Artificial Intelligence
Lec 8

Ass. Prof. Taher Hamza
Dr. Sara Elmetwally

Contents

Local search algorithms

- In many optimization problems, the path to the goal is irrelevant; the goal state itself is the solution, **EX** : N-queen problem
- In such cases, we can use local search algorithms
- keep a single "current" state, try to improve it
- All states have an objective function
- Goal is to find state with the best objective value

Local search algorithms

1. Hill climbing local search

- loop that continuously moves in the direction of increasing value (Aka as greedy local search)
- Terminates when a peak is reached
- Hill climbing does not look ahead of the immediate neighbors of the current state "trying to find the top of Mount Everest while in a thick fog"

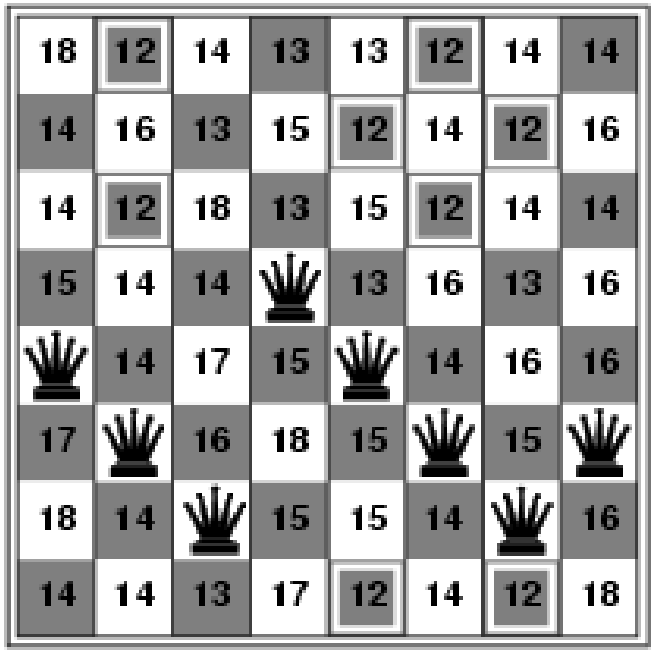
Local search algorithms

1. Hill climbing local search

```
function HILL-CLIMBING(problem) returns a state that is a local maximum
    inputs: problem, a problem
    local variables: current, a node
                     neighbor, a node

    current ← MAKE-NODE(INITIAL-STATE[problem])
    loop do
        neighbor ← a highest-valued successor of current
        if VALUE[neighbor] ≤ VALUE[current] then return STATE[current]
        current ← neighbor
```

Local search algorithms

1. 8-queen problem , example

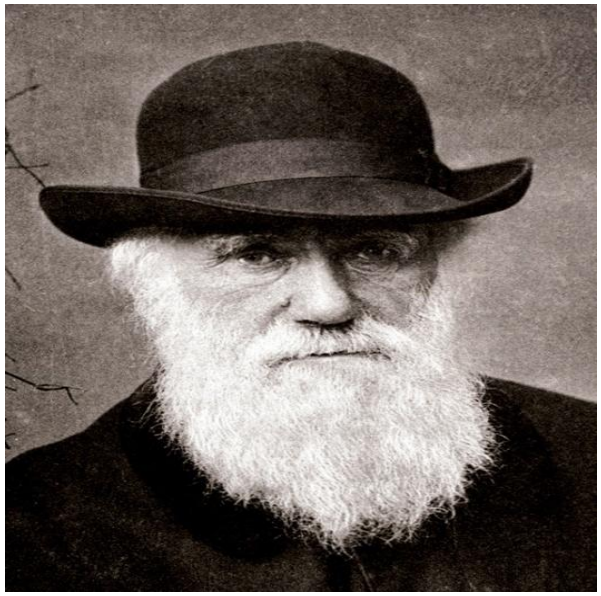


- Objective function = 17

  (**number of pairs of queens that are attacking each other**)

- Hill climbing search for the move which minimize this value

Genetic Algorithm (GA)

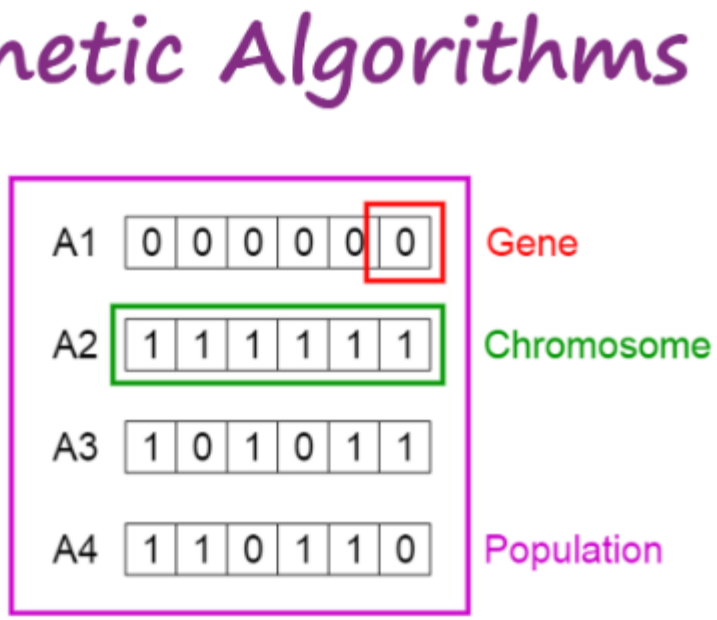


- Computations adopt Darwin principles 'survival of the fittest'
- Useful when search space very large or too complex
- Suitable for optimization problems

Genetic Algorithm (GA)

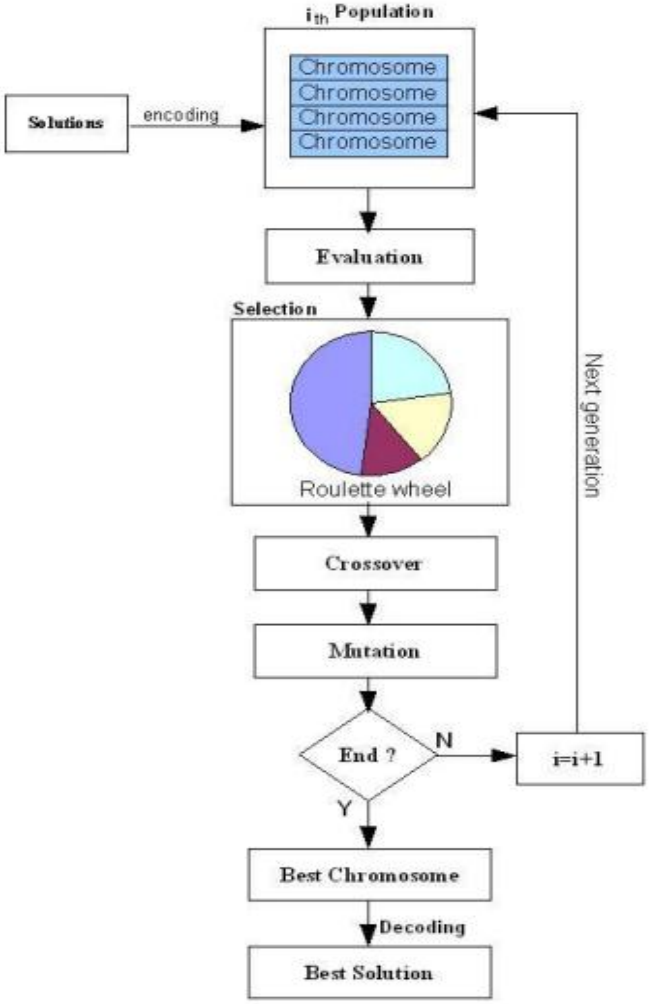
- GA is one of the evolutionally computation algorithms
- Start with k randomly generated states (population)
- A state (chromosome) is represented as a string of numbers (Genes)
- Each state is possible solution for the problem
- Chromosomes are evaluated by fitness function
- Selection of chromosomes is based on fitness function

Genetic Algorithm (GA)

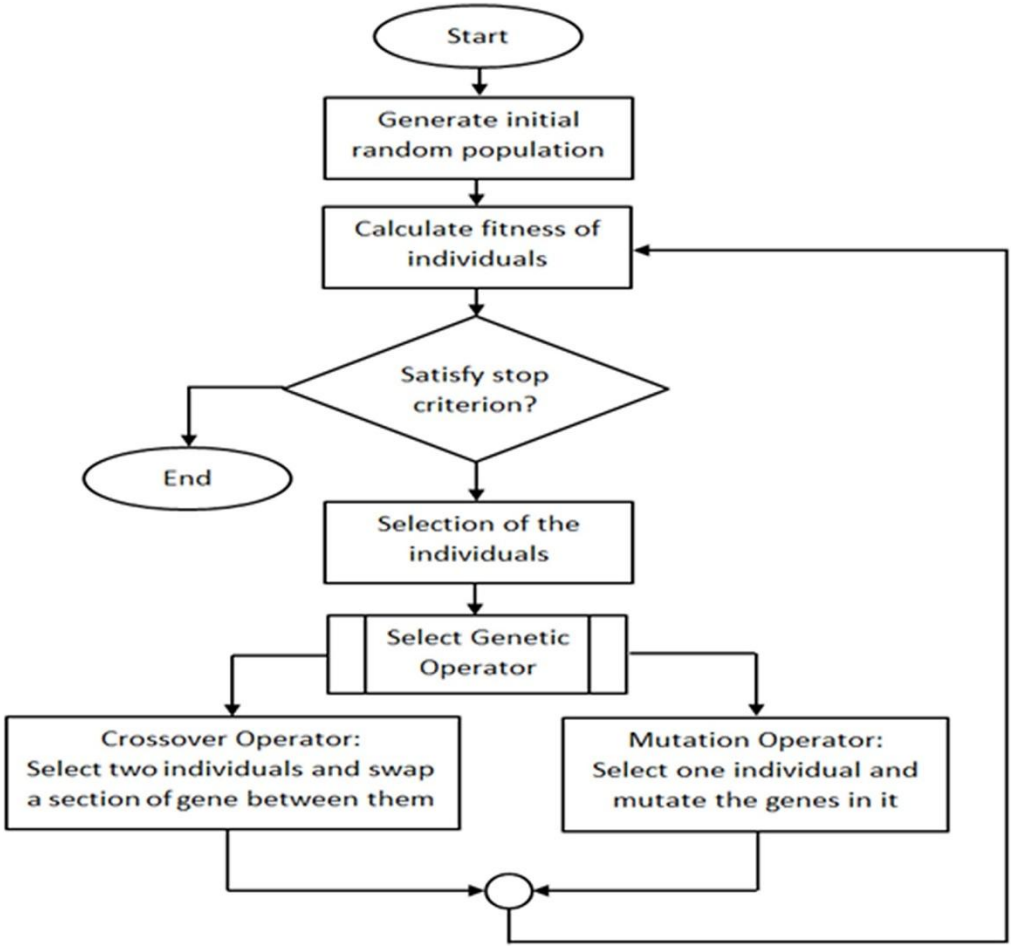


https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3

Genetic Algorithm (GA)



