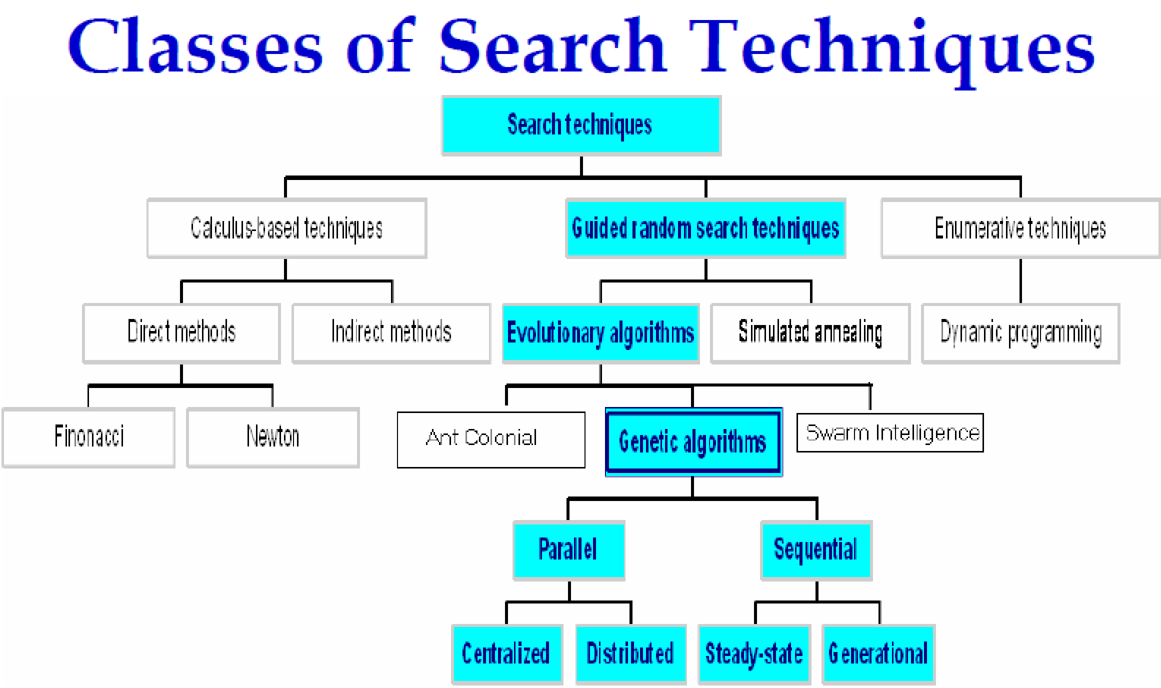
MODULE-III (10 HOURS)

Derivative-free Optimization Genetic algorithms: Basic concepts, encoding, fitness function, reproduction. Differences of GA and traditional optimization methods. Basic genetic programming concepts Applications.

LECTURE-1

Introduction:

Most real world optimization problems involve complexities like discrete, continuous or mixed variables, multiple conflicting objectives, non-linearity, discontinuity and non-convex region. The search space (design space) may be so large that global optimum cannot be found in a reasonable time. The existing linear or nonlinear methods may not be efficient or computationally inexpensive for solving such problems. Various stochastic search methods like simulated annealing, evolutionary algorithms (EA) or hill climbing can be used in such situations. EAs have the advantage of being applicable to any combination of complexities (multi-objective, non-linearity etc) and also can be combined with any existing local search or other methods. Various techniques which make use of EA approach are Genetic Algorithms (GA), evolutionary programming, evolution strategy, learning classifier system etc.



The principle of Darwinian evolution theory i.e., *survival of the fittest is evaluated by a fitness function* derived from objective function. Every individual in a population searches to be the best according to a fitness function in its own way(randomly).

Basic Concepts:

Optimization means to make the objective function max or min. That means in evolutionary computing where the individuals/ elements represent possible solutions, an element exists such that the fitness of that element is the maximum or minimum among all others' fitness depending on it is maximization or minimization problem.

Optimization can be classified as:

1. **Deterministic**-Uses derivative or gradient to reach final solution

2. **Stochastic**- Derivative free optimization, a type of random search, suitable for non-linearity, discontinuity escape from local optima and non-convex region

Components of Genetic Algorithm:

The individuals are genes which encode a trait or a parameter. The design space is to be converted to genetic space. It is parallel processing by a population used when single point approach of traditional methods cannot find a possible solution with in the required time frame.

