Genetic Algorithm

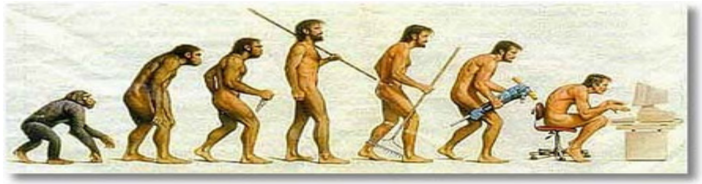
Er. RUDRA NEPAL

Background

- On 1 July 1858, Charles Darwin , presented his theory of evolution. This day marks the beginning of a revolution in Biology.
- Darwin's classical theory of evolution , together with Weismann's theory of natural selection and Mandel's concept of genetics , now represent the Neo-Darwinism
- Neo=Darwinism is based on process of reproduction, mutation, competition and selection.

Background

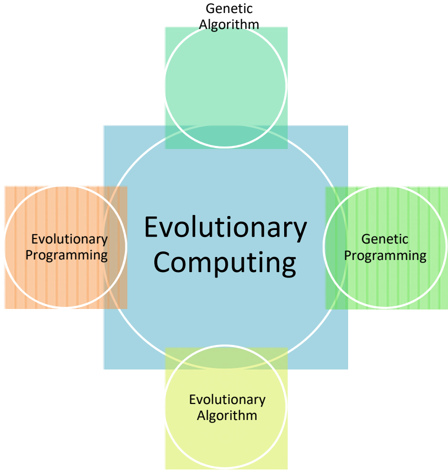
- Evolution can be seen as a process leading to the maintenance of a population's ability to survive and reproduce in a specific environment. This ability is called evolutionary fitness.
- Evolutionary fitness can also be viewed as a measure of the organism's ability to anticipate changes in its environment.
- The fitness, or the quantitative measure of the ability to predict environmental changes and respond adequately, can be considered as the quality that is optimized in natural life



Evolutionary Computation

- Evolutionary Computation stimulates evolution on a computer. The result of such simulations is a sense of optimisation algorithms
- Optimisation iteratively improves the quality of solutions until an optimal, or near-optimal, solution is found
- The evolutionary approach is based on computational models of natural selection and genetics. We call them evolutionary computation ,an umbrella term that combines genetic algorithms, evolutionary programming, evolutionary algorithms and genetic programming

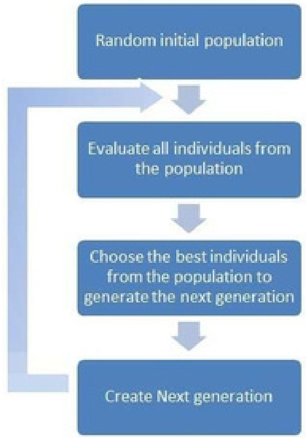
Evolutionary Computation



Evolutionary Algorithm (EA)

- EA are stochastic search and optimization heuristics derived from the classic evolution theory, which are implemented on computers in the majority of cases.
- Basic Idea
  - If only those individuals of a population reproduce, which meet a certain selection criteria, and other individuals of the population die, the population will converge to those individuals that best meet the selection criteria.
  - Population dynamics follow the basic rule of Darwin evolution theory, which can be described in short as the “survival of the fittest.”

EA General Process



Genetic Algorithm

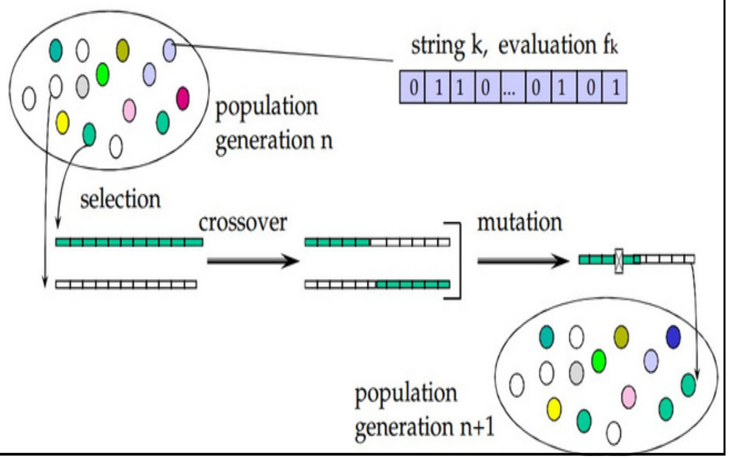
- In early 1970s John Holland introduced the concept of genetic algorithm
- His aim was to make computers do what nature does. Holland was concerned with algorithms that manipulate strings of binary digits
- Each artificial "chromosomes" consists of a number of "genes", and each gene is represented by0or1

