Smart Systems

Eng. Khadija M.Hesham

Genetic Algorithm

Genetic Algorithms (GAs) are search based algorithms based on the concepts of natural selection and genetics.

GAs are a subset of a much larger branch of computation known as Evolutionary Computation.

Genetic Algorithms are sufficiently randomized in nature, but they perform much better than random local search (where we just try random solutions, keeping track of the best).

Genetic Algorithm

A genetic algorithm belongs to a class of evolutionary algorithms that is broadly inspired by biological evolution.

Biological evolution it is a selection of parents, reproduction, and mutation of offsprings. The main aim of evolution is to reproduce offsprings that are biologically better than their parents.

A genetic algorithm mainly bases on Darwin’s Theory of Evolution by Natural Selection, and it tries to simulate the same.

Applications

- Filtering and signal processing
- Finding hardware bugs.
- Image processing: Dense pixel matching

How it works

- we have a pool of possible solutions to the given problem.
- Solutions undergo recombination and mutation (like in natural genetics), produces new children,
- Each individual (or candidate solution) is assigned a fitness value (based on its objective function value)
- the fitter individuals are given a higher chance to mate and yield fitter individuals. (Darwinian Theory of Survival of the Fittest).
- the process is repeated for various generations and it keeps evolving better individuals or solutions over generations, till it reaches a stopping criterion.

Steps in a Genetic Algorithm

1. Initialize population
2. Select parents by evaluating their fitness
3. Crossover parents to reproduce
4. Mutate the offsprings
5. Evaluate the offsprings
6. Merge offsprings with the main population and sort

Initialize population

Starts with the randomly generated population.

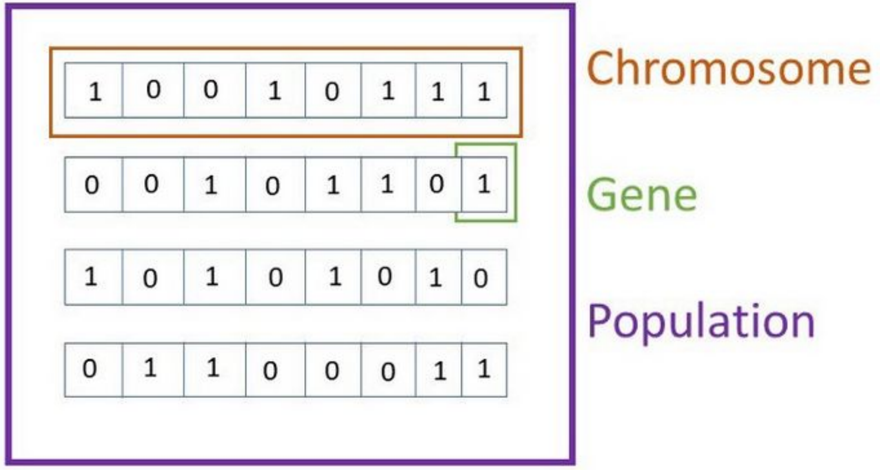
Population use 0s and 1s

Encoding or, uniformly

distributed numbers

(used in the code)

to represent each gene.



Parent Selection

a portion of the existing population is selected to breed a new generation.

Individual solutions are selected through a fitness-based process.

In generation 0, we do not have offsprings, We select parents from our randomly generated population.

There are three main methods to define the best fit individuals and select for breeding.

Parent Selection

Random selection:

Shuffles the population by performing permutation and select the first two individuals as parents

not recommended , it does not follow “Darwin’s Theory of Evolution by Natural Selection,” wherein individuals are selected based on their fitness, not randomly.

Parent Selection

Tournament selection:

Based on the probability of selection of each individual.

The probability of each individual is called the fitness of the individual.

The algorithm is done by tournaments among a randomly selected group of individuals.

select one individual from each group as the winner, and again run the tournament by grouping winners from the first iteration.

The best member of each group in every iteration has the highest probability of selection.