GA- Flow chart

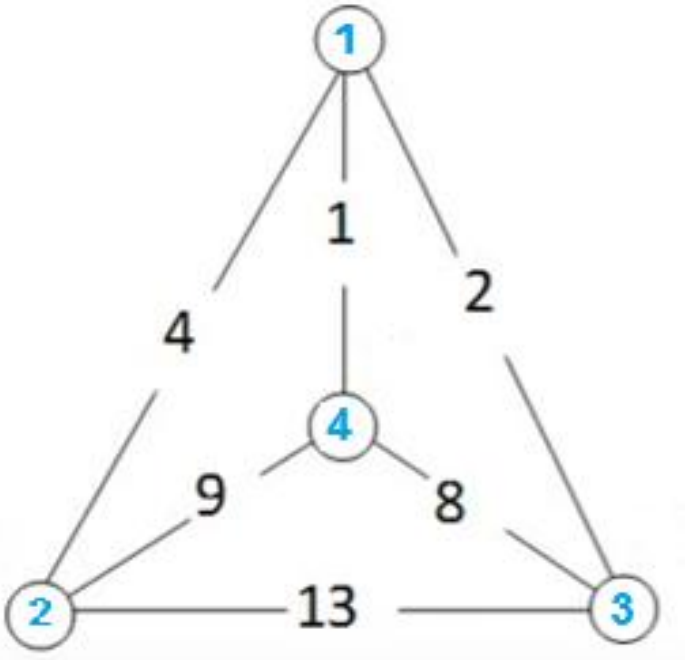


https://plos.figshare.com/articles/_Flow_Chart_of_Genetic_Algorithm_with_all_steps_involved_from_beginning_until_termination_conditions_met_6_/1418786

GA , example

Traveling Salesman Problem (TSP):

- Used to Control routing system
- Can be in one city at a time
- Each city visited once and only once
- Our objective is to determine the shortest route through N cities (nodes)



GA , example

Traveling Salesman Problem (TSP):

- Testing every possibility would require N! separate additions

- For a 15 city tour:

  - 15! = 1.31 x 10^{12} separate calculations

  - Assuming 1 million calculations per second → 15 days

(complexity!)

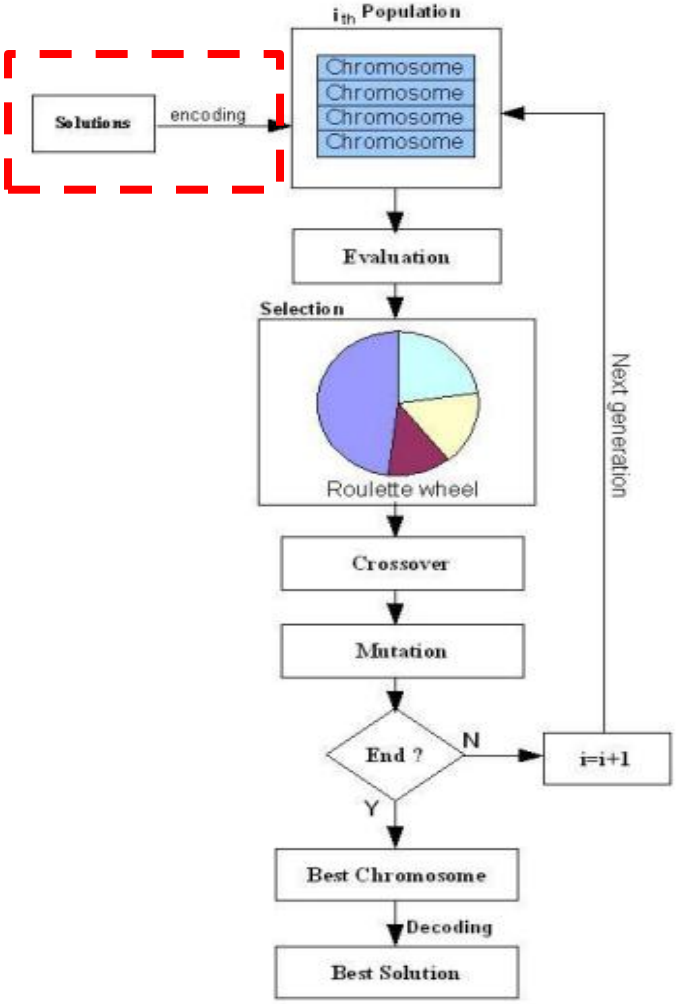
GA , example

TSP in real life:

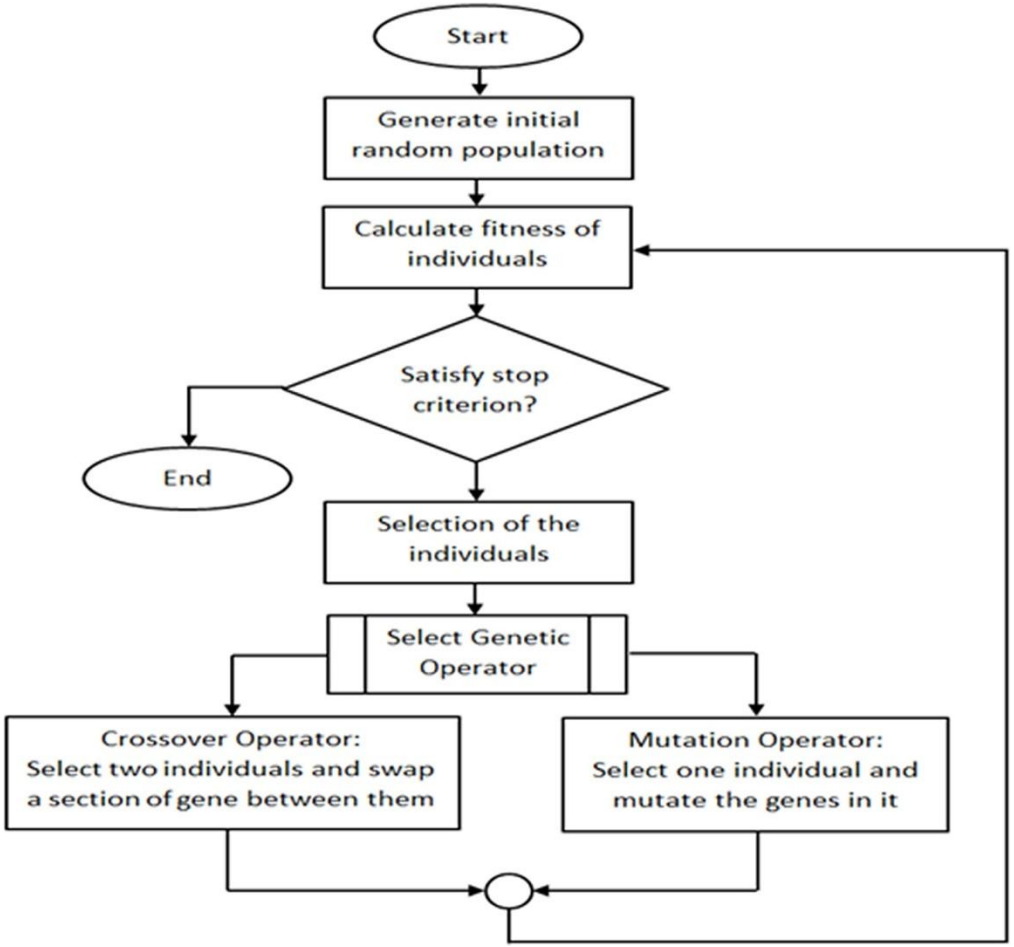
GA , example

TSP in real life:

Genetic Algorithm (GA)

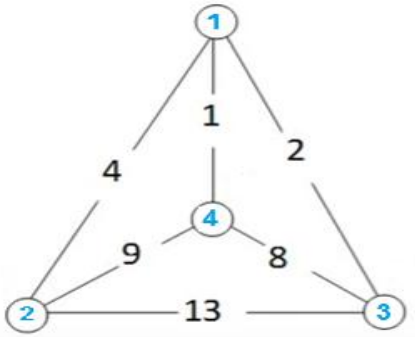


GA- Flow chart



https://plos.figshare.com/articles/_Flow_Chart_of_Genetic_Algorithm_with_all_steps_involved_from_beginning_until_termination_conditions_met_6_/1418786

GA, example



Solution Encoding:

