#2018UIC3116#

Date / /
Page No.

Que 3 Write the detailed notes on GA and PSO.

→ Genetic Algorithm :

> Introduced by prof. John Holland in 1975.

> Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection & crossover (also called recombination)

> Genetic algorithms are categorized as global search heuristic algorithm.

Fitness function F(x) : ✱ evaluates how close a given solution is to optimization solution of problem.

> derived from objective function f(x)

> Most often used function is

$$f(x) = 1/(1+f(x))$$

GA operators :

① Reproduction ② Crossover ③ Mutation

Reproduction:

> It selects good (above average) strings in a population and forms a mutation pool.

> The probability for selecting i^{th} string is

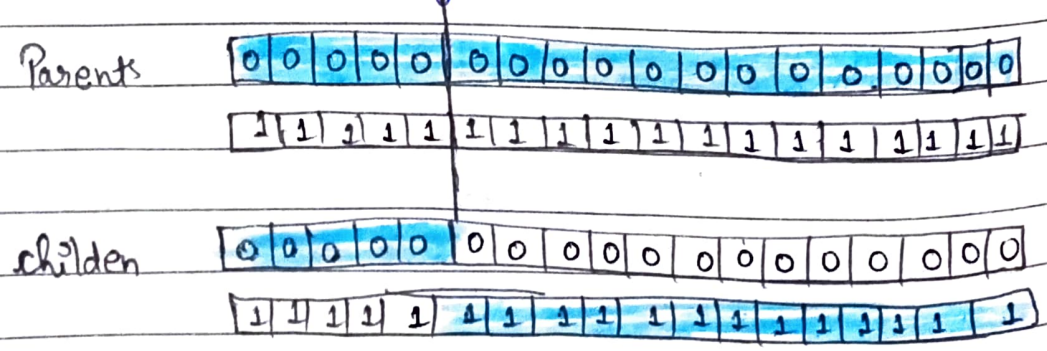
$$P_i = \frac{F_i}{\sum_{j=1}^{n} F_j} \quad ; i = 1, 2, \ldots n$$

n = population size

> One way to implement the selection scheme is roulette-wheel selection mechanism.

Crossover:

> Choose a random point on the two parents
> Split parents at this crossover point
> Create children by exchanging tails
> Crossover probability is 'pc'.



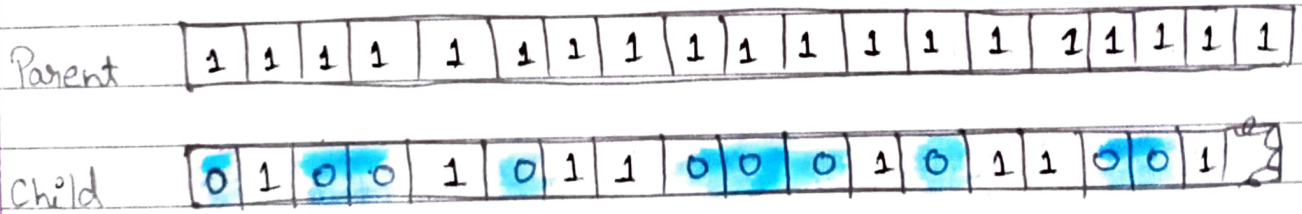
Mutation :

> The mutation operator is applied to the new strings with a specific small mutation probability, p_m.

> The mutation operator changes the binary digit 1 to 0 and vice versa.

> p_m is called the mutation rate
  - Typically between 1/pop_size and 1/chromosome_length

> It maintains diversity in the population



Steps in GA :

1. Choose a coding to represent problem parameters, a selection operator, a crossover operator, mutation operator, population size, crossover probability and mutation probability.

2. Initialize random population of strings of size l, t_{max} set $t = 0$.

3. Evaluate each string in population

4. If $t > t_{max}$ or other termination criteria is satisfied, terminate.

5. perform reproduction on the population.

6. perform Crossover on random pairs of string

7. perform mutation on every string

8. Evaluate strings in the new population. set $t = t+1$ and go to step 3.

Advantages:

> It is very potential algorithm.

> used for complex engineering problems.

> A population of points is used for starting the procedure instead of a single design point.

> GAs use only the values of the objective function. The derivates are not used in the search procedure.

Date / /
Page No.