1	0	1	1	0	1	0	0	0	0	0	1	0	1	0	1

Genetic Algorithm

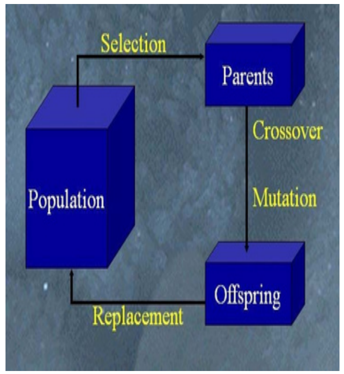
- The genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution.
- The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.
- You can apply the genetic algorithm to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear.

Basic Genetic Algorithm



Genetic Algorithms: Process

- The states with best value of fitness function is selected for Reproduction
- Crossoverpoint is randomly chosen for crossing of strings (parents), which yields offspring for each generation
- Each individual of next generation offspring is subjected to random mutation with a small independent probability
- The mutated offspring are added to the population (next generation) and the process is repeated until the goal state (solution) is obtained



Genetic Algorithm Operators

- Fitness function: The fitness function is defined over the genetic representation and measures the *quality* of the represented solution.
- Selection Operator: Selects parents for reproduction based on relative fitness of candidates in the population
  - Roulette Wheel Selection
  - Ranking Selection

Genetic Algorithm Operators

- Crossover Operator:
  - Exchanges part of chromosome between two parent chromosomes with some crossover rate(probability), typically0.4– 0.8
  - The main operator to provide exploitation in search building up good genes in chromosome
    - One Point Crossover: randomly chooses a crossover point where two parent chromosomes “break” and then exchanges the chromosome parts after that point. As a result, two new offspring is created
    - Two Point Crossover: randomly chooses two crossover points in two parent chromosomes, and then exchanges the chromosome parts between these points. As a result, two new offspring are created.

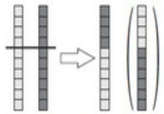


Fig. .a: Single-point Crossover (SPX).

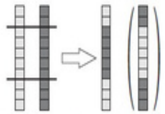


Fig. .b: Two-point Crossover (TPX).

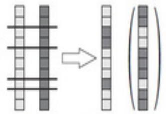
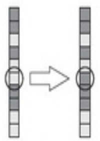


Fig. .c: Multi-point Crossover (MPX).

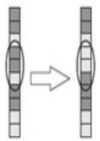
Genetic Algorithm Operators

- MutationOperator:
  - Changes a randomly selected gene in the chromosome
  - mimics random changes in genetic code Background
  - operator to provide exploration in search to avoid being trapped on a local optimum
  - Mutation probability is quiet small in nature and is kept low for GAs, typically in the range between [0.001 – 0.01] or by formula :

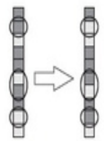
  p(m)=1/no.of bits in chromosomes



a: Single-gene mutation.



b: Multi-gene mutation



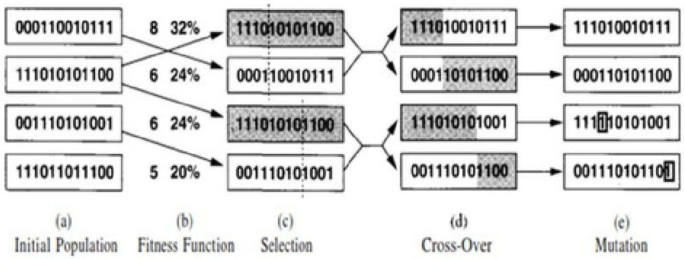
c: Multi-gene mutation

Genetic Algorithm Operators

- Elitism
  - Elitism refers to the safeguarding of the chromosome of the most fit individual in a given generation.
  - If elitism is used, only N-1 individuals are produced by recombining the information from parents. The last individual is a copy of the most fit individual from the previous generation.
  - This ensures that the best chromosome is never lost in the optimization process due to random events.