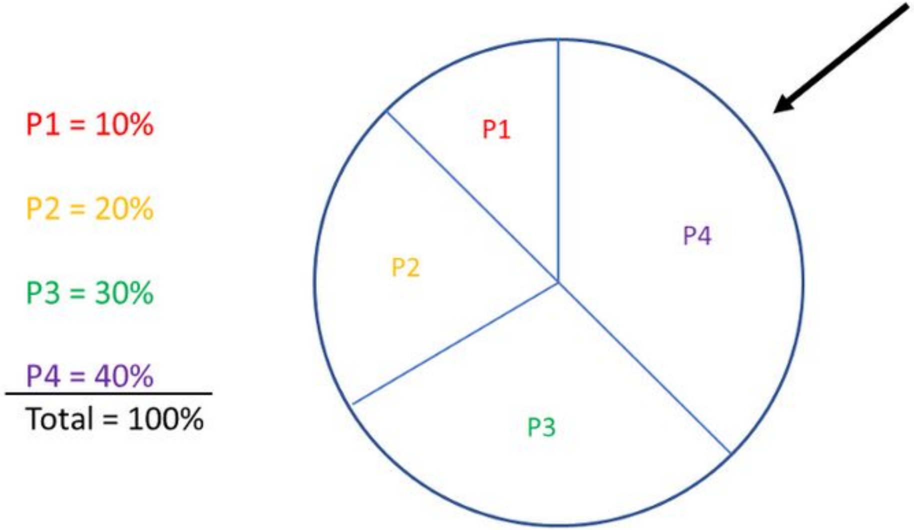
repeat the process until we converge to two winners parents.

Parent Selection

Roulette wheel selection:

widely used and most efficient

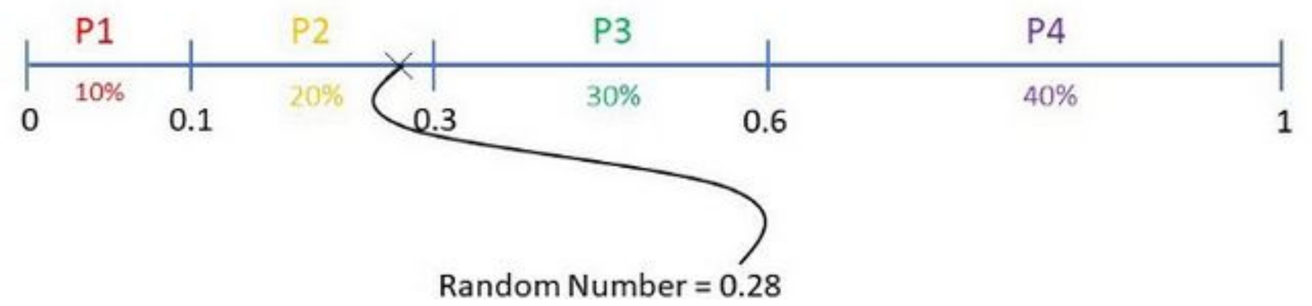
each parent has an equal probability of selection in the roulette, but in GA we define the probability for each parent.



Parent Selection

Roulette wheel selection:

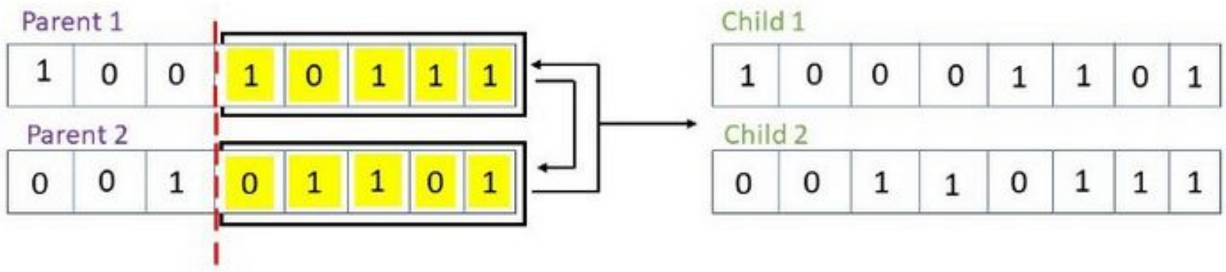
To choose the parent, we Generate a random number between 0 and 1 will act like the arrow.