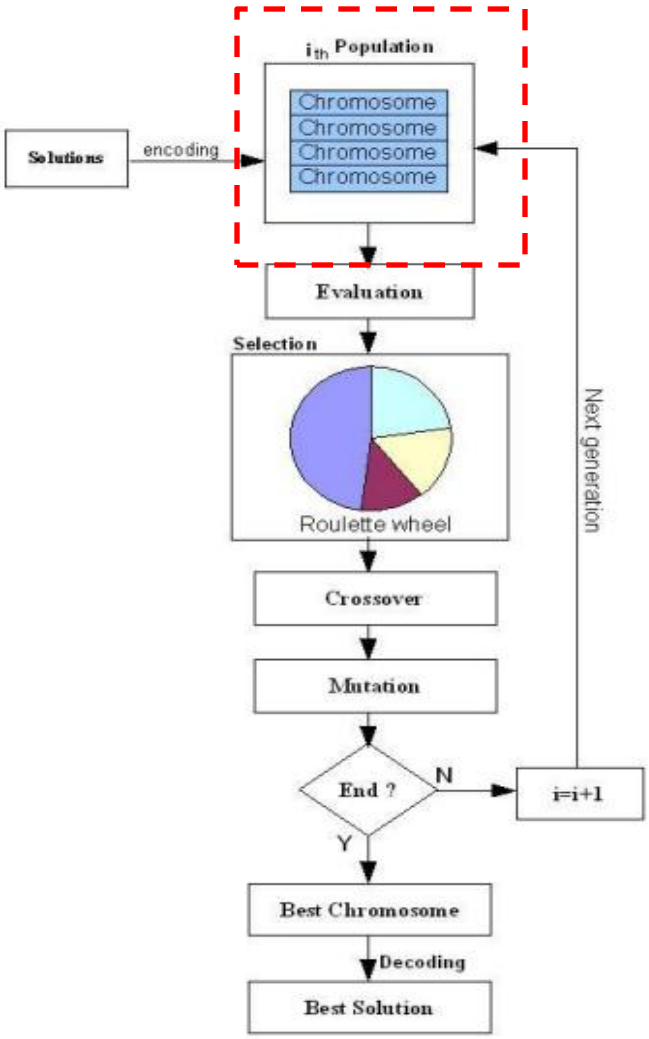
- Each solution (chromosome) will be represented by 4-genes (number of cities in the tour)

EX: "**1432**" means that: (Tour form node 1 to 4 to 3 to 2)

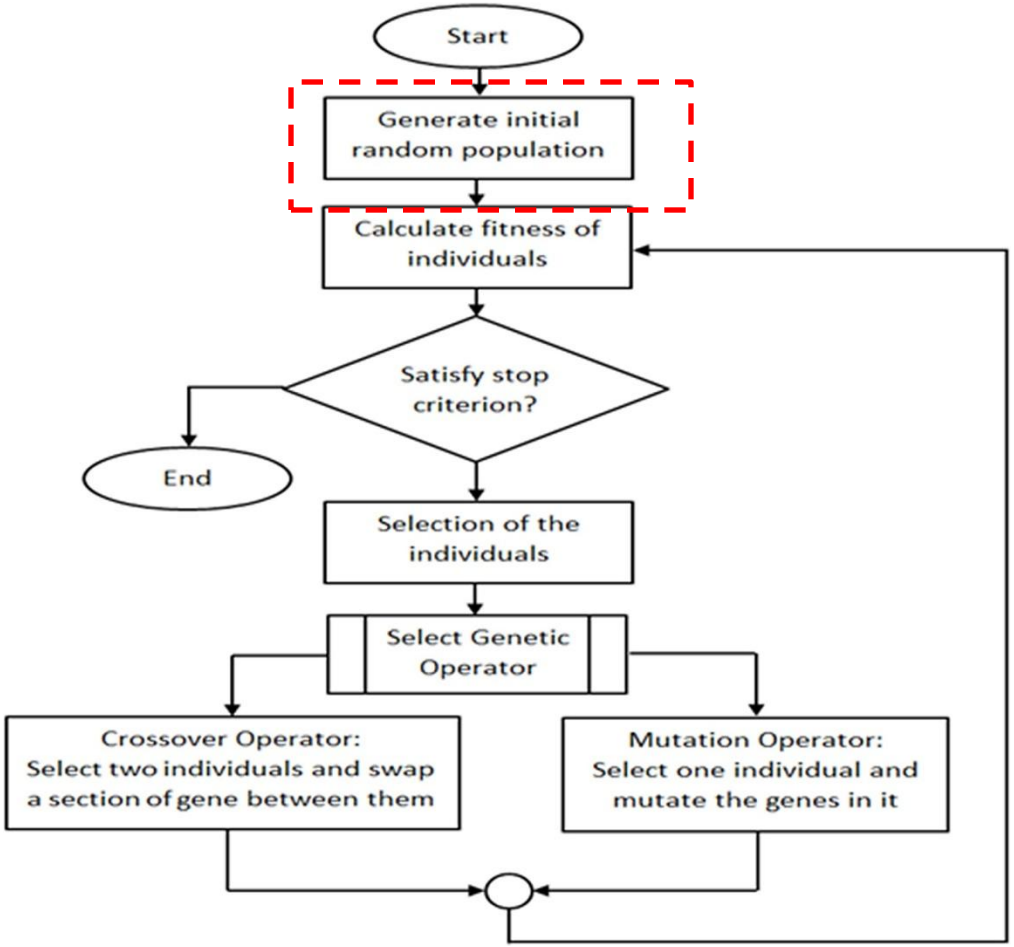
- This tour yield a total cost of 22

- Our job is to search for the tour with minimum cost

Genetic Algorithm (GA)



GA- Flow chart



https://plos.figshare.com/articles/_Flow_Chart_of_Genetic_Algorithm_with_all_steps_involved_from_beginning_until_termination_conditions_met_6_/1418786

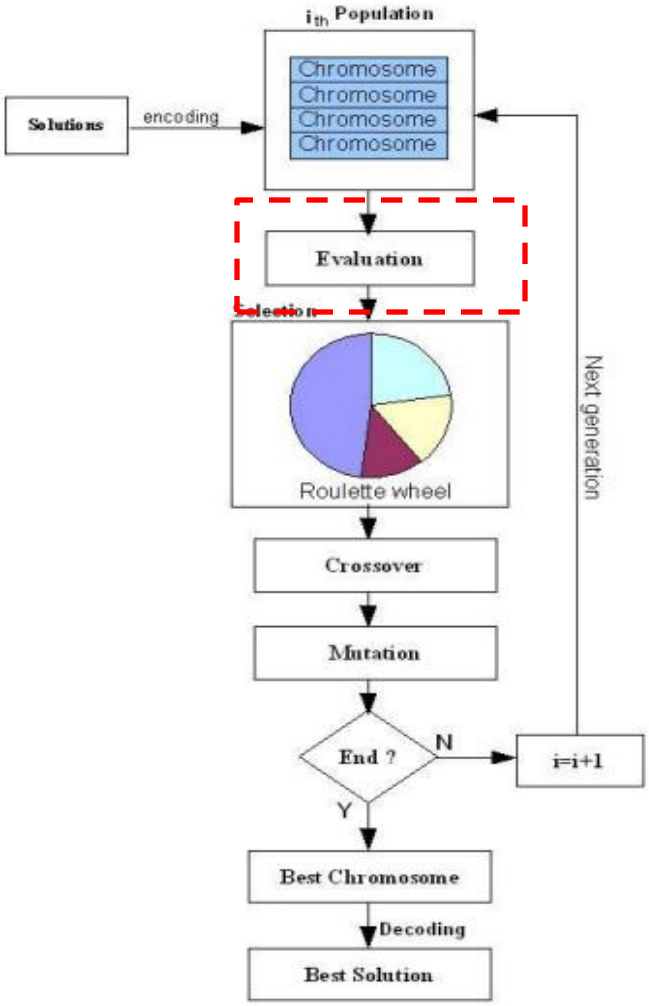
GA , example

Population initialization:

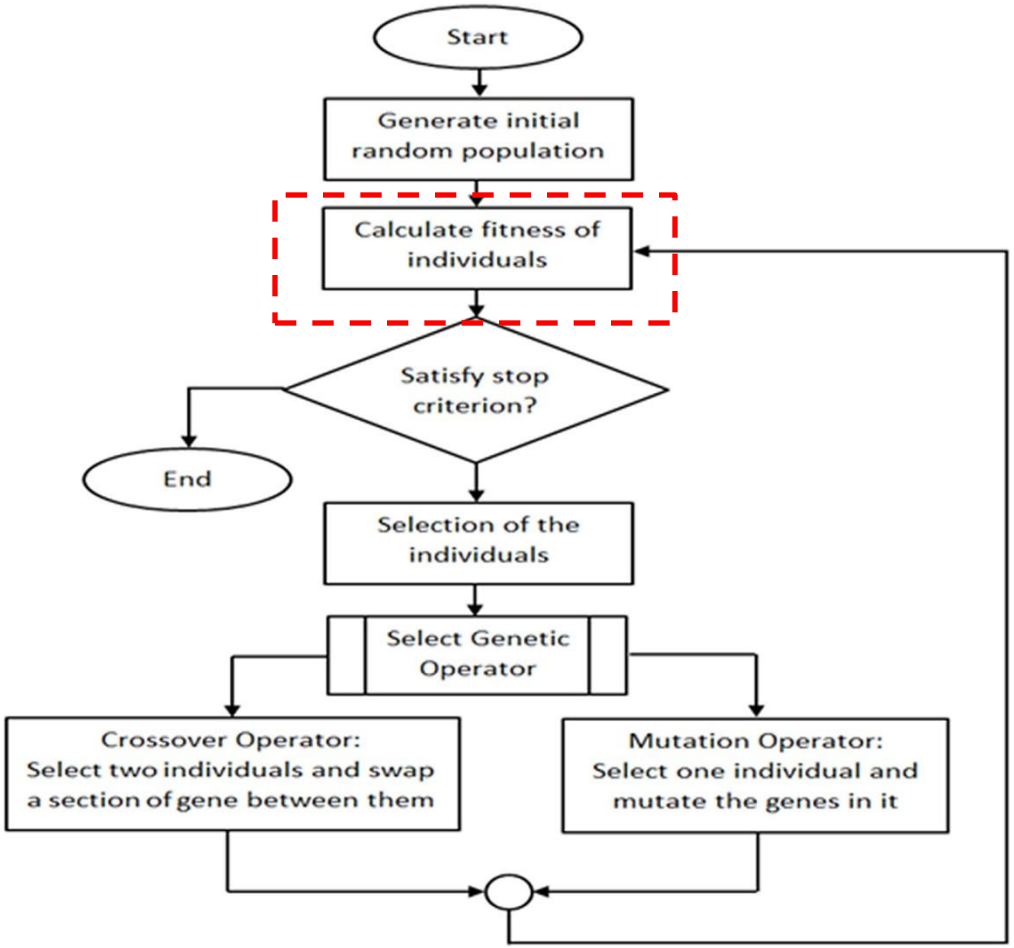
- Suppose we have population size **4**
- Construct GA initial population randomly
- Each chromosome (solution) is consist of **4** genes.