Important common aspects of evolutionary/swarm optimization algorithms: It is an iterative process where best solution is searched by a population in search space evaluating a fitness function.

1. **Search space**-Space for all feasible solutions is called search space.
2. **Solution-** It is the point with maximum or minimum value of fitness function.
3. **Fitness function**- A function derived from objective function
4. **Population size**- A number of points in a search space used in parallel for computing is called population, generally ranging from 30 to 200.
5. **Constraints**- Lower and upper bounds
6. **Stopping criteria**- it can be no. of iterations, or minimum value of error in fitness or minimum improvement from previous iteration

LECTURE-2

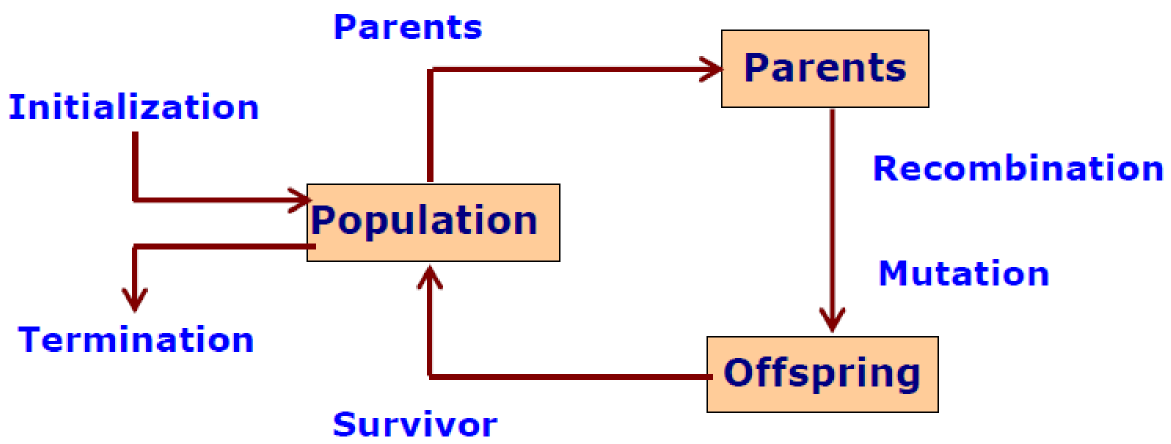


Fig. Basic cycle of EA

Basic flow chart of EA:

The initial population is usually generated randomly in all EAs. The termination condition may be a desired fitness function, maximum number of generations etc. In selection, individuals with better fitness functions from generation '*i*' are taken to generate individuals of '*i+1*'th generation. New population (*offspring*) is created by applying *recombination* and *mutation* to the selected individuals (*parents*). Recombination creates one or two new individuals by swaping (crossing over) the genome of a *parent* with another. Recombined individual is then *mutated* by changing a single element (genome) to create a new individual.

Finally, the new population is evaluated and the process is repeated. Each step is described in more detail below with special reference to GA. GA was proposed by Davis E. Goldberg.

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

Psuedocode of Genetics Algorithm

- Choose the initial population of individuals
- Evaluate the fitness of each individual in population
- Repeat until termination condition satisfied:
  - Selection: Select the individuals with greater fitness for reproduction
  - Crossover: Breed new individuals through crossover
  - Mutation: Apply probabilistic mutation on new individuals
  - Form a new population with these offsprings.
- Terminate

LECTURE-3

Special features of GA:

Encoding-

Objects forming possible solution sets to the original problem is called *phenotype* and the encoding (representation) of the individuals in the EA is called *genotype*. In GA each possible solution is coded in to genetic space. The coding may be binary coding, real coding, hexadecimal coding, value coding and tree coding.

<u>Binary coding:</u>

If each design variable is given a string of length '*l*', and there are *n* such variables, then the design vector will have a total string length of *'nl'*. For example, let there are 3 design variables and the string length be 4 (not necessarily fixed for all problems, depends on accuracy in representing variable)for each variable. The variables are x1=4,x2=7 & x3=1. Then the *chromosome* length is 12, where 4 bit in binary representing x1=0100 , x2=0111, x3=0001 are genes. So each string/ chromosome represents a different solution.

An individual consists a genotype and a fitness function. *Fitness* represents the quality of the solution (normally called *fitness function*). It forms the basis for selecting the individuals and thereby facilitates improvements.