Genetic Algorithm Parameters

- Basic GA Parameters:
  - Population size Crossover rate (Probability)
  - Mutation Rate (Probability) Number of
  - Generation (a Stopping Criterion)
  -



Steps in Genetic Algorithm

1. Represent the problem variable as a chromosome of a fixed length, choose the size of a chromosome population N, the crossover probability p(c) and the mutation probability p(m) ( )

2. Define a fitness function to measure the fitness of an individual chromosome in the problem domain.

3. Randomly generate an initial population of chromosomes of size **N: x1, x2, x3...f(xN)**

4. Calculate the fitness of each individual chromosome: f(x1),f(x2)...f(xN)

Steps in Genetic Algorithm

5. Select a pair of chromosomes for mating from the current

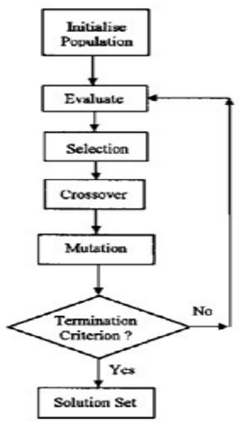
populationbasedontheirfitness.

6. Create a pair of offspring chromosomes by applying the genetic operators − crossover and mutation.
7. Place the created offspring chromosomes in the new population.
8. Repeat step 5 until the size of the new chromosome

population becomes equal to the initial population

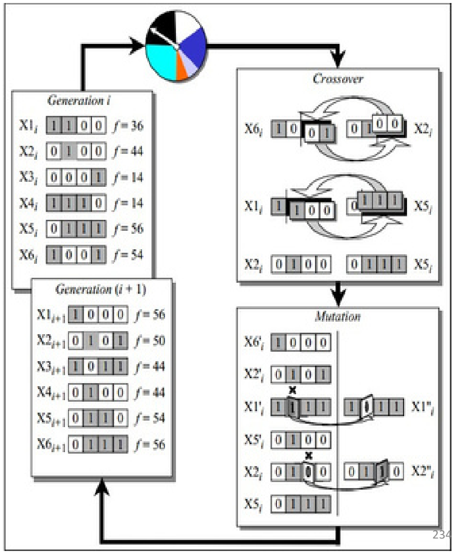
9. Replace the initial (parent) chromosome population with the new (offspring)population.
10. Go to step 4 and repeat until termination criterion

Genetic Algorithm Flow Chart



GA: Case Study

The Genetic Algorithm Cycle



GA : Case Study

Example of Selection

Evolutionary Algorithms is to maximize the function $f(x) = x^2$ with x in the integer interval $[0, 31]$, i.e., $x = 0, 1, \ldots 30, 31$.

1. The first step is encoding of chromosomes; use binary representation for integers; 5-bits are used to represent integers up to 31.
2. Assume that the population size is 4.
3. Generate initial population at random. They are chromosomes or genotypes; e.g., 01101, 11000, 01000, 10011.
4. Calculate fitness value for each individual.
   (a) Decode the individual into an integer (called phenotypes),
   $01101 \to 13$; $11000 \to 24$; $01000 \to 8$; $10011 \to 19$;
   (b) Evaluate the fitness according to $f(x) = x^2$,
   $13 \to 169$; $24 \to 576$; $8 \to 64$; $19 \to 361$.
5. Select parents (two individuals) for crossover based on their fitness in p_i. Out of many methods for selecting the best chromosomes, if **roulette-wheel** selection is used, then the probability of the i^{th} string in the population is $p_i = F_i \,/\, (\sum_{j=1}^{n} F_j)$, where
   F_i is fitness for the string i in the population, expressed as $f(x)$
   p_i is probability of the string i being selected,
   n is no of individuals in the population, is population size, $n=4$
   $n * p_i$ is expected count

GA : Case Study

String No	Initial Population	X value	Fitness Fi $f(x) = x^2$	p i	Expected count N * Prob i
1	0 1 1 0 1	13	169	0.14	0.58
2	1 1 0 0 0	24	576	0.49	1.97
3	0 1 0 0 0	8	64	0.06	0.22
4	1 0 0 1 1	19	361	0.31	1.23
Sum			1170	1.00	4.00
Average			293	0.25	1.00
Max			576	0.49	1.97

The string no 2 has maximum chance of selection.