- The objective function value corresponding to a design vector plays the role of fitness in natural genetics.
- GA efficiently explore the new combinations with the available knowledge to find a new generation with better fitness value.

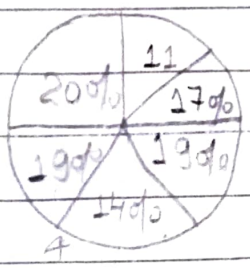
Genetic operators: Reproduction

There are many techniques for Reproduction operators

1. Roulette wheel selection
2. Tournament selection
3. Ranked position selection
4. Steady state selection
5. Proportionate selection

1. Roulette wheel selection:

The circumference of Roulette wheel is divided into segments and marked for each string proportionate to the fitness value.



- The wheel is spun n times; each time a segment is selected by the wheel pointer.
- The segment with highest fitness value will have more probability for selection.

Roulette wheel selection Contd..

- The individual fitness values are mapped to contiguous segments of a line.
- using fitness values of the string selection probability is computated using

$$P_i = F_i' / \Sigma F_j \quad j = 1, 2, \ldots n$$

- Cumulative probability for each string is computed: the i^{th} string in the population represents the cumulative probability from P_{i-1} to P_i.
- A random number is generated and the string for which the random number lies in the cumulative probability range is selected for the mating pool.
- The process is repeated until the desired number of individuals (called mating population)

is obtained.

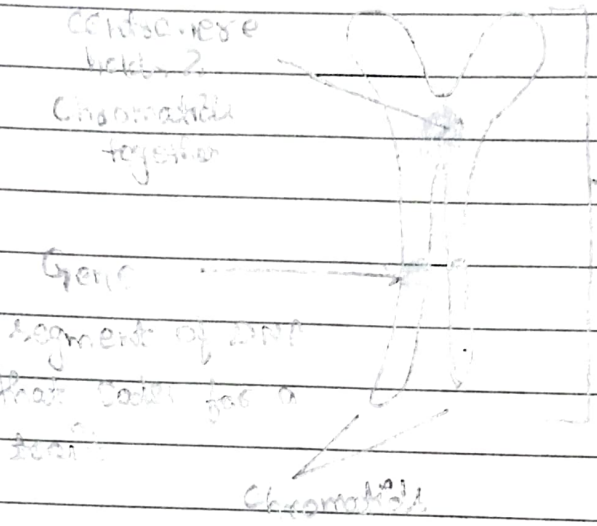
2. Tournament Selection:

- In tournament selection several tournaments are played among a few individuals. The individuals are chosen at random from the population.
- The winner of each tournament is selected for next generation.
- Selection pressure can be adjusted by changing the tournament size.
- Weak individuals have a smaller chance to be selected if tournament size is large.

How to implement Crossover?

The crossover operator is used to create new solutions from the existing solutions available in the mating pool after applying selection operator.

This operator exchanges the gene information between the solutions in the mating pool.



Encoding of solution is necessary so that our solutions looks like a chromosome.

→ Encoding?

The process of representing a solution in the form of a string that conveys the necessary information.

Date / /
Page No.

Just as in chromosome, each gene controls a particular characteristic of the individual similarly, each bit in the string represents a characteristic of the solution.

Encoding methods:

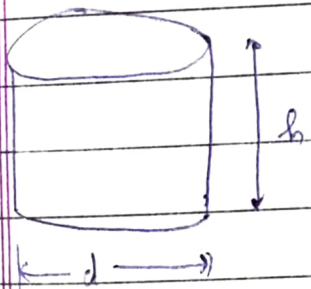
minimize $f(d,h) = c\left((\pi d^2/2) + \pi dh\right)$

subject to $g_1\{d,h\} \equiv \{\pi d^2 h/4\} \geq 300$

Variable bounds $d_{min} \leq d \leq d_{max}$

$h_{min} \leq h \leq h_{max}$

defining a string $\underbrace{[0100}_{d}\underbrace{01010]}_{h}$



$(d,h) = \{8,10\}$ cm

Chromosome = [0100001010]

Crossover methods :

1. Real Valued crossover :

The following methods are used to get a new offspring from two parent strings

1.1 Discreate Crossover
1.2 Intermediate Crossover
1.3 Line crossover

2. Binary Valued Crossover :

The following methods are used to get a new offspring from two binary encoded strings of parents selected from mating pool.