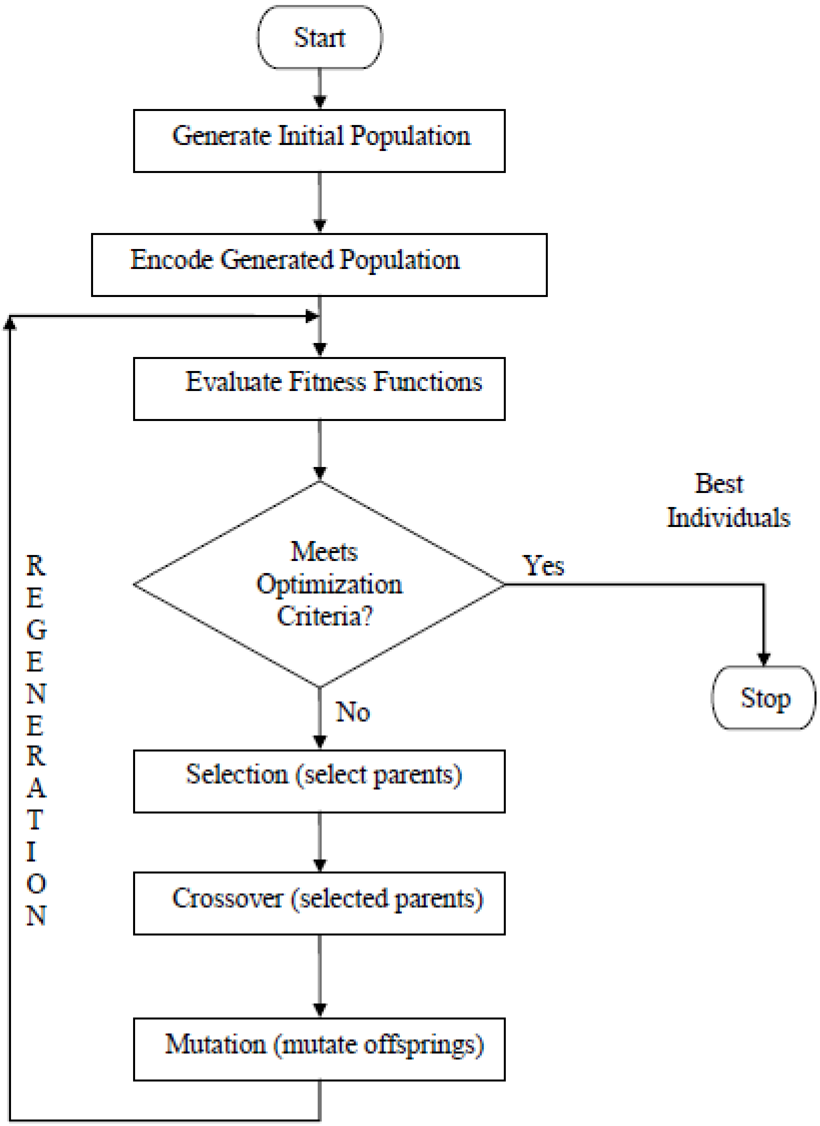


Fig. Flow chart of GA

Decoding:
If xi^L & xi^U correspond to 0000 to 0111 ; ni the bit length of coding decoded value xi will be

$$Xi = X_i^L + \frac{(X_i^U - X_i^L)}{(2^{ni} - 1)} \times \text{(decoded value of string)}$$

LECTURE-4

PARENT SELECTION:

After fitness function evaluation, individuals are distinguished based on their quality. According to Darwin's evolution theory the best ones should survive and create new offspring for the next generation. There are many methods to select the best chromosomes.

1. Roulette wheel selection
2. Boltzmann selection
3. Tournament selection
4. Rank selection
5. Steady state selection

The first one is briefly described.

<u>Roulette Wheel Selection:</u> Parents are selected according to their fitness i.e., each individual is selected with a probability proportional to its fitness value. In other words, depending on the percentage contribution to the total population fitness, string is selected for mating to form the next generation. This way, weak solutions are eliminated and strong solutions survive to form the next generation. For example, consider a population containing four strings shown in the Table 1. Each string is formed by concatenating four substrings which represents variables a,b,c and d. Length of each string is taken as four bits. The first column represents the possible solution in binary form. The second column gives the fitness values of the decoded strings. The third column gives the percentage contribution of each string to the total fitness of the population. Then by "Roulette Wheel" method, the probability of candidate 1 being selected as a parent of the next generation is 28.09%. Similarly, the probability that the candidates 2, 3, 4 will be chosen for the next generation are 19.59, 12.89 and 39.43 respectively. These probabilities are represented on a pie chart, and then four numbers are randomly generated between 1 and 100. Then, the likeliness that the numbers generated would fall in the region of candidate 2 might be once, whereas for candidate 4 it might be twice and candidate 1 more than once and for candidate 3 it may not fall at all. Thus, the strings are chosen to form the parents of the next generation.

