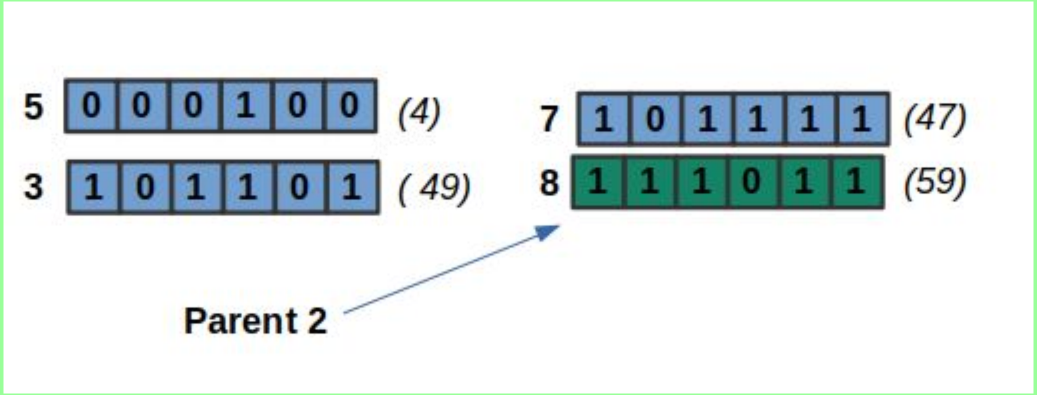


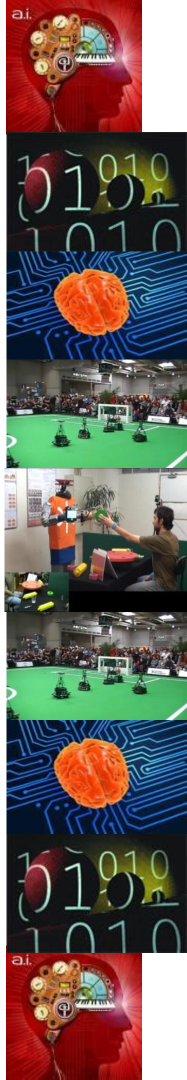
Selection Parent 1



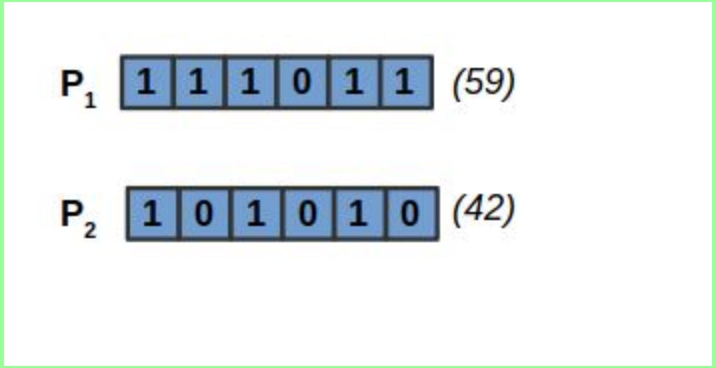


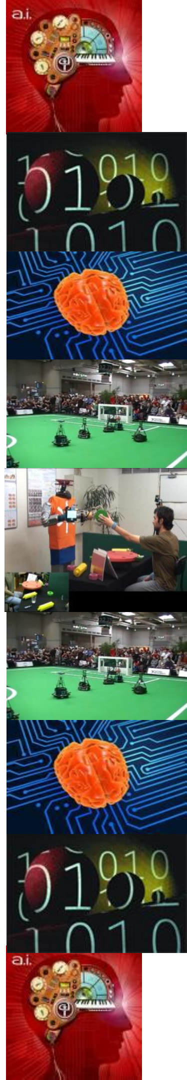
Selection of Parent 2





Selected Parents





Recombine(crossover)

