Genetic Algorithm/ Evolutionary programming

What are Genetic Algorithms?

- Genetic Algorithm (GA) is a search-based optimization technique (process of making something better) based on the principles of **Genetics and Natural Selection**.

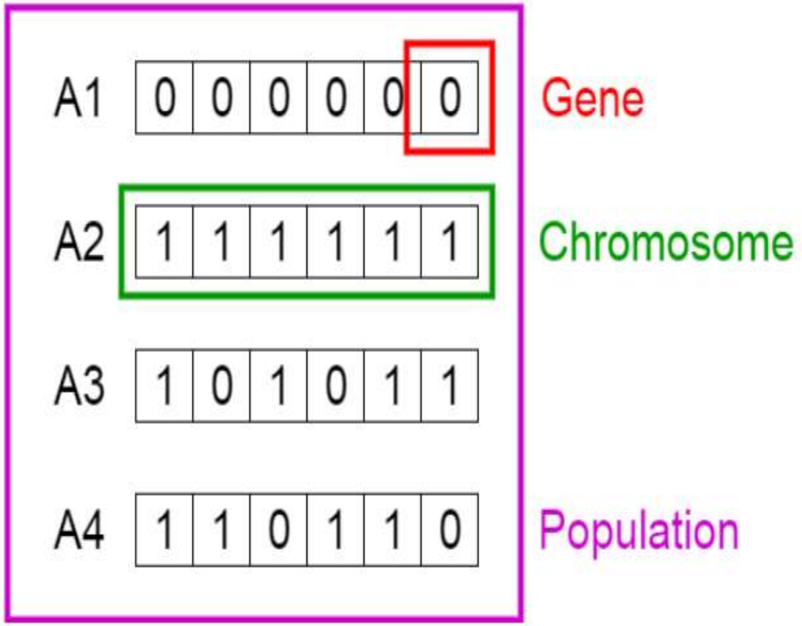
- Reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of next generation.

- GAs were developed by **John Holland** and his students and colleagues at the University of Michigan.

GENETIC ALGORITHM

- Five phases are considered in a genetic algorithm.
  1. Initial population
  2. Fitness function
  3. Selection
  4. Crossover
  5. Mutation

1. **Initial population**
    - Begins with a set of individuals which is called a **Population**.
    - Individual is characterized by a set of parameters (variables) known as **Genes**.
    - Genes are joined into a string to form a **Chromosome.**
    - Set of genes of an individual is represented using a string, in terms of an alphabet.



Population, Chromosomes and Genes

2. **Fitness Function**

- Determines how fit an individual is.
- Gives a **fitness score** to each individual.

3**. Selection**

- Select the fittest individuals and let them pass their genes to the next generation.

4. **Crossover**

- Parents to be mated.
- **crossover point** is chosen at random from within the genes.
- **Offspring** are created by exchanging the genes of parents.