Table 1

Candidate	Fitness value	Percentage of total fitness
1011 0110 1101 1001	109	28.09
0101 0011 1110 1101	76	19.59
0001 0001 1111 1011	50	12.89
1011 1111 1011 1100	153	39.43
Total	388	100

The one with higher probability p_i calculated is judged as per

$$p_i = F_i \Big/ \left(\sum_{j=1}^{n} F_j \right)$$

where Fi the fitness and n is the number of chromosome.

Choice of Methods of selecting chromosomes to be parents depends on

1. <u>Population diversity</u>- good ones exploited but new areas are explored but slow
2. <u>Selective pressure</u>-better individuals are preferred but chance of getting local max/min but fast

Generation:
Population of design vector obtained after one computation on all individuals.
Generation gap:
Proportion of individuals in the population that are replaced in each generation. It should be increased to improve results.
Reproduction:
Generally after choosing a method of selection population with above average fitness function in one generation are taken as parents and inserted in next population in multiple(clones) so that good trait is transferred to child.

LECTURE-5

CROSSOVER:
It is a recombination operator. Selection alone cannot introduce any new individuals into the population, i.e., it cannot find new points in the search space. These are generated by genetically-inspired operators, of which the most well known are *crossover* and *mutation*.
Types-
1. One-point
2. Two-point
3. Uniform
4. Arithmetic
5. Heuristic
6. Matrix

In one point crossover, selected pair of strings is cut at some random position and their segments are swapped to form new pair of strings.

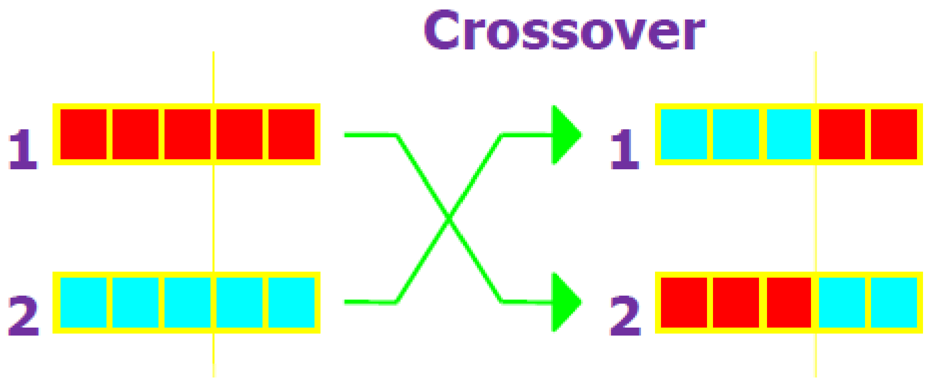


Fig. One point crossover

In two-point scheme, there will be two break points in the strings that are randomly chosen. At the break-point, the segments of the two strings are swapped so that new set of strings are formed. For example, let us consider two 8-bit strings given by '10011101' and '10101011'. Then according to one-point crossover, if a random crossover point is chosen after 3 bits from left and segments are cut as shown below: 100 | 11101 101 | 01011 and the segments are swapped to form 10001011 10111101 According to two-point crossover, if two crossover points are selected as 100 | 11 | 101 101 | 01 | 011 Then after swapping both the extreme segments, the resulting strings formed are 10001101 10111011 Crossover is not usually applied to all pairs of individuals selected for mating. A random choice is made, where the probability of crossover P_c being applied is typically between 0.6 and 0.9.
MUTATION:
Mutation is applied to each child individually after crossover. This creates diversity. It randomly alters each gene with a small probability P_m(generally not greater than 0.01).

Types-
1. Flip-bit
2. Boundary
3. Uniform
4. Non- Uniform
5. Gaussian

It injects a new genetic character into the chromosome by changing at random a bit in a string depending on the probability of mutation. Example: 10111**0**11 is mutated as 10111**1**11 It is seen in the above example that the sixth bit '0' is changed to '1'. Thus, in mutation process, bits are changed from '1' to '0' or '0' to '1' at the randomly chosen position of randomly selected strings. The mutation may be for one individual or a group.

