Limitation it is a greedy approach.
even though it has a very fast convergence rate
but it may reach the local optimal solution
& trap there.

1 What is population?
2 How will you define convergence rate of an algorithm?
3 in what kind of problems we use sharing fitness approach?
4 what will happen if we remove (best − x^i) from the velocity equation
5 What is Roulette wheel selection operator in GA state with an example.
6 What equations or operators are used in GA for exploiting & exploring the search space
7 What will happen if we apply sharing fitness approach in PSO?
8 What is a chromosome? genetic representation of it.

ML problem identify the best fit function
hard :- because we can apply many methods & have many functions.
prediction problem decide the i/p problems its up to you to correctly relate the i/p choose the relevant parameters — ML problem

15 Oct '19 Genetic Programming

It is a randomized algorithm used to design function for a given problem.

optimizing at functional level (not at solution level (GA)).

problems are of 2 types

↳ P problem → algorithms for which deterministic solution can be designed are polynomial problems it has to process all the inputs.

↳ NP problems → which cannot be solved in polynomial time using traditional algorithms

eg - optimization problem → max or min

② all problems for which we apply ML are NP hard problems like association, clustering, function prediction etc.

Function prediction algorithm →

x	f(x)
1	1
2	4
3	9
4	16
5	?

$f(x) = x^2$ or $x^2 + (x-1)(x-2)(x-3)(x-4)$

There might be infinite number of solutions (functions) for a given table.

therefore it is Hard problem.

Applications → where we can earn something.

Used in weather forecasting, disaster management

x	f(x)
1	2
2	4
3	9
4	16
5	25

① $x^2 - x + 1$

② $2x^2 - 3x + 4$

③ $x^3 - x^2 + x + 1$

④ $x^2 + 2x + 1$

to define the optimal function, we need to define optimality. we can guess these functions or will use existing

① linear regression
② multi regression (curve fitting)

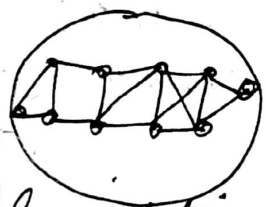
③ new method to make function is neural network

2 types of model
Mathematical model → always give some mathematical function

Biological model → in this model, this function is inherited in the shape of the model; will never give you exact model or relationship.
eg neural n/w is a biological model. here we used neurons, train the model & you will never know the working of this model; once you suffply i/p you will get output.

we use computer for problem solving.
2 types of problem →
① problem requiring computation (+, -, concat strings)
② face recognition

processing capability of computer & human is different
computer do it process by process, step by step
while brains have several neurons & they are interconnected



hardware implementation of neural network is not yet produced.
This is simulated in software & is known as artificial neural network (ANN)

18 Oct

Application of genetic algo in ML

Input Iris data set ↓ play Tennis

outlook	temprature	Humidity	wind	Play Tennis
Sunny	High (Hot)	High	weak	No
Sunny	hot	high	strong	No
cloudy	cool	normal	weak	Yes
Rainy	hot	Normal	Weak	Yes
cloudy	mild	High	strong	No

PT = f (O, T, H, W) how to find f?

fitness function over this is a hypothesis which will satisfy the maximum number of cases.

Training data set → we are going to form function from this

Binary GA → OTHW output

how to apply binary GA? represent i/p as binary

outlook can be represented as < sunny / cloudy / rainy (SCR)

so 3 bits

Temp < hot (HMC) / mild / cool

Humidity < high (HN) / normal

wind < strong (WS) / weak (WW)

Play Tennis < No (NY) / Yes

if we say outlook is sunny then what?

SCR HMC HN SW N

1 0 0 1 1 1 1 1 1 1 ~~0~~ 0 $= \frac{2}{5}$

in this case ignore all other variables & they will be in don't care condition

$\frac{2}{5}$ bcoz out of 5 cases it is satisfying 2 values at of 5.

110 001 0 1201
SC C N S
Sunny cool Normal strong
or cloudy

$= \frac{0}{5}$

outlook
sunny 100
don't care 111

010 010 10 100
100 001 01 011

⚠ always mention the sequence of coding otherwise invalid

Note:- output of data have 1 bit only i.e No don't care so in this data ~~only~~ fitness will be $\frac{3}{5}$ or $\frac{2}{5}$ or $\frac{0}{5}$

We want to find the max. fitness or max. data is to be satisfied ∴ we use binary GA. This binary chromosome is initial population

Limitation of GA is that it cannot create a generalised function

5th Nov 2019

another variant of GA.

(encoding) → Real parameter of GA → float GA
→ quantum parameter of GA

$f(x) = \frac{x^2}{1+x^2}$ $0 \le x \le 5$
max.

in binary GA partition the range in 2^n values but it cannot cover the entire search space

Quantum GA → random 0-5

2 3 1 4

$x_1 = 2.314$
$x_2 = 3.124$

in float GA → 0.0 to 5.0

use real parameter in problem which belong to continious search space. while binary is used where

$P_1 = 1101$ advantage of real no. GA is we have step of
$P_2 = 0011$ conversion from binary to real by GA.
When we change the encoding we have to change parameters as well.
Here crossover & mutation cannot be used for generating new solutions.