2.1 Single-point crossover
2.2 Multi-point crossover
2.3 Uniform Crossover
2.4 Shuffle Crossover

Date / /
Page No.

Mutation methods :

1. Binary Valued Mutation

- This is widely used encoding mechanism.
- In this real value of the variables are transformed to binary codes and the genetic operator work on these coded strings.

The bit-wise operators are as below

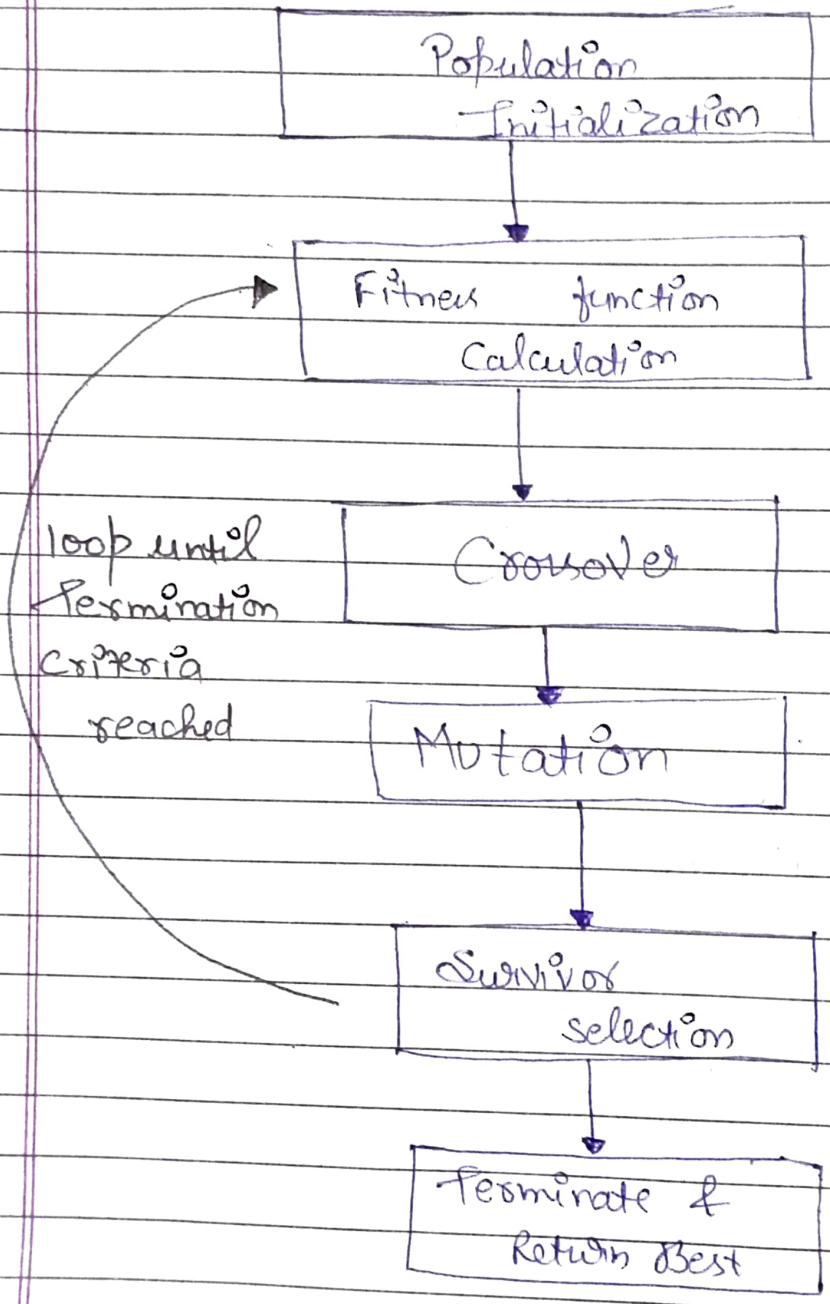
1.1 one's complement operator
1.2 Logical bitwise operator
1.3 shift operator
1.4 swap operator
1.5 inverse operator

2. Real Valued Mutation

Date / /
Page No.

Basic Structure:

The Basic structure of GA is:



Date / /
Page No.

Particle Swarm optimization:

PSO is an technique that can be used to find approximate solutions to extremely difficult or impossible numeric maximization and minimization problems.

How it works?