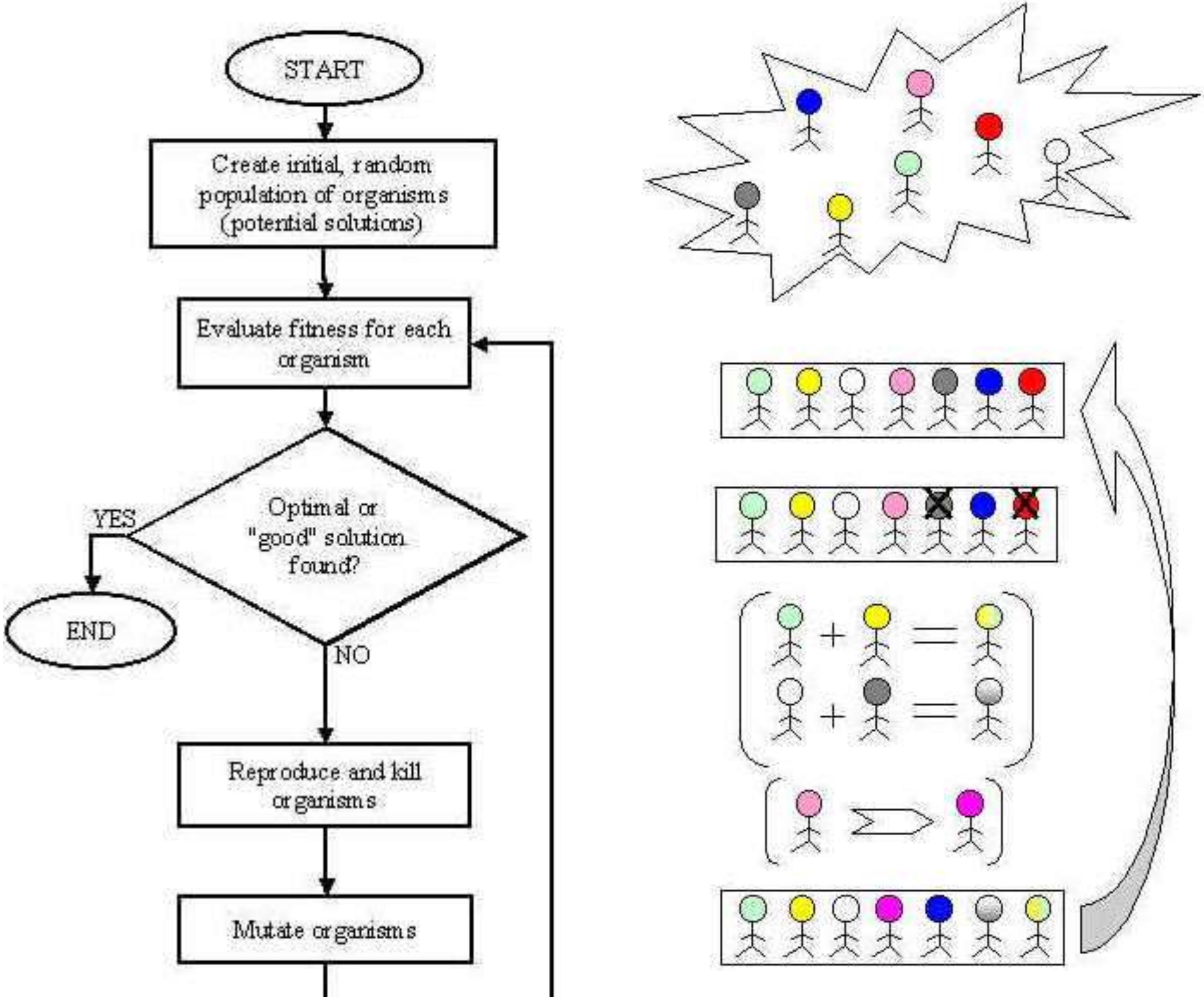
5. **Mutation**

- In certain new offspring formed, some of their genes can be subjected to a **mutation** with a low random probability.

- **Genotype** − Genotype is the population in the computation space. In the computation space, the solutions are represented in a way which can be easily understood and manipulated using a computing system.

- **Phenotype** − Phenotype is the population in the actual real world solution space in which solutions are represented in a way they are represented in real world situations.

- **Decoding and Encoding** − Decoding is a process of transforming a solution from the genotype to the phenotype space, while encoding is a process of transforming from the phenotype to genotype space. Decoding should be fast as it is carried out repeatedly in a GA during the fitness value calculation.



START
Create initial, random population of organisms (potential solutions)
Evaluate fitness for each organism
Optimal or "good" solution found?
YES
END
NO
Reproduce and kill organisms
Mutate organisms

GENETIC ALGORITHM--POPULATION

- Population is a subset of solutions in the current generation. It can also be defined as a set of chromosomes.
- The population is usually defined as a two dimensional array of – **population size, chromosome size**.
- Population Initialization: There are two primary methods to initialize a population in a GA. They are −
  - **Random Initialization** − Populate the initial population with completely random solutions.
  - **Heuristic initialization** − Populate the initial population using a known heuristic for the problem.

GENETIC ALGORITHM—FITNESS FUNCTION

- The fitness function is a function which takes a **candidate solution to the problem as input and produces as output** how "fit" our how "good" the solution is with respect to the problem in consideration.

- A fitness function should possess the following characteristics −
  1. The fitness function should be sufficiently fast to compute.
  2. It must quantitatively measure how fit a given solution is or how fit individuals can be produced from the given solution.

GENETIC ALGORITHM—PARENT SELECTION

- Parent Selection is the process of selecting parents which mate and recombine to create off-springs for the next generation.