		Interpretation:
Chromosome _1	**1432**	1→4→3→2
Chromosome _2	**3241**	3→2→4→1
Chromosome _3	**4123**	4→1→2→3
Chromosome _4	**1324**	1→3→2→4

Genetic Algorithm (GA)

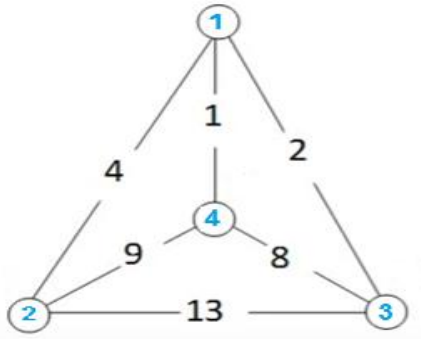


GA- Flow chart



https://plos.figshare.com/articles/_Flow_Chart_of_Genetic_Algorithm_with_all_steps_involved_from_beginning_until_termination_conditions_met_6_/1418786

GA , example

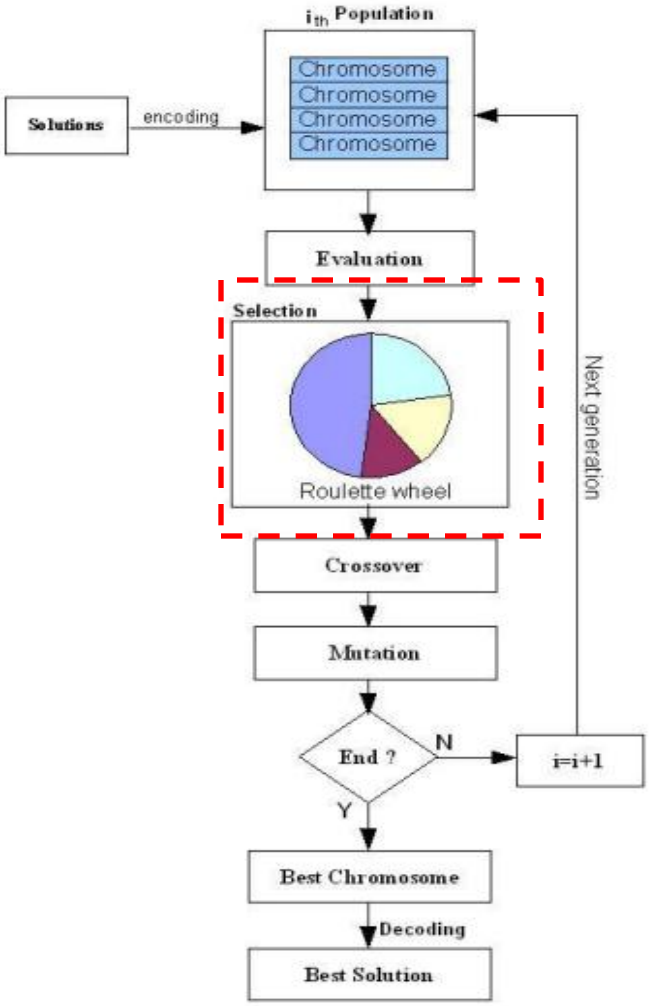


Population Evolution (fitness function):

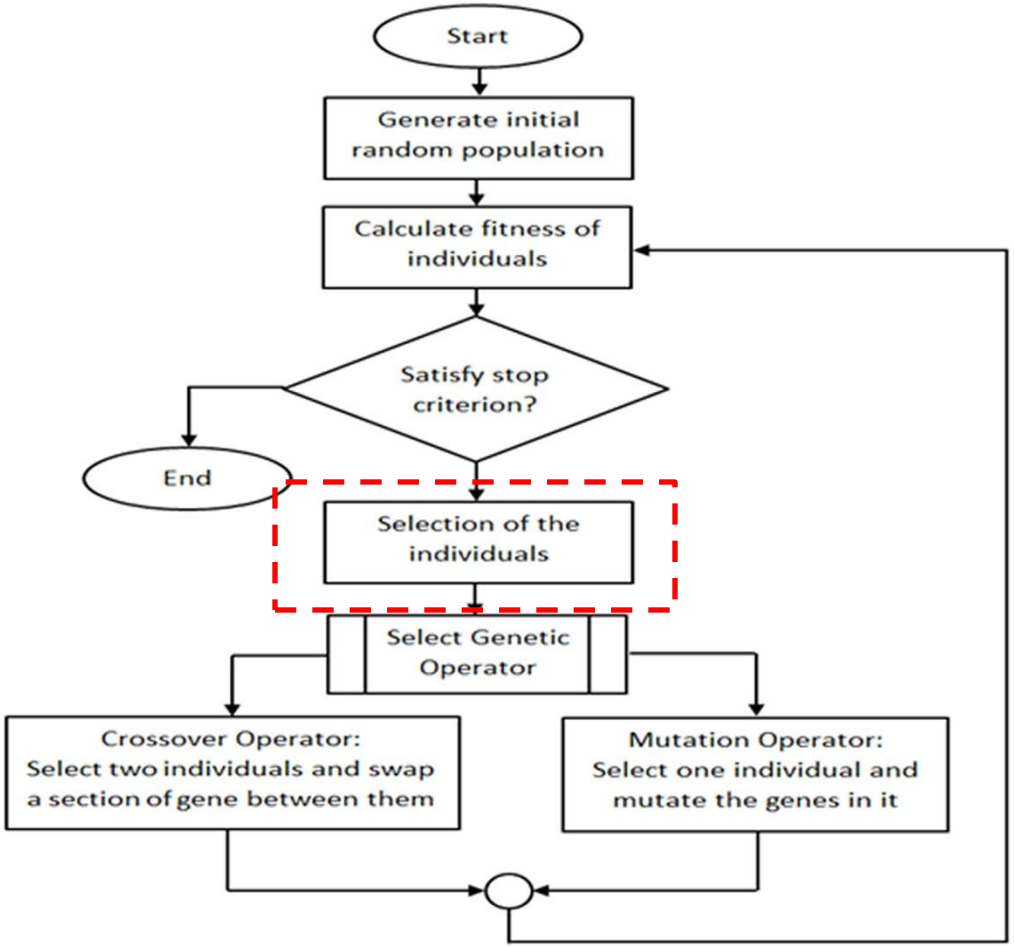
Determine the fitness value for each solution, and the **total fitness**, $F = \Sigma f_i$.

i	Initial pop.	F_i
1	**1432**	**22**
2	**3241**	**23**
3	**4123**	**18**
4	**1324**	**24**

Genetic Algorithm (GA)



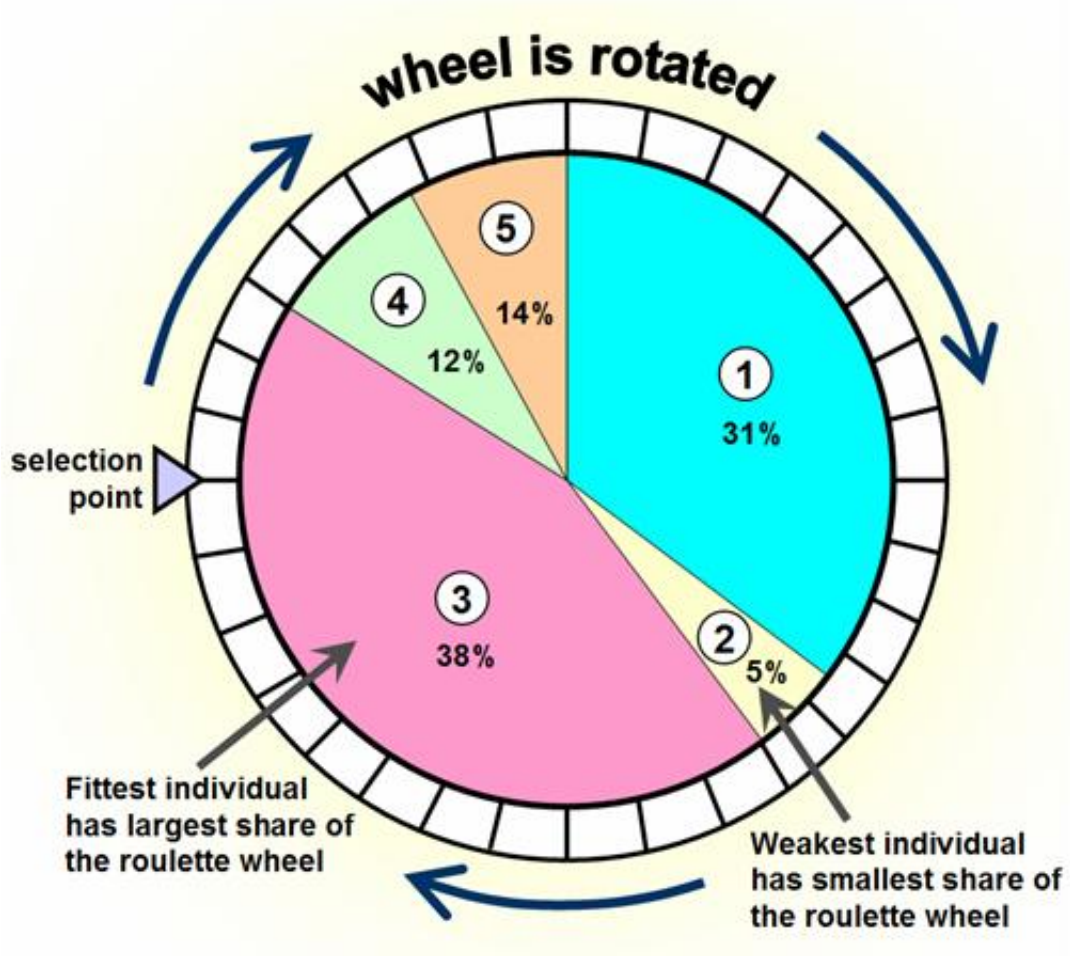
GA- Flow chart



https://plos.figshare.com/articles/_Flow_Chart_of_Genetic_Algorithm_with_all_steps_involved_from_beginning_until_termination_conditions_met_6_/1418786