$x_1 = 2.314$
$x_2 = 3.124$

crossover here is $\frac{x_1 - x_2}{2}$ & $\frac{x_1 + x_2}{2}$

here we are using those operators that can be applied to arithmetic no.s → generate 2 new solutions

$f(x_1, x_2) = x_1^2 + x_2^2$
max.

$0 \le x_1 \le 5$
$0 \le x_2 \le 6$

initial population → (4) →
(2.1, 2.2)
(3.6, 4.3)
(1.2, 2.4)
(0.7, 5.7)

we can apply real parameter GA as search space is continuous

~~2.1, 2.2~~
3.6, 4.3
2.1, 2.2
Crossover
2.1, 4.3
3.6, 2.2

This problem is a 2-dimensional problem. GA is used to solve N-dimensional problem not only one or two dimension variable eg- big data problems, satellite communication (noise very high in this case)

mutation → subtract from upper limit
5 - 2.1
5 - 3.6

2. problems where binary GA is applicable one belongs to software engineering and another belongs to cloud computing.

in S/w engineering, test case generation

software is a product. process of software is

Resource gathering → Design → Code → testing → Maintainance

Problem with n dimensional problem
computational complexity

to solve n-dimensional problem reduce the dimension & build a model.

we are reducing dimensions because of our constraints
- hardware & software

how to apply mutation
in binary, flipping i.e. complementing
operators are user defined
complement with 1 0

Mutations is used for exploration diversity in solution space.

exploitation
- crossover } both used for exploration & exploitation
- mutation }

Test case generation problem
how s/w is different from a program
program -> educational method to solve a problem
software a product, related the market
some motivation behind it is satisfy the requirement of the client - profit is related.
most important & hard step in s/w engineering

hard because we have to check each & every i/p where code may fail.

very important test cases

test those test cases which can be used by the user.

create those test cases where product may fail

since we cannot generate all test cases we test those where client can test or product may fail.

eg :- buying a TV user can check Wifi in case of smart TV.

8 NOV '2019

① first check whether testing can be done in finite amount of time or not

eg -: check how many test cases are required whether we are producing correct output or not

```
int x, y, z;
z = x + y;
```

range of int is 2^{32}

$2^{32} \times 2^{32} = 2^{64}$

for a processor having speed 1 GHz

in 1 nano second $= 2^{30}$

2^{64} instructions may take one year

→ test cases are imp → that produce some fault

test case

good bad

→ company performs only those test cases that can be performed by the client

Types of Testing -:

① Black Box Testing

→ usually used in companies

→ everything is hidden from you.

you assume that you don't ~~know~~ know any details

→ since you know the correct output you check whether o/p produced is correct or not

Test oracle is a person or program (software) matching the output produced with actual & expected output.

output

correct incorrect

This test case is very important

eg x=3 y=3

o/p = 9 corr = 6

∴ some error

→ some times we create test cases on the boundary

boundary line testing

② White box Testing

→ tester knows everything about the program & the program can be converted into a flow graph.

→ used by academicians.

→ program consists of instructions.
sequential, arithmetic, looping

→ to check if a program is behaving correctly, we check if each & every path is behaving correctly.

3 criterias for white box testing **

① covered all the statements of the program or not
→ statement coverage

② covered all the branch statement of the program
→ branch coverage

③ covered all the paths or not
→ path coverage:

```
int x, y, z, d;
if (x > y)
    z = x + y;
else
    d = x - y;
```

perform white box testing
in graph simple assignment statement can be represented as a line.
" if " represented as a branch

flow graph for the above program is

control flow graph (CFG).

statements that can be grouped - will form a block.

Paths = 2 : if we cover both the path program is correct.

if a program passes all test cases - a single test case cannot cover all the paths.

eg - : first declaration is ignored
the if, else are conditions not statements.

eg for x, y as 2, 3 = $\frac{1}{2}$ = 50% fitness
↳ statements covered

(3, 2) = 50%
(1, 1) = 50%

identify those test cases which will cover the maximum program.

There are several tools which are used for testing.

↳ Gcov - g coverage ; linux based tool; helps in finding

Lcov line based covering; will also create a graph.

the above tools can help you in many things

eg:-
```
if (x > y)
{
    d = a - k;
    c = z - d;
    e = f - c;
}
else
    m = n + e;
```

test cases having $x > y$ will satisfy >50% of test cases

eg 3 -: if $(x > y \&\& (x > y + z))$
here it becomes complex

→ When the if's become nested, the program becomes more complex; the testing becomes more complex as the test domain decreases.