6. Produce a new generation of solutions by picking from the existing pool of solutions with a preference for solutions which are better suited than others:
   We divide the range into four bins, sized according to the relative fitness of the solutions which they represent.

Strings	Prob i	Associated Bin
0 1 1 0 1	0.14	0.0 ... 0.14
1 1 0 0 0	0.49	0.14 ... 0.63
0 1 0 0 0	0.06	0.63 ... 0.69
1 0 0 1 1	0.31	0.69 ... 1.00

By generating 4 uniform (0, 1) random values and seeing which bin they fall into we pick the four strings that will form the basis for the next generation.

Random No	Falls into bin	Chosen string
0.08	0.0 ... 0.14	0 1 1 0 1
0.24	0.14 ... 0.63	1 1 0 0 0
0.52	0.14 ... 0.63	1 1 0 0 0
0.87	0.69 ... 1.00	1 0 0 1 1

GA : Case Study

7. Randomly pair the members of the new generation

Random number generator decides for us to mate the first two strings together and the second two strings together.

8. Within each pair swap parts of the members solutions to create offspring which are a mixture of the parents :

For the first pair of strings: 01101 , 11000

- We randomly select the crossover point to be after the fourth digit. Crossing these two strings at that point yields:

01101 ⇒ 0110|1 ⇒ 01100

11000 ⇒ 1100|0 ⇒ 11001

For the second pair of strings: 11000 , 10011

- We randomly select the crossover point to be after the second digit. Crossing these two strings at that point yields:

11000 ⇒ 11|000 ⇒ 11011

10011 ⇒ 10|011 ⇒ 10000

GA : Case Study

9. Randomly mutate a very small fraction of genes in the population :

With a typical mutation probability of per bit it happens that none of the bits in our population are mutated.

10. Go back and re-evaluate fitness of the population (new generation) :

This would be the first step in generating a new generation of solutions. However it is also useful in showing the way that a single iteration of the genetic algorithm has improved this sample.

String No	Initial Population (chromosome)	X value (Pheno types)	Fitness $f(x) = x^2$	Prob i (fraction of total)	Expected count
1	01100	12	144	0.082	0.328
2	11001	25	625	0.356	1.424
3	11011	27	729	0.415	1.660
4	10000	16	256	0.145	0.580
Total (sum)			1754	1.000	4.000
Average			439	0.250	1.000
Max			729	0.415	1.660

Observe that :

1. Initial populations : At start step 5 were

   01101, 11000, 01000 , 10011

   After one cycle, new populations, at step 10 to act as initial population

   01100, 11001, 11011 , 10000

2. The total fitness has gone from **1170** to **1754** in a single generation.
3. The algorithm has already come up with the string 11011 (i.e **x = 27**) as a possible solution.

Genetic Algorithm Applications

Genetic algorithms are used to solve many large problems including:

-Scheduling

- Transportation

-Chemistry, Chemical Engineering

-Layout and circuit design

-Medicine

-Data Mining and Data Analysis

-Economics and Finance

-Networking and Communication

-Game etc.

Genetic Algorithm Advantages and Disadvantages

Advantages:

- It can solve every optimization problem which can be described with the chromosome encoding.
- It solves problems with multiple solutions.
- Since the genetic algorithm execution technique is not dependent on the error surface, we can solve multi-dimensional, non-differential, non-continuous, and even non-parametrical problems.
- Structural genetic algorithm gives us the possibility to solve the solution structure and solution parameter problems at the same time by means of genetic algorithm.
- Genetic algorithm is a method which is very easy to understand and it practically does not demand the knowledge of mathematics.
- Genetic algorithms are easily transferred to existing simulations and model

Disadvantages

- May be Slow
- May be drop of the quality because of crossover.