Crossover

Single-point crossover:

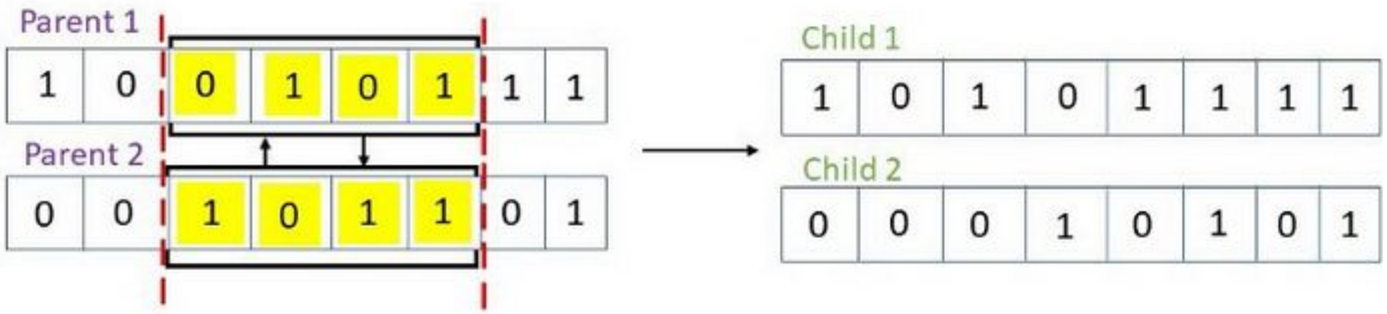
parents chromosomes are cut at the same random point, and the leftover parts are swapped to produce two new offspring chromosomes. Yellow-colored genes represent the cutoff part of the chromosome.



Crossover

Two-point crossover:

Same as before, the only difference is that the parent chromosomes are cut at two random points.



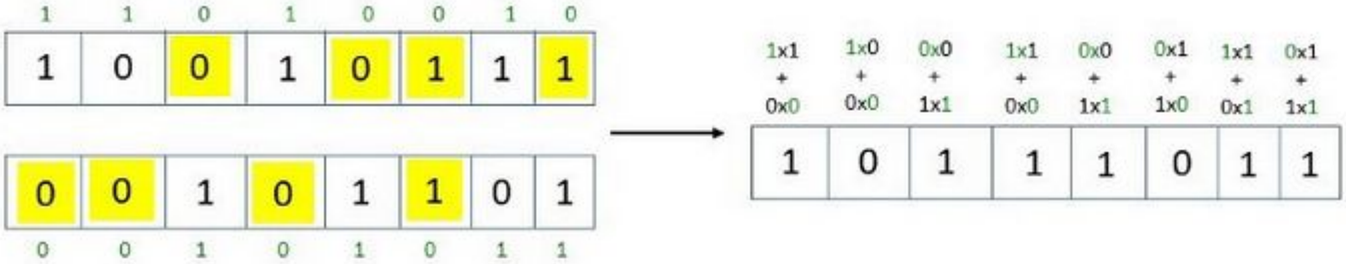
Crossover

Uniform crossover:

randomly choose which genes to be inherited from the parent chromosomes .

The gene to be inherited is encoded as 1, and the gene that should not be inherited is encoded as 0. (alpha value)

Multiply the gene value with the corresponding alpha value for both the parents and then add the results to generate a single gene



Mutation

Mutation is a natural process that occurs due to an error in replication or copying of genes.

While performing crossover, we replicated the parent chromosomes by mix-matching genes of both the parents. There is no guarantee that the copying of the parent gene is 100% accurate.

Mutating the chromosome in the genetic algorithm is necessary because it may result in revolutionary results.

Mutation

Three parameters: the child chromosome(c), the mutation rate(mu), and step size of mutation (sigma).

The mutation rate(mu) determines the percentage of the child chromosome to be applied mutation on.

(mutated) genes generated by multiplying the step size(sigma) with randomly generated value and adding it to the original gene.

Mutation

How it works:

To define which genes will be mutated,

we generate random numbers and compare them to the mutation rate.

we find the indices of the child chromosome(position) that have values less than the mutation rate.