GA , example

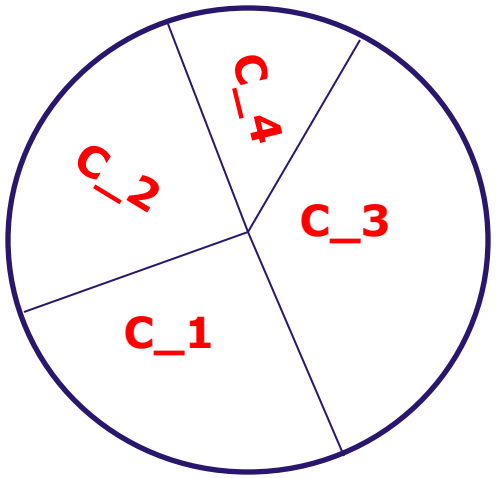
Selection (Roulette wheel):



GA , example

Selection (Roulette wheel):

i	Initial pop.	f_i	p_i
1	1432	22	0.252
2	3241	23	0.264
3	4123	18	0.207
4	1324	24	0.28
Total		87	

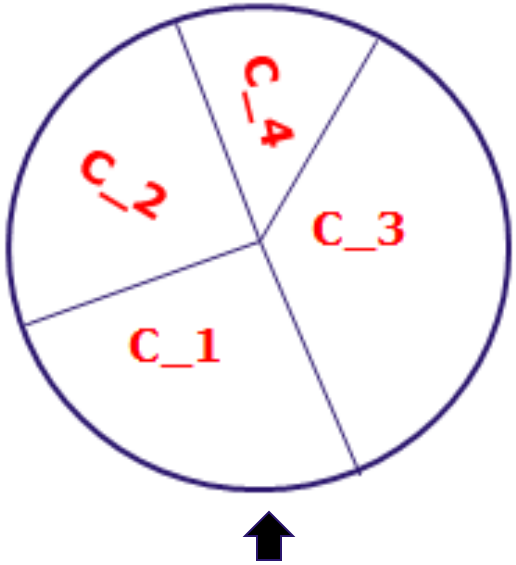


-Do not forget the objective here (minimization)

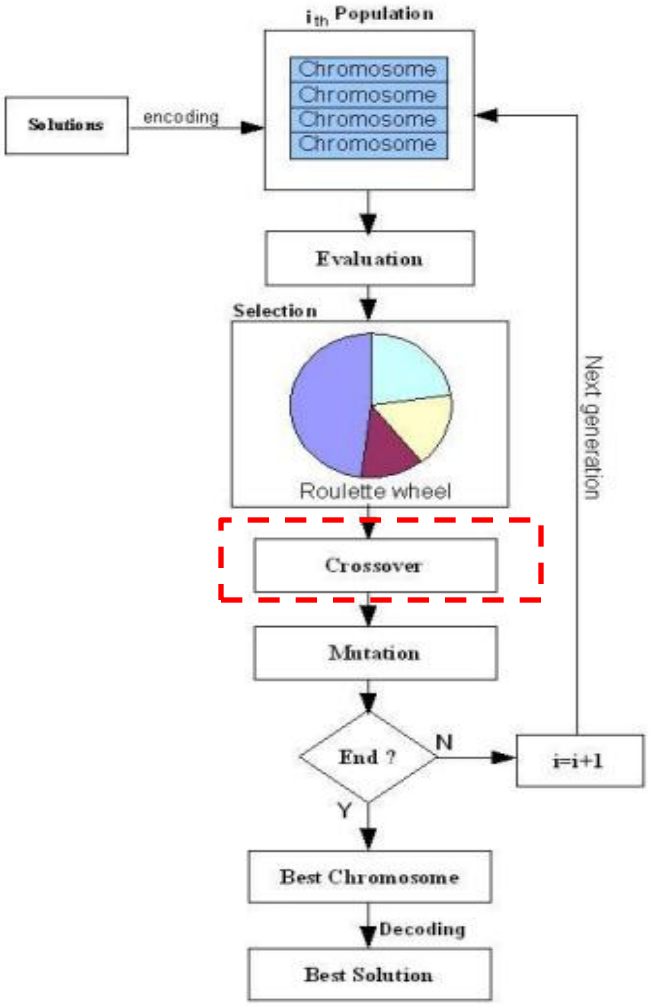
GA , example

Selection (Roulette wheel):

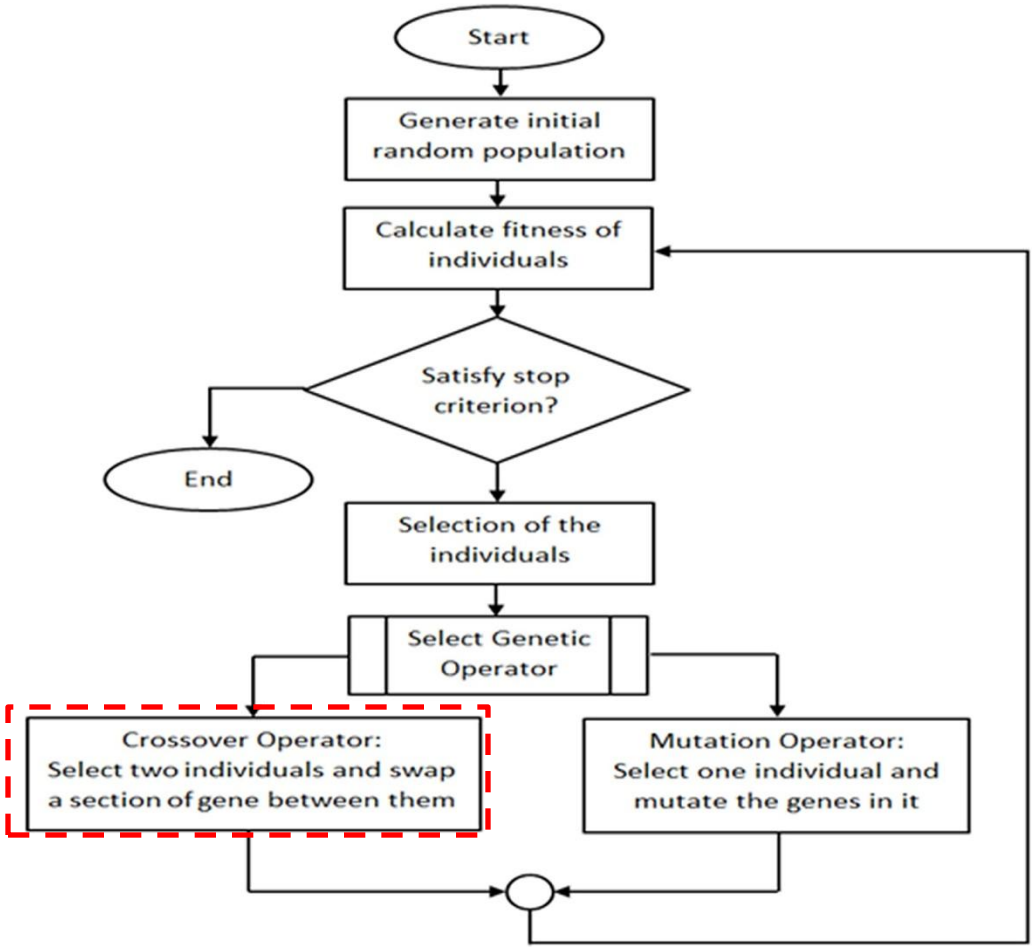
- Select random according to Roulette wheel



Genetic Algorithm (GA)



GA- Flow chart



https://plos.figshare.com/articles/_Flow_Chart_of_Genetic_Algorithm_with_all_steps_involved_from_beginning_until_termination_conditions_met_6_/1418786

GA , example

Crossover operation:

- Pair chromosomes are selected randomly for crossover operation
- Crossover point is selected randomly
- Paris of offspring's are generated
- Crossover is performed according to pre-fixed crossover probability