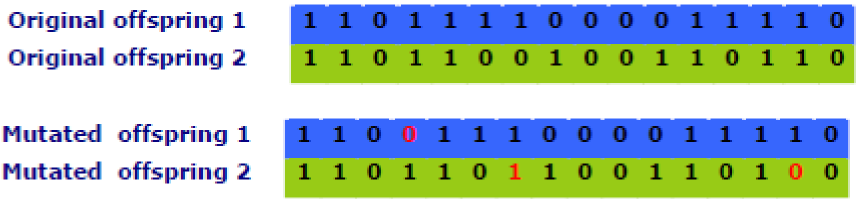


Fig. Mutation bits shown in red

LECTURE-6

ADVANTAGES AND DISADVANTAGES OF EA:

EA can be efficiently used for highly complex problems with multi-objectivity, non-linearity etc. It provides not only a single best solution, but the 2nd best, 3rd best and so on as required. It gives quick approximate solutions. EA methods can very well incorporate with other local search algorithms. There are some drawbacks also in using EA techniques. An optimal solution cannot be ensured on using EA methods, which are usually known as heuristic search methods. Convergence of EA techniques are problem oriented. Sensitivity analysis should be carried out to find out the range in which the model is efficient. Also, the implementation of these techniques requires good programming skill.

Differences and similarities between GA and other traditional methods:

Differences:
1. GA uses coding which discretizes search space even though function is continuous
2. A discrete function can be handled with no extra cost
3. Works with a population of points instead of single, so multiple optimal solutions can be captured at a time, reducing no. of run of algorithm

Similarities:
1. Search direction in traditional algorithms is used to find new point, where as 2 points are used to define search direction a in crossover in GA
2. Search direction is not fixed for all points, as mutation works in GA

New Variants of GA:

According to the encoding, selection, crossover or mutation methods and adaptive changing of the probabilities with convergence many variants like continuous GA, binary GA, RCGA, NSGA etc have been developed. In addition the new ones are

1. Messy GA
2. Parallel GA
3. Multiobjective GA

LECTURE-7

Issues for GA Parameter settings:

Choosing basic implementation issues:

1. Representation
2. Population size, mutation rate, ...
3. Selection, deletion policies
4. Crossover, mutation operators
5. Termination Criteria
6. Performance, scalability
7. Solution is only as good as the evaluation function (often hardest part)

Benefits of Genetic Agorithms

1. Concept is easy to understand
2. Modular, separate from application
3. Supports multi-objective optimization
4. Good for "noisy"environments
5. Always gives answer; answer gets better with time
6. Inherently parallel; easily distributed
7. Multiple ways to speed up and improve a GA-based application as knowledge about problem domain is gained
8. Easy to exploit previous or alternate solutions
9. Flexible building blocks for hybrid applications
10. Substantial history and range of use

Shortcomings of GA:

1. **Minimal deception problem-** Some objective functions may be very difficult to optimize by GA. R**epresenting the solution accuracy depends on coding.**
2. **GA drift(Bias)-** Loss of population diversity may seek suboptimal solution with a smaller population size
3. Real time& online issues- It does not guarantee response time, which is vital in real time issues. Works offline satisfactorily.
4. Computationally expensive and time consuming
5. Issues in representation of problem
6. Proper writing of fittness function
7. Proper values of size of population, crossover and mutation rate
8. Premature Convergence
9. No one mathematically perfect solution since problems of biological adaptation don't have this issue

LECTURE-8 to 10

Applications of Genetic Algorithms:

1. <u>Optimization and design</u>

numerical optimization, circuit design, airplane design, factory scheduling, drug design, network optimization

2. Automatic programming
   evolving computer programs (e.g., for image processing), evolving cellular automata
3. Machine learning and adaptive control
   robot navigation, evolution of rules for solving “expert”problems, evolution of neural networks, adaptive computer security, adaptive user interfaces

4. Complex data analysis and time-series prediction
   prediction of chaotic systems, financial-market prediction, protein-structure prediction
5. Scientific models of complex systems
   economics, immunology, ecology, population genetics, evolution, cancer

References:

1. Chapter 7 J.S.R.Jang, C.T.Sun and E.Mizutani, “*Neuro-Fuzzy and Soft Computing”*, PHI, 2004, Pearson Education 2004.
2. Chapter 8 & 9 S. Rajasekaran & GA Vijayalakshmi Pai “Neural Networks, Fuzzy Logic, and Genetic Algorithms synthesis and application”, PHI
3. Internet sources