→ This process of testing is called as white box testing

suppose -:

Test Suite -: $\{T_1, T_2, \ldots T_n\}$

set/collection of test cases used to cover the entire program

Regression Testing -: used during maintainance

Maintainance - whenever the problem occurs it is corrected.

problem in software → the test suite will not pass

after maintainance, the portion not changed or the test cases from the test suite that are working - selecting those test cases is known as test case selection problem.

this is lg A problem

eg -:

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
0	1	0	1	0	1	0	0	0	0

binary chromosome ↑

running subset = $\{T_2, T_4, T_6\}$

→ we are interested in those test cases which cover max portion of the program

Mutation Testing -: used in very sensitive programs

eg: in a nuclear reaction; checking the position of the piston

programs are not so big in this case but

in a program, 2 things are important:
data
operation

we always check if we can change the data what will happen not checked for operation

eg:: 3,7 → P: x+y → exp=10, o/p=21

→ for (2,2) o/p=4 exp=4

so sensitivity of this testcase is very low

we create a mutant program by changing the operator such as:

x+y | x-y | x*y | x/y

- we calculate the mutancy score of the test cases. we need the test case to kill all the programs.
- we need a test case which gives diff values for all mutants; most sensitive testcases.

eg: 2,2 can kill x-y & x/y

cloud computing -:

→ like a n/w which provides services to the user

→ cloud model is a service model. if you are having resources, you can earn money like computers in PG lab are not used at night.

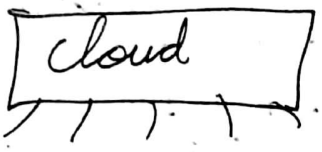
→ we have cloud as a infrastructure

→ there are several model

① infrastructure as a service

② software

③ platform



Cloud may be connected to many users, all users are requesting.

① to run an application, we need a work flow

- to execute a program, we need to divide the program in small steps.
- they will try to execute this workflow on the given infrastructure so that they run in the shortest time span

this is known as optimization problem

- This is known as work flow scheduling problem and solved using GA.

(II) how to allocate resources to the users so that his profit is maximized?

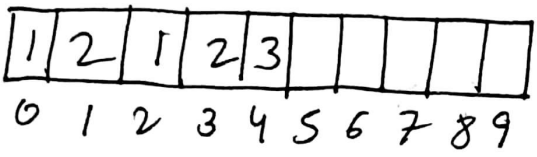
2nd optimization problem - solved using GA.

for solving using GA - first we randomly generate a solution.

→ indexes of the array represent ~~virtual machine~~ tasks.

eg -: to run 10 tasks, we select an array of size 10.

→ tasks are mapped to machine ~~indexes~~

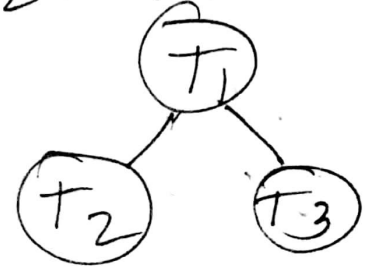


this is the work flow - which VM is assigned to which task Tasks are represented as indexes

- What is typical in this?

- these tasks are not independent - there is some dependency involved in the tasks

eg -: T_2 will run only after T_1



flowgraph.

PSO can be used but we need to redesign the operators - +, * cannot be used, for fitness, we calculate cost - executioncost - transfer cost.

execution cost :- executing data on a VM

	VM_1	VM_2	VM_3	...	VM_n
T_1	30₹/KB				
T_2					
T_3					

transfer cost :- cost of transferring data from one VM to another

	T_1	T_2	T_3	..	T_{10}
T_1					
T_2					
:					
T_{10}					

total fitness = cost + time taken

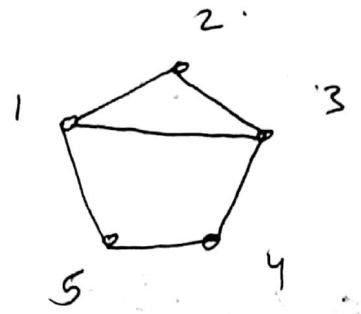
Note :- ① PSO cannot be used in whitebox / regression testing.
② Regression testing comes in white box testing
③ Real parameter by A is used in white box testing
④ function optimisation → genetic programming
⑤ which other problems belong to cloud?

15 Nov

genetic algo → Travelling Salesman Problem (TSP)

2^n permutations

NP hard problem



2^5 permutations in this case

GA → selection, crossover & Mutation

You can write new selection mechanism which suits your algorithm

like in TSP 12345 is the chromosome

13425

cutpoint crossover will not work as some cities will repeat

12|345 12425 → 12435

13|425 13345

solution is not acceptable

so can we design new crossover operator

you can apply this operator but you have to check every number must be distinct

i.e updation of previous operator (in case of repeating city, replace with city not present)

operator is same only updation is required

for improving convergence rate, initial population is to be improved & exploitation is