- For example, suppose we randomly selected chromosomes _**1** and _**3**

  chromosome _1 **1432**

  chromosome _3 **4123**

- Suppose that the crossover point is **2**

  chromosome _1 **14 | 32**

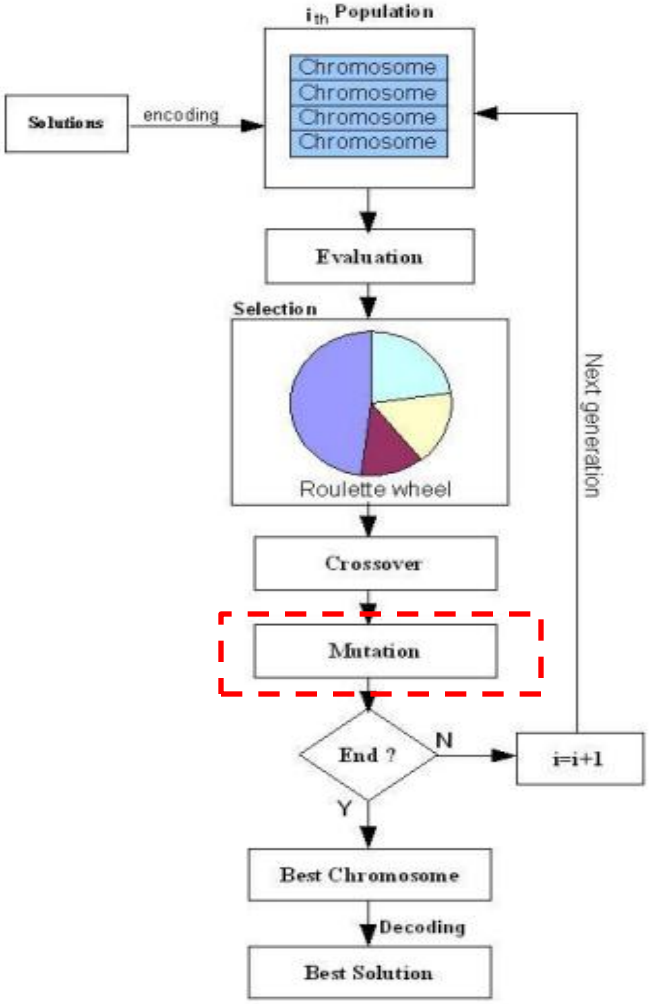
  chromosome _3 **41 | 23**

- Crossover takes place, generating two new strings:

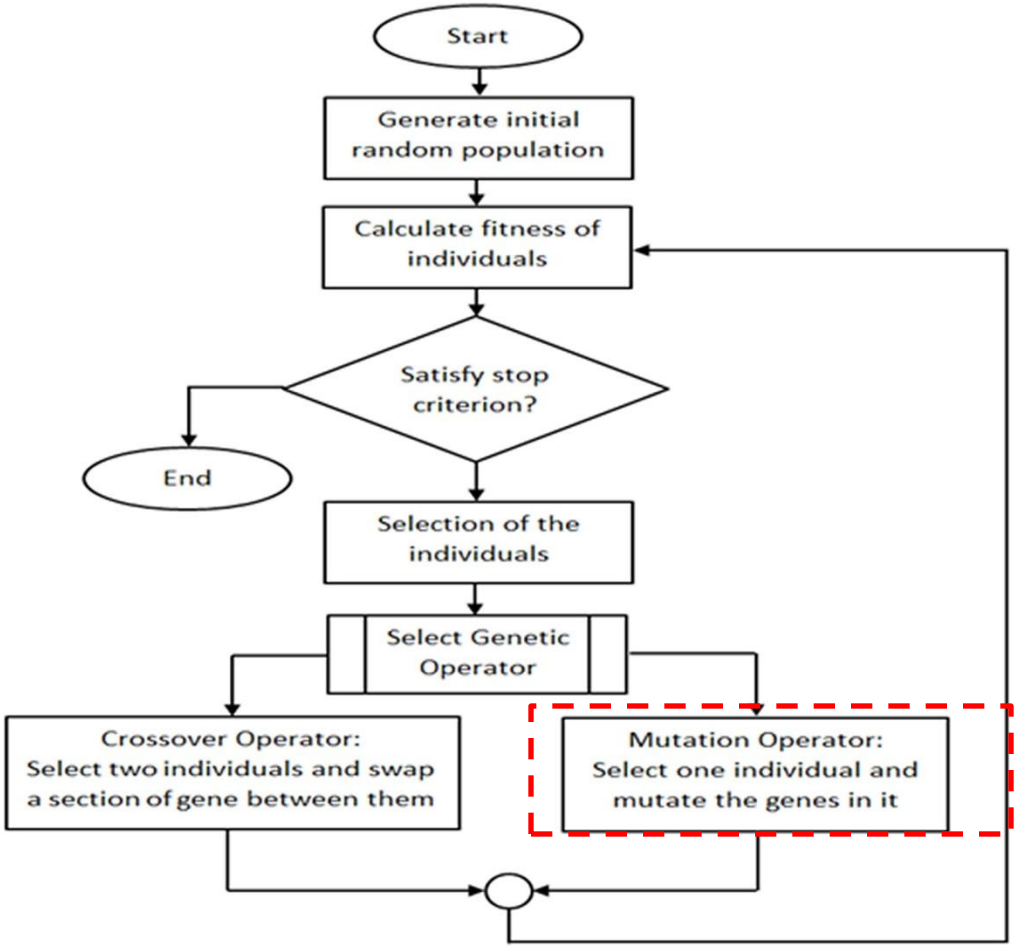
  offspring _1 **1423**

  offspring _2 **4132**

Genetic Algorithm (GA)



GA- Flow chart



https://plos.figshare.com/articles/_Flow_Chart_of_Genetic_Algorithm_with_all_steps_involved_from_beginning_until_termination_conditions_met_6_/1418786

GA , example

Mutation operation:

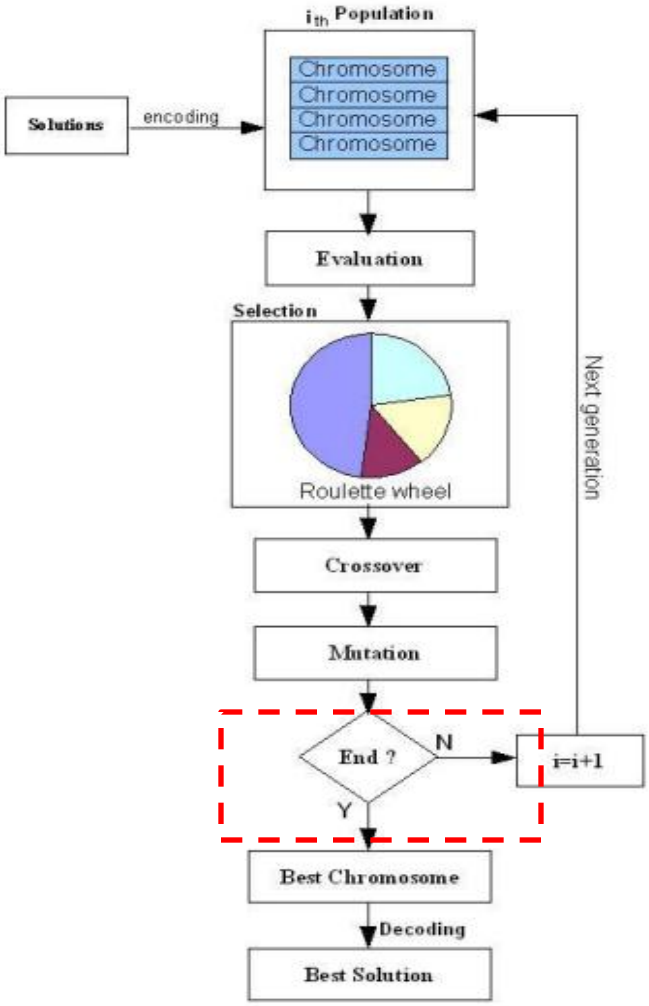
- Single chromosome is input for mutation operation
- Mutation point(s) are selected randomly
- Single offspring is generated
- Mutation is performed according to pre-fixed mutation probability
- Mutation occurs infrequently (very low probability)

For example,

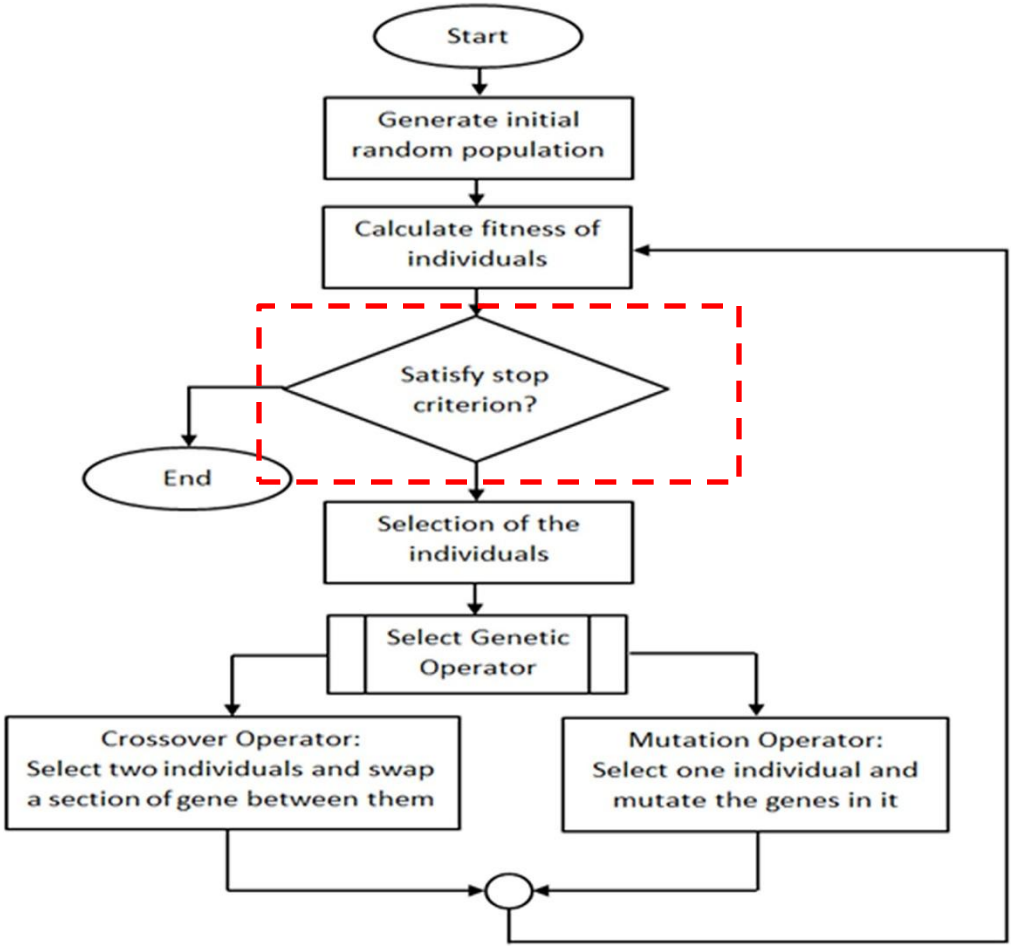
- Suppose that mutation operation is decided to be performed on offspring _1 "**1423**"
- Let the mutation points are **2,3**
- Swap operation is performed in locations 2, 3, the string change its value from "**1423**" to "**1243**"
- After the crossover and one mutation operations, we now have two offspring's for the new generation: **1243**