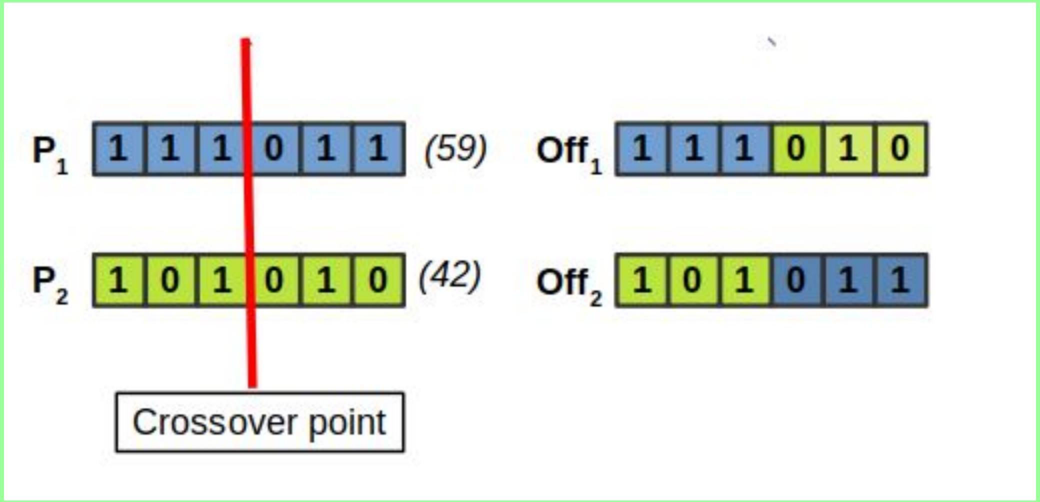
- If crossover probability allows.
- Randomly select a crossover point between 0 - 5.

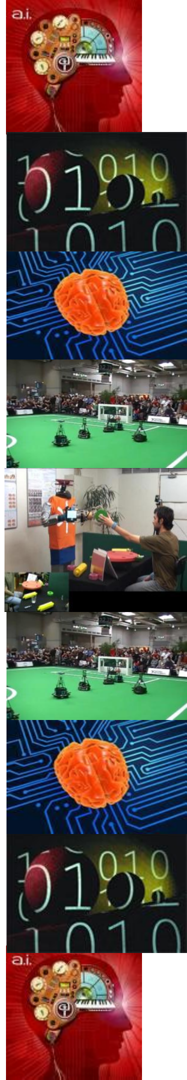
One-point crossover.

Two-point crossover.

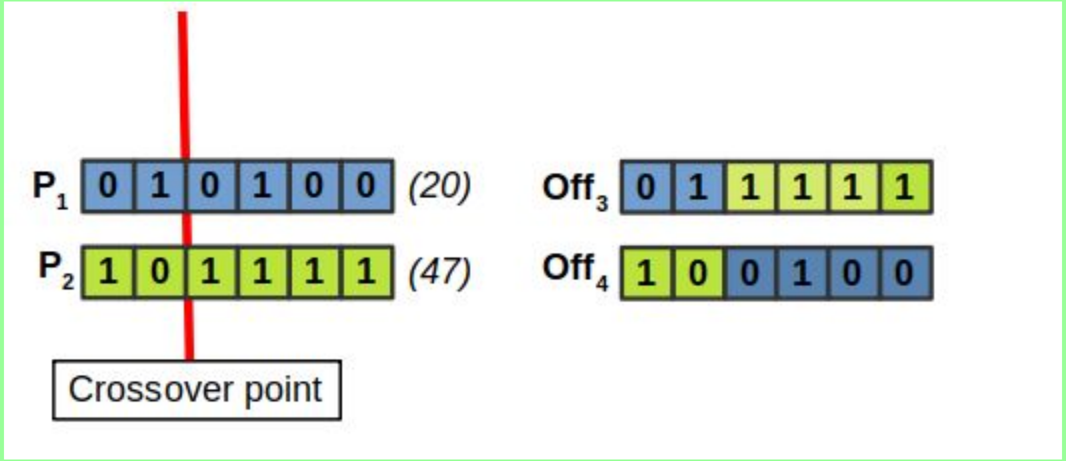
Uniform crossover.