- PSO is initialized with a group of random particles and then searches for optimal by updating generations.
- particles move through the solution space, and are evaluated according to some fitness criterion after each time step. In every iteration, each particle is updated by following two "best values".
- The first one is the best solution (fitness) it has achieved so far, (the fitness value is also stored). This value is called pbest.
- Another "best value" that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the population. The second best value is a global best and called gbest.

PSO Algorithm: (General)

Searches Hyperspace of problem for optimum

- Define problem to search
- How many dimensions?
- Solution Criteria?
- Initialize population
- Random initial positions
- Random initial velocities
- Determine Best position
- Global Best position
- personal Best position
- update velocity & position equations

Implementation:

Step: 01

- Initialize PSO parameters which are necessary for the algorithm.
- population size which indicates the number of individuals,
- number of generations necessary for the termination criterion.
- Cognitive Constant, social constant,

- Variation of inertia weight, maximum velocity.
- Number of design variables are respective ranges for the design variables.

Step: 02

- Generate random population equal to the population size specified.
- Each population member contains the value of all the design variables. This value of design variables is randomly generated in between the design variable range specified.
- Population means the group of birds (particles) which represents the set of solutions.

Step: 03

- Obtain the values of the objective function for all the population members.
- For the first interation, value of objective function indicates the pBest for respective particle in the solution.
- If the problem is constrained optimization problem, then a specific approach such as static penalty, dynamic penalty &

adaptive penalty is used to convert the constrained optimization problem into the constrained optimization problem.

Step : 04

- Update the velocity of each particle & check for the maximum velocity.
- If the velocity obtained exceeds the maximum velocity.
- then reduce the existing velocity to the maximum velocity.

Step : 05

- Update the position of the particles.
- Check all the design variables for the upper and lower limits.

Step : 06

- Obtain the value of objective function for all the particles.
- The new solution replaces the pBest if it has better function value.

- Identify the gBest from the population
- Update the Value of interia weight if required.

Step : 07

- Best obtained results are saved using elitism.
- All elite members are not modified using crossover & mutation operators can be replaced if better solutions are obtained in any iteration.

Step : 08

- Repeat all steps (from step 4) until the specified number of generations or termination criterion is reached.

Advantages :

- Pso is based on the intelligence. it can be applied into both scientific research & engineering use.
- Pso have no overlapping and mutation calculation.

- The search can be carried out by the speed of particle. During the development of several generations, only the most optimist particle can transmit information onto the other particles and the speed of the researching is very fast.
- PSO adopts the real number code, & it is decided directly by the solution.

Algorithm

- The PSO algorithm may be written in pseudo code as follows.
- Algorithm parameters

A : Population of events.
Pi : position of agent ai in solution space.
f : objective function.
vi : velocity of agent ai.

V(ai) : Neighborhood of agent ai (fixed)

[x*] = PSO()
P = particle — Initialization();
For i = 1 to it_max
for each particle P in P do

Date / /
Page No.

```
fp = f(p);
  if fp better than f(Pbest)
    Pbest = P;
  end
end
  gbest = best P in P;
  for each particle P in P do
    Vi = Vi + c1 * rand() * (Pbest - P) + c2 * rand() *
                                          (gbest - P);
    P = P + V;
  end
  end algorithm
```

Algorithm Contd...

1. First we create a population of agents or particles to make a swarm, uniformly distributed over a space X.

2. Evaluate the position of each particle according to objective function.

3. If a particle's current position is better than its initial position, then update it.

4. determine best particle (acc^n to particle's

previous best position).

5. Update particles position

$$V_i^{t+1} = \underbrace{V_i^t}_{\text{inertia}} + \underbrace{C_1 U_1^t (Pb_i^t - P_i^t)}_{\text{personal influence}} + \underbrace{C_2 U_2^t (g_b^t - P_i^t)}_{\text{social influence}}$$