  4132

Genetic Algorithm (GA)



GA- Flow chart



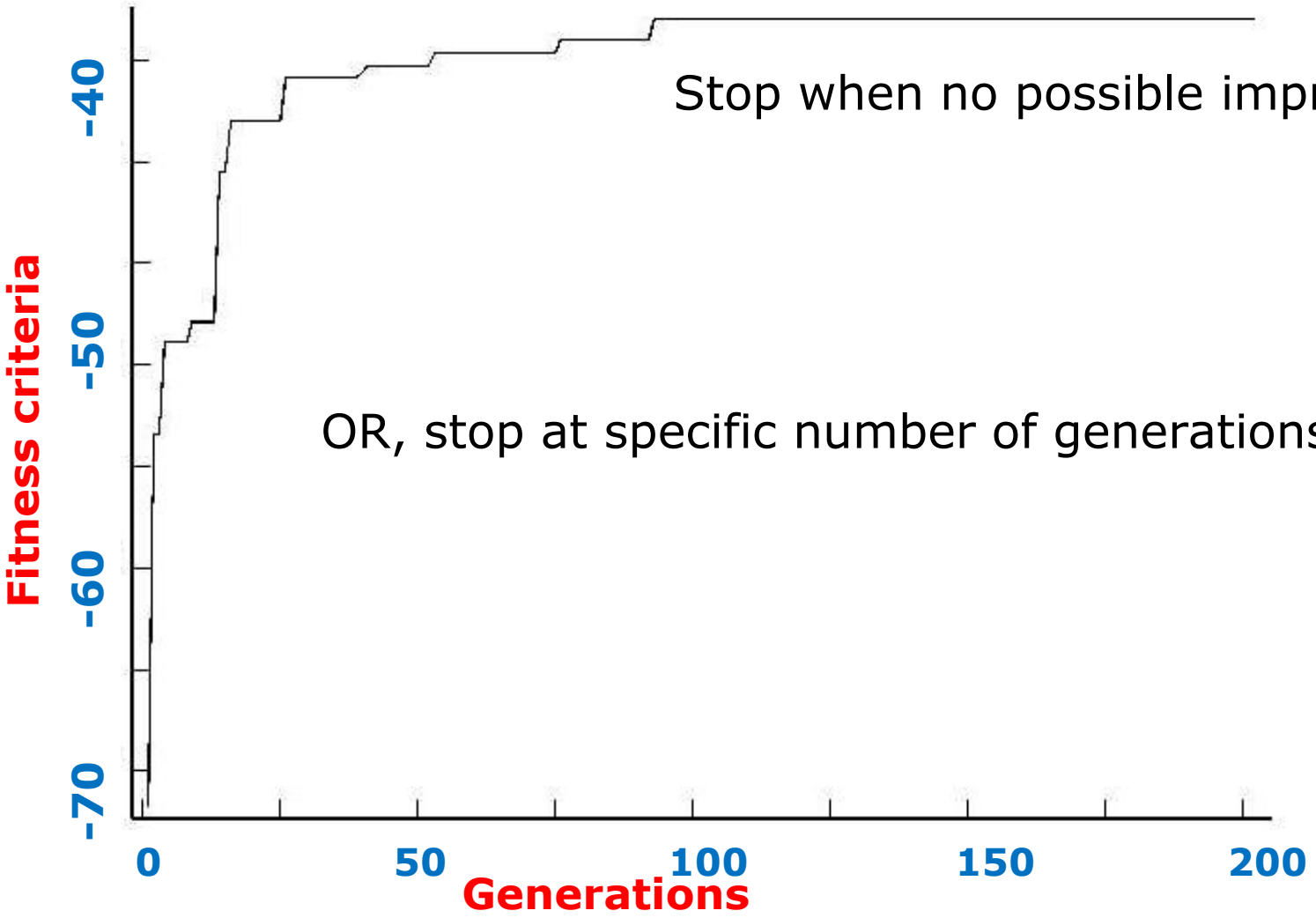
https://plos.figshare.com/articles/_Flow_Chart_of_Genetic_Algorithm_with_all_steps_involved_from_beginning_until_termination_conditions_met_6_/1418786

GA , example

- These operations are repeated until reach to the fixed size population
- Suppose that, chromosomes _2 and chromosomes _3 are selected for the crossover operation
  - Suppose that crossover is decided not to be performed
  - Suppose that also mutation is decided not to be performed
  - This case, the chromosomes 2 and 3 are copied to the new generation without modifications
- Now the new generation after performing these operations is:

Generation (2)		Generated by:
chromosome _1	**1243**	Crossover followed by mutation
chromosome _2	**4132**	Crossover operation
chromosome _3	**3241**	Copied form mating pool
chromosome _4	**4123**	Copied form mating pool

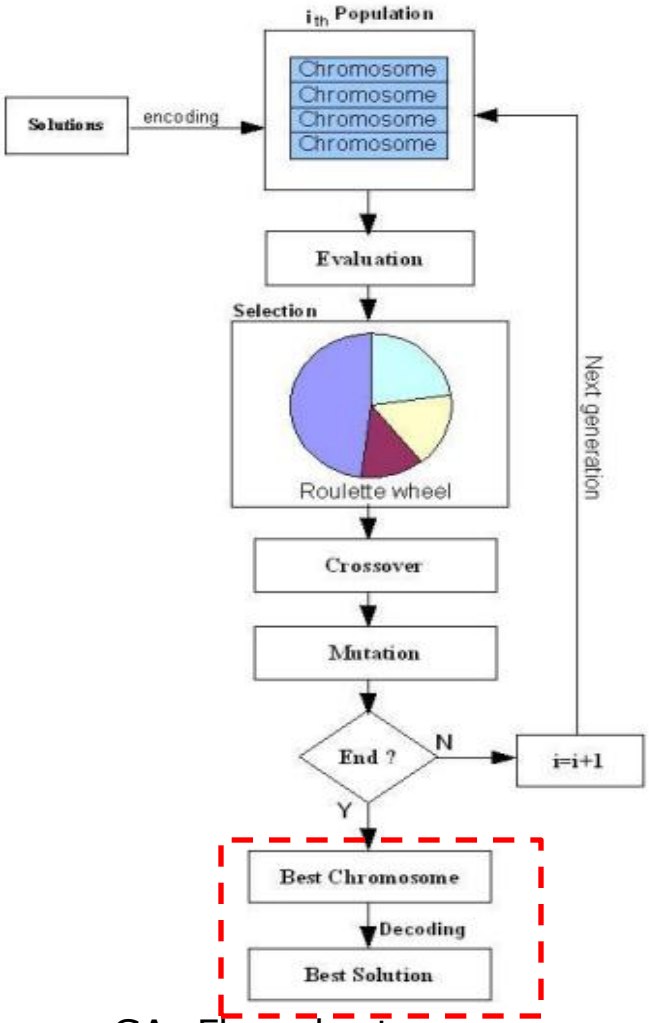
GA, Stopping condition



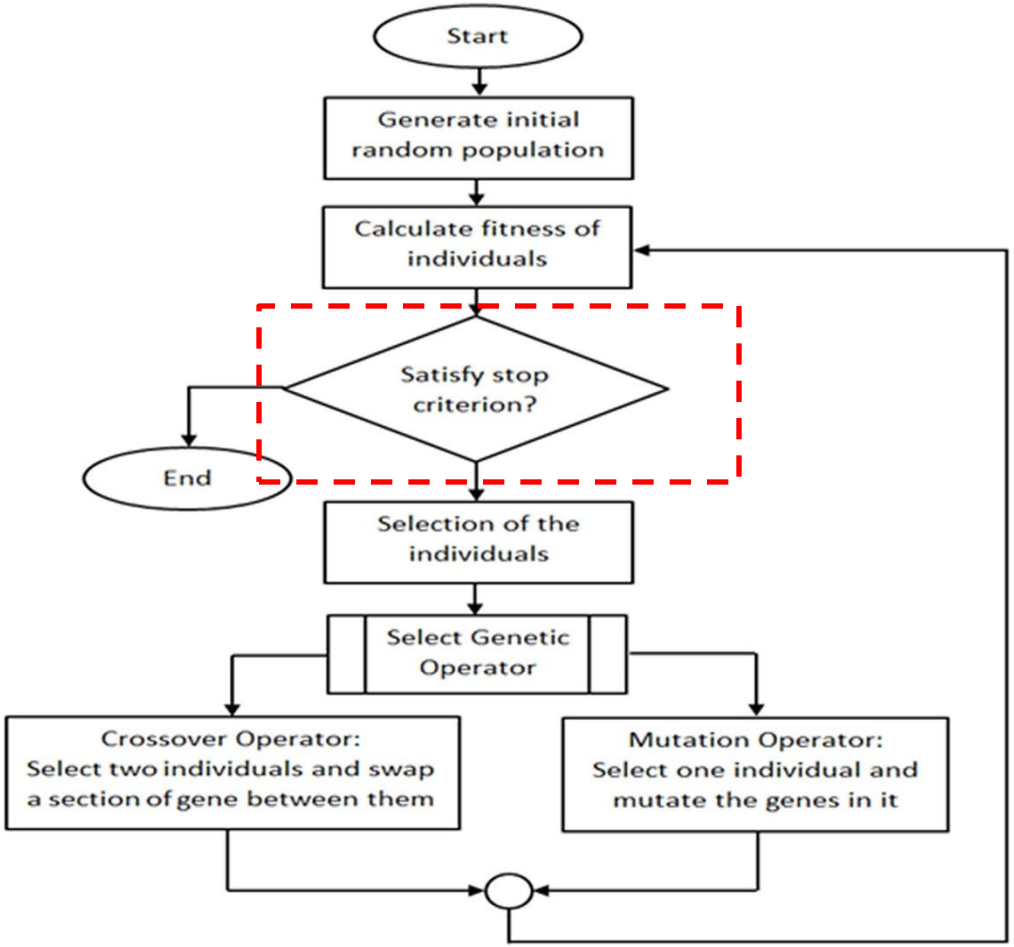
Stop when no possible improvement

OR, stop at specific number of generations

Genetic Algorithm (GA)