Crossover


P_1
1 1 1 0 1 1
(59)
P_2
1 0 1 0 1 0
(42)
Off_1
1 1 1 0 1 0
Off_2
1 0 1 0 1 1
Crossover point

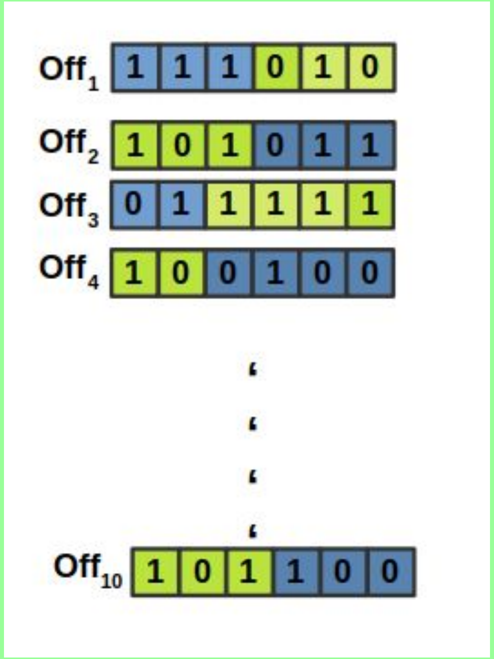


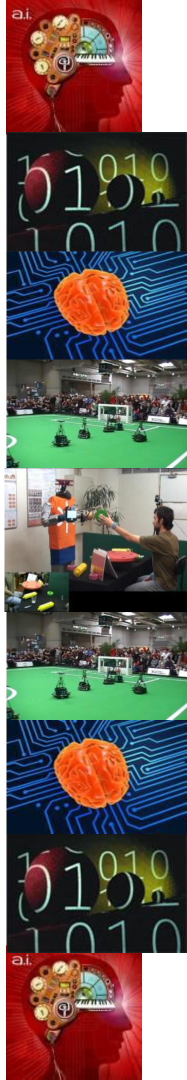
Crossover


P1 0 1 0 1 0 0 (20)
P2 1 0 1 1 1 1 (47)
Off3 0 1 1 1 1 1
Off4 1 0 0 1 0 0
Crossover point