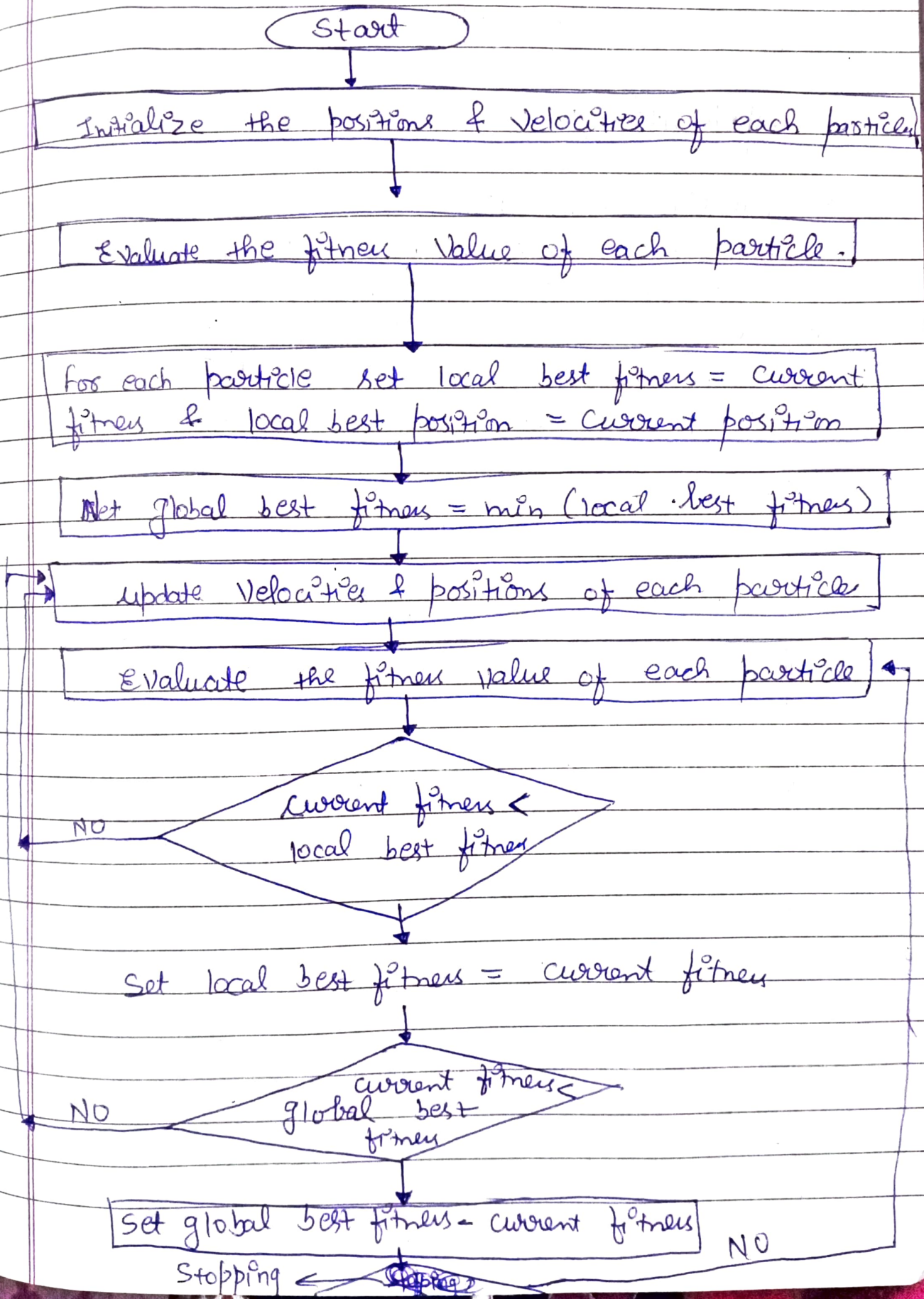
6. Move particles to new positions.

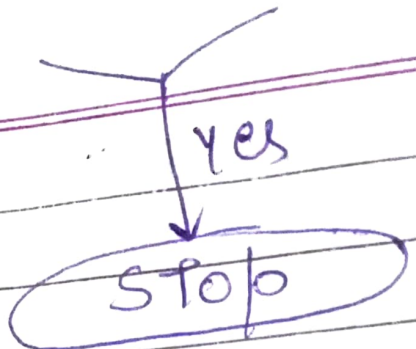
$$P_i^{t+1} = P_i^t + V_i^{t+1}$$

7. Goto ~~set~~ step 2 until stopping criteria are met.

Date / /
Page No.

Flowchart





Applications:

- Function optimization
- Artificial neural network training
- Identification of parkinson's disease
- Extraction of rules from fuzzy networks
- Image recognition
- Areas where GA can be applied.