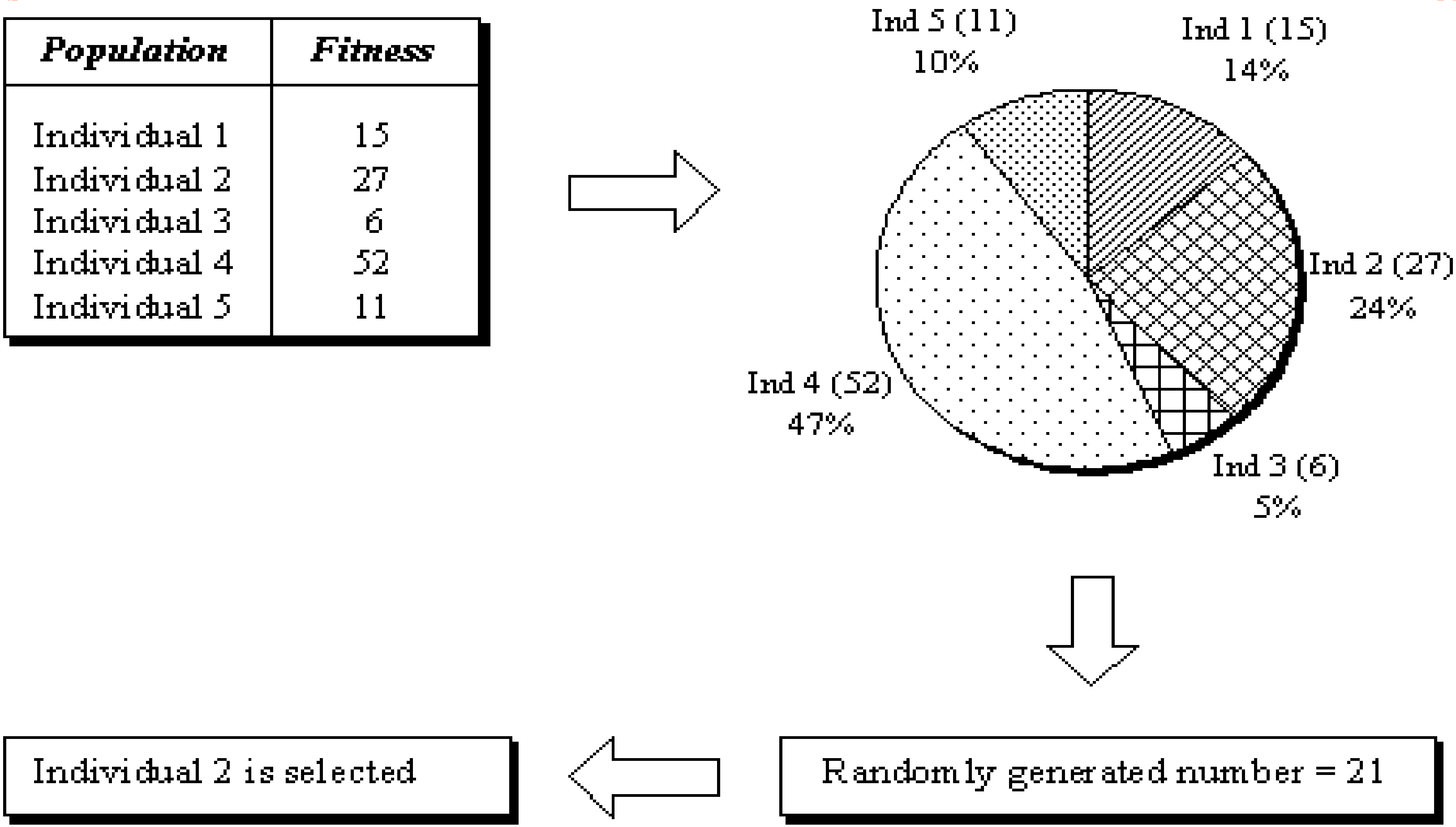
- Parent selection is very crucial to the convergence rate of the GA as good parents drive individuals to a better and fitter solutions.
- Method used for Selection:

    Roulette Wheel Selection

    Boltzmann Selection

    Tournament Selection

    Rank Selection



Roulette Wheel Selection: **Depending on percentage contribution to the total population fitness, string is selected for mating to form the next generation.**

Genetic Algorithm -- Crossover

- The crossover operator is analogous to reproduction and biological crossover.

- In this more than one parent is selected and one or more off-springs are produced using the genetic material of the parents.

- Crossover is usually applied in a GA with a high probability – $\boldsymbol{p_c}$.

One Point Crossover

In this one-point crossover, a random crossover point is selected and the tails of its two parents are swapped to get new off-springs.

Multi Point Crossover

Multi point crossover is a generalization of the one-point crossover wherein alternating segments are swapped to get new off-springs.

Uniform Crossover

In a uniform crossover, we don't divide the chromosome into segments, rather we treat each gene separately. In this, we essentially flip a coin for each chromosome to decide whether or not it'll be included in the off-spring. We can also bias the coin to one parent, to have more genetic material in the child from that parent.

Genetic Algorithm -- Mutation

- Mutation may be defined as a small random tweak in the chromosome, to get a new solution.

- It is used to maintain and introduce diversity in the genetic population and is usually applied with a low probability – $\boldsymbol{p_m}$.