Crossover

Crossover until the population has been replaced by new individuals



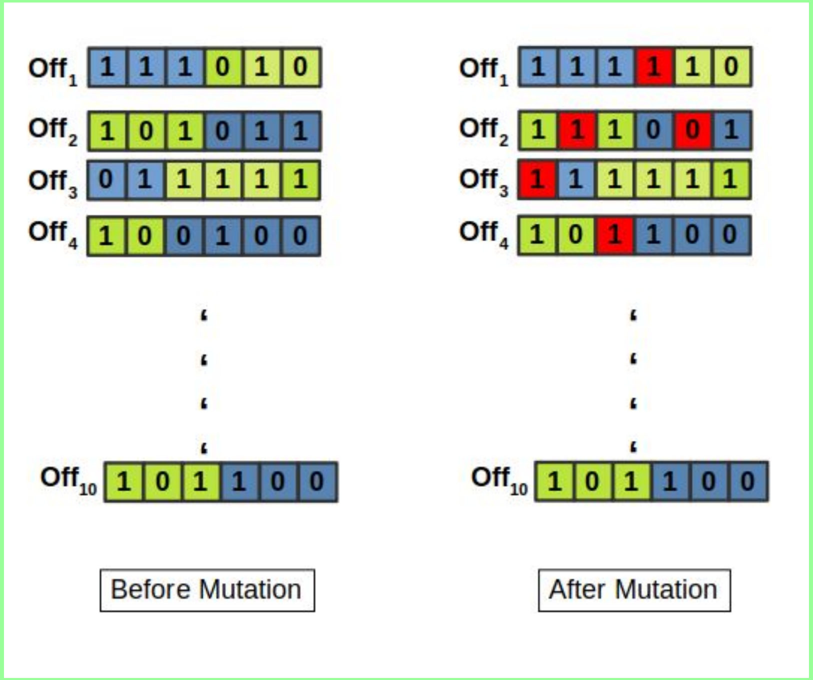


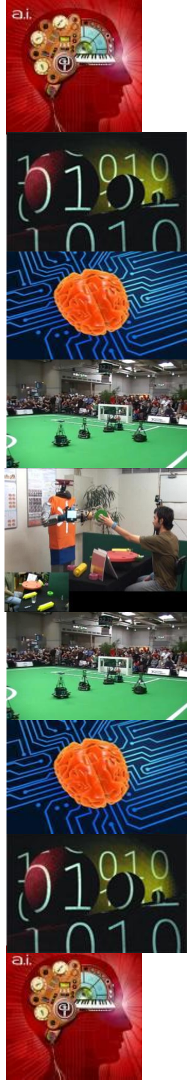
Mutation

- To each individual of the new population we apply the mutation operator.

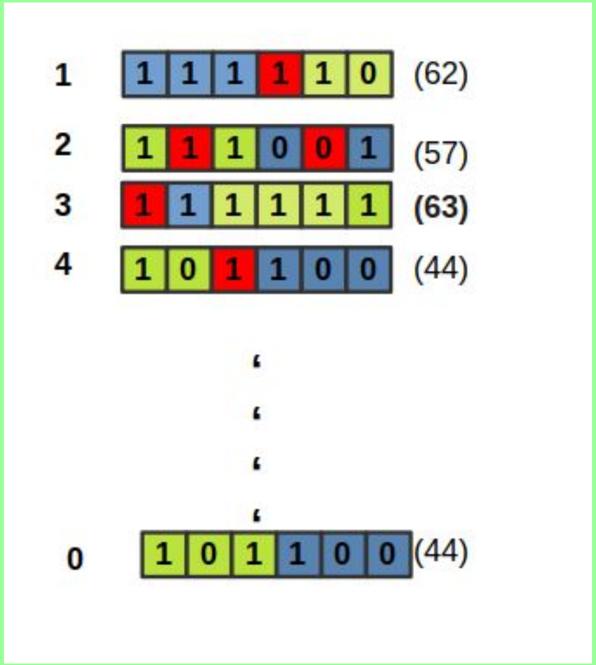
- We consider each gene (bit) of the individual for mutation based on the mutation probability rate.

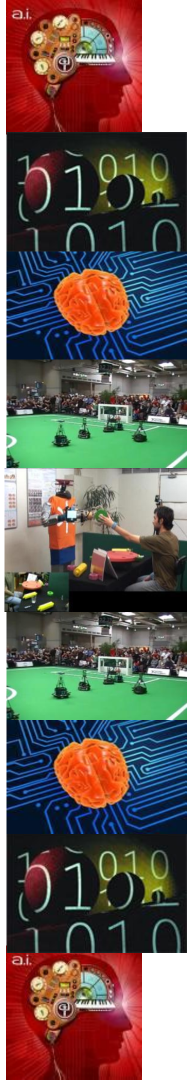
Bit Flip Mutation


Off_1 1 1 1 0 1 0
Off_2 1 0 1 0 1 1
Off_3 0 1 1 1 1 1
Off_4 1 0 0 1 0 0
Off_{10} 1 0 1 1 0 0
Before Mutation
Off_1 1 1 1 1 1 0
Off_2 1 1 1 0 0 1
Off_3 1 1 1 1 1 1
Off_4 1 0 1 1 0 0
Off_{10} 1 0 1 1 0 0
After Mutation