GA- Flow chart



https://plos.figshare.com/articles/_Flow_Chart_of_Genetic_Algorithm_with_all_steps_involved_from_beginning_until_termination_conditions_met_6_/1418786

GA , example

- The best chromosome in this final generation is the optimum solution found by GA-run

- Apply the tour represented by the best chromosome found in the final generation

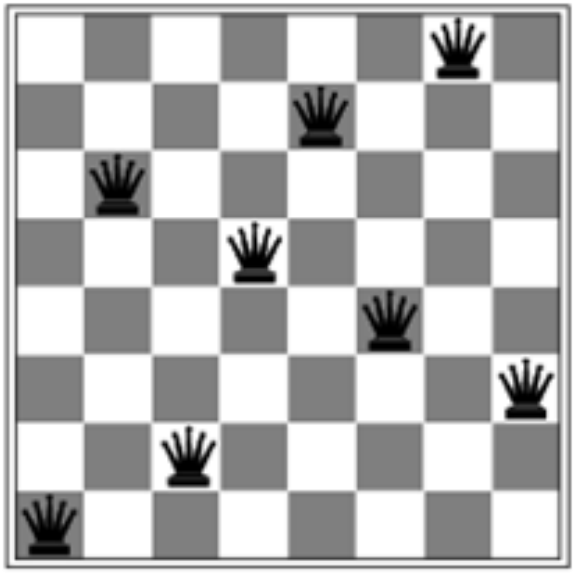
GA , basic algorithm

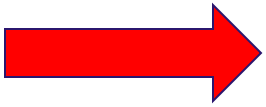
```
function GENETIC-ALGORITHM( population, FITNESS-FN) returns an individual
  inputs: population, a set of individuals
          FITNESS-FN, a function that measures the fitness of an individual

  repeat
      new_population ← empty set
      for i = 1 to SIZE( population) do
          x ← RANDOM-SELECTION( population, FITNESS-FN)
          y ← RANDOM-SELECTION( population, FITNESS-FN)
          child ← REPRODUCE(x, y)
          if (small random probability) then child ← MUTATE(child)
          add child to new_population
      population ← new_population
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to FITNESS-FN
```

GA , example

- N-queen example

Solution Encoding:





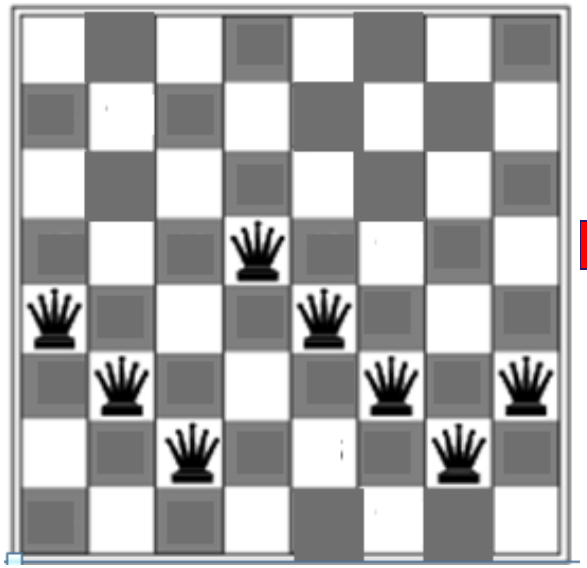
1	6	2	4	7	4	8	3

GA , example

- N-queen example

Evaluation (Fitness Function)

fitness=number of pairs of queens that are attacking each other, either directly or indirectly



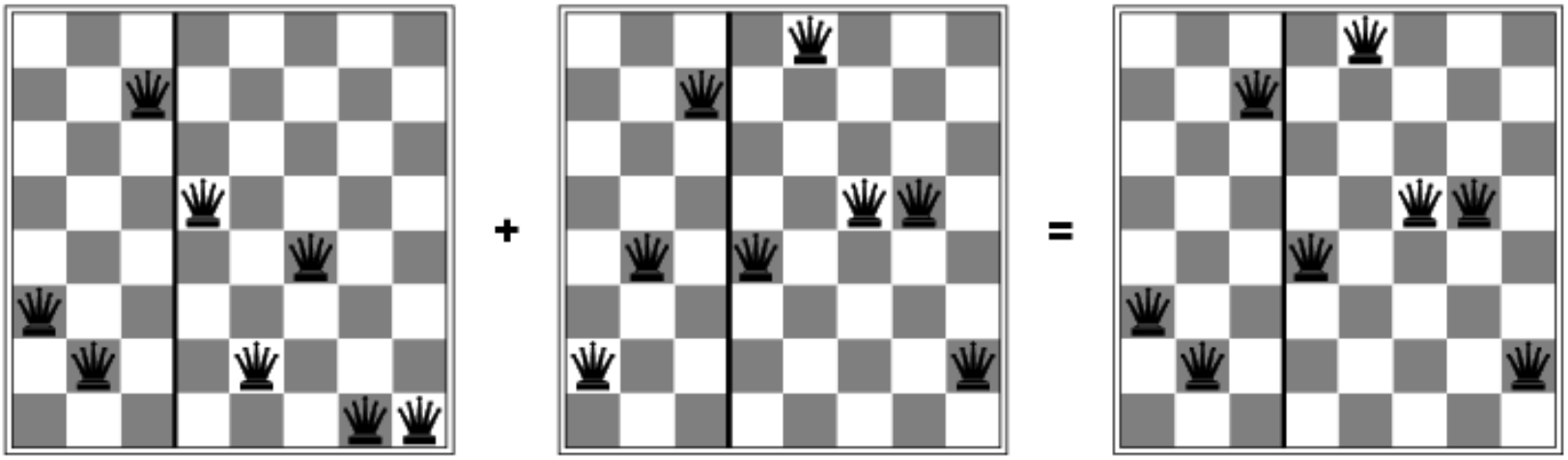
Fitness= 17

The target is minimization

GA , example

- N-queen example

Cross over operation

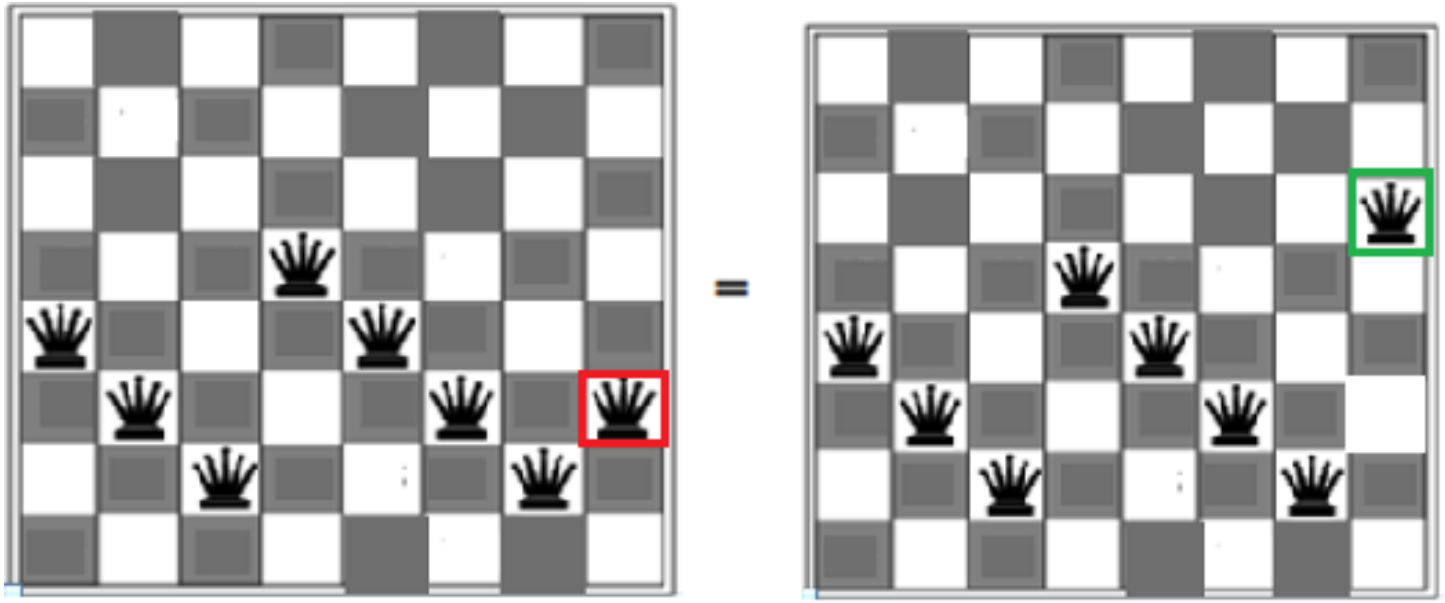


Recall, this is just the first offspring for this crossover!

GA , example

- N-queen example

Mutation operation



Bio-inspired algorithms

- Genetic Algorithm,(1974)
- Ant Colony Optimization,(1992)
- Genetic Programming,(1994)
- Particle swarm optimization,(1998)
- Fish Swarm Algorithm,(2012)
- Artificial Bee Colony Algorithm,(2005)
- Ant Lion Optimizer,(2014)
- Grey wolf optimizer,(2014)


Thank You !