- Mutation is the part of the GA which is related to the "exploration" of the search space.

Mutation Operator

Bit Flip Mutation

In this bit flip mutation, we select one or more random bits and flip them. This is used for binary encoded GAs.

0	0	1	**1**	0	1	0	0	1	0

=>

0	0	1	**0**	0	1	0	0	1	0

Random Resetting

Random Resetting is an extension of the bit flip for the integer representation. In this, a random value from the set of permissible values is assigned to a randomly chosen gene.

Swap Mutation

In swap mutation, we select two positions on the chromosome at random, and interchange the values. This is common in permutation based encodings.

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

=>

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

Scramble Mutation

Scramble mutation is also popular with permutation representations. In this, from the entire chromosome, a subset of genes is chosen and their values are scrambled or shuffled randomly.

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

=>

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

Survivor Selection

- The Survivor Selection Policy determines which individuals are to be kicked out and which are to be kept in the next generation.

- It is crucial as it should ensure that the fitter individuals are not kicked out of the population, while at the same time diversity should be maintained in the population.

TERMINATION

- The termination condition of a Genetic Algorithm is important in determining when a GA run will end. It has been observed that initially, the GA progresses very fast with better solutions coming in every few iterations, but this tends to saturate in the later stages where the improvements are very small.

Usually, we keep one of the following termination conditions –

- When there has been no improvement in the population for X iterations.
- When we reach an absolute number of generations.
- When the objective function value has reached a certain pre-defined value.

EXAMPLE

Maximize the function $f(x)=x^2$ with x in interval [0,31]

x= 0,1,………….30,31

1. Generate **initial population** at random. They are chromosomes or genotypes.

   eg. 13, 24, 8, 19

2. **Calculate fitness**

   a) Decode into an integer (called phenotypes)

      01101 (13), 11000 (24), 01000 (8), 10011 (19)

   b) Evaluate fitness $f(x)= x^2$

      13 – 169, 24 – 576, 8 – 64, 19 – 361

3. Select parents(2 individuals) based on their fitness in P:

$$P_i = F_i / (\sum_{j=1}^{n} F_j)$$

F_i : Fitness for String i in population, expressed as f(x)

P_i : Probability of String i being selected.

n: Number of individuals in the population

n* P_i : Expected count

String No.	Initial Population	X value	Fitness F_j $f(x)=x^2$	P_i	Expected count
1	01101	13	169	0.14	0.56
2	11000	24	576	0.49	1.97
3	01000	8	64	0.06	0.22
4	10011	19	361	0.31	1.23
SUM			1170	1.00	4.00

Cross-Over Operator:

Can be of either one-point or multipoint crossover.

String No.	Mating Pool	Crossover point	Offspring after Crossover	n value	Fitness $f(x)=x^2$
1	01101	4	01100	12	144
2	11000	4	11001	25	625
3	11000	2	11011	27	729
4	10011	2	10000	16	256
SUM					1754

Mutation:

Applied to each child individually after crossover.

Bits are changed from 0 to 1 or from 1 to 0 at randomly chosen position of randomly selected strings.

String No.	Offspring after Crossover	Offspring after Mutation	n value	Fitness function $f(x)=x^2$
1	01100	11100	28	784
2	11001	11001	25	625
3	11011	11011	27	729
4	10000	10100	20	400
SUM				2538

- Compare the fitness value, initially it is 1170 and after performing genetic algorithm it comes out to be 2538.
- So we get better results after applying genetic algorithm.

EXAMPLE

Suppose a genetic algorithm uses chromosomes of the form $x = abcdefgh$ with a fixed length of eight genes. Each gene can be any digit between 0 and 9. Let the fitness of individual x be calculated as:

$$f(x) = (a + b) - (c + d) + (e + f) - (g + h) \ ,$$

and let the initial population consist of four individuals with the following chromosomes:

$$\begin{aligned} x_1 &= 6\,5\,4\,1\,3\,5\,3\,2 \\ x_2 &= 8\,7\,1\,2\,6\,6\,0\,1 \\ x_3 &= 2\,3\,9\,2\,1\,2\,8\,5 \\ x_4 &= 4\,1\,8\,5\,2\,0\,9\,4 \end{aligned}$$

a) Evaluate the fitness of each individual, showing all your workings, and arrange them in order with the fittest first and the least fit last.

b) Perform the following crossover operations:

i) Cross the fittest two individuals using one–point crossover at the middle point.

ii) Cross the second and third fittest individuals using a two–point crossover (points b and f).

By looking at the fitness function and considering that genes can only be digits between 0 and 9 find the chromosome representing the optimal solution (i.e. with the maximum fitness). Find the value of the maximum fitness.