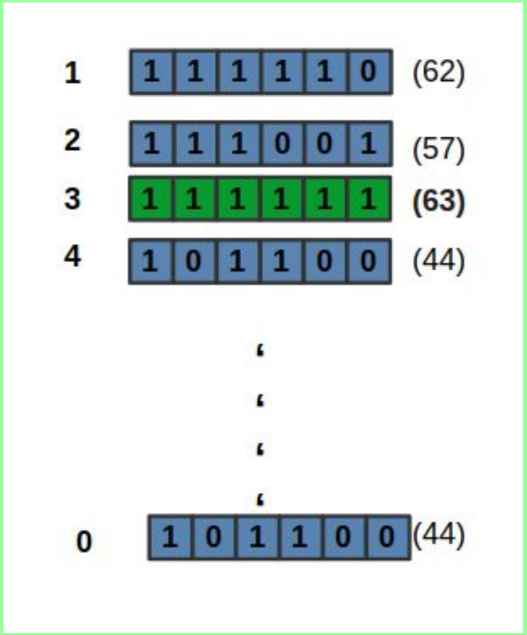


Evaluate the Fitness





Update The Population



Repeat

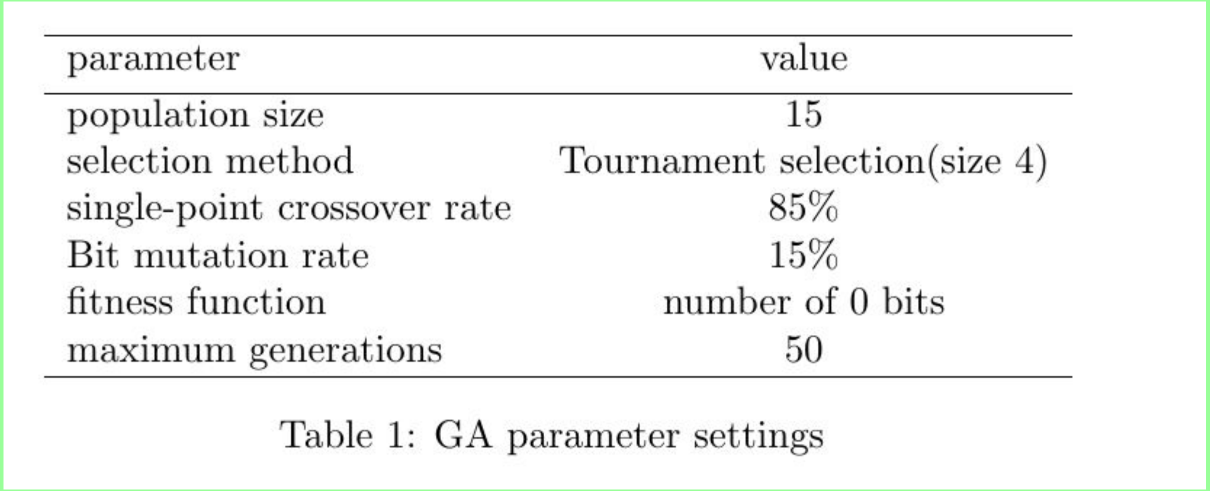
- Check if the stopping criteria has been met.
- Stopping criteria is problem dependant.
- Stopping criteria - no more improvement or number of generations have been achieved

GA Parameters

- Population Size.
- Selection type
- Crossover type/rate.
- Mutation type/rate.
- Stopping criteria
  - Number of Generations.
  - Number of runs.
  - Convergence

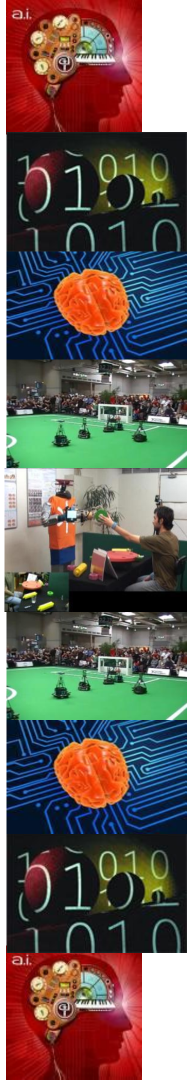
Example - GA Parameters

This is how we report on GA parameters



parameter	value
population size	15
selection method	Tournament selection(size 4)
single-point crossover rate	85%
Bit mutation rate	15%
fitness function	number of 0 bits
maximum generations	50

Table 1: GA parameter settings



- Which part of the GA are low level heuristics ?

- Which heuristic performs exploration ?

- Which heuristic performs exploitation ?

Questions ????