Replace those indices with new mutated genes generated by multiplying the step size(sigma) with randomly generated value and adding it to the original gene.

Evaluating the Offsprings

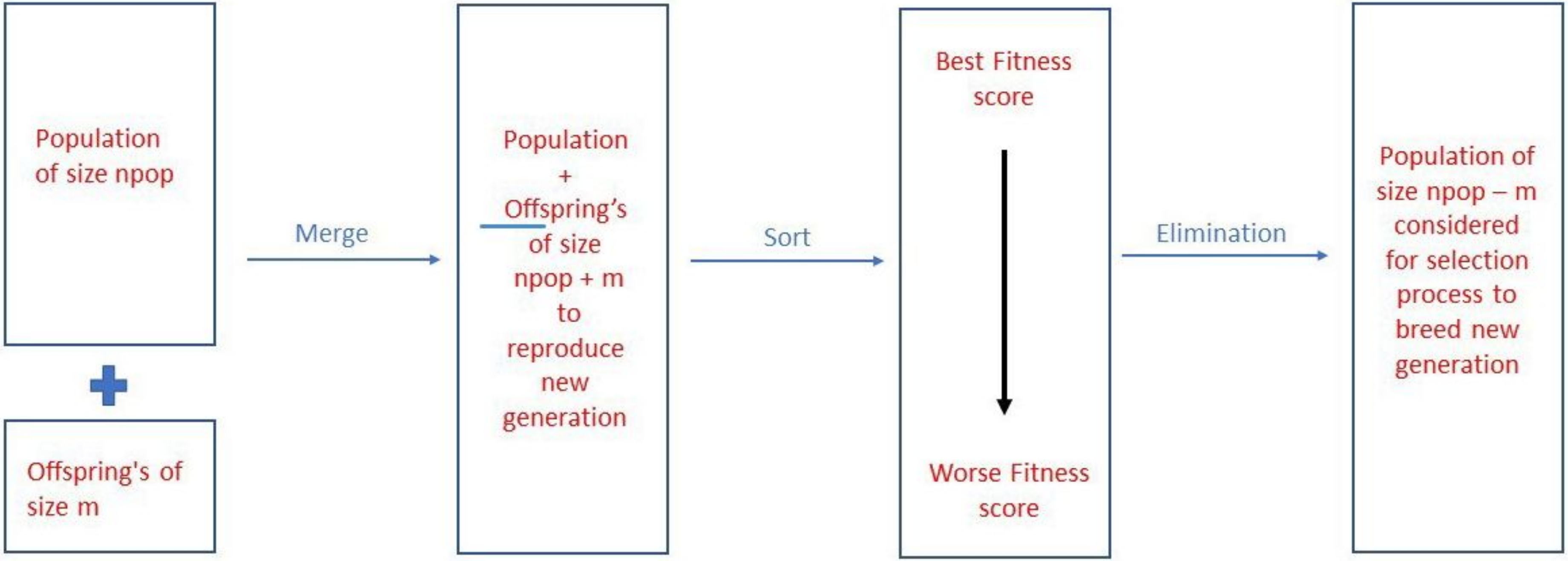
After mutation, we evaluate offsprings with the cost function to define their fitness. Also, replace the best solution in every generation/iteration.

Merge Offsprings with the Main Population and Sort

Merging the offsprings is vital for them to be considered as parents to reproduce the next generation.

sorting ranks better individuals at the top.

Merge Offsprings with the Main Population and Sort



References

[1] Artificial Intelligence with python - tutorialspoint.

[2] https://pub.towardsai.net/genetic-algorithm-ga-introduction-with-example-code-e59f9bc58eaf

Code Example (OneMax Problem):

https://colab.research.google.com/github/jfogarty/machine-learning-intro-workshop/blob/master/notebooks/genetic-algorithms-with-deap.ipynb#scrollTo=STsQs-Hg_JaT

GA Implementation (optional):

https://colab.research.google.com/drive/161ijkvn8wG_seVtQexm-p3fW3r5p8s_x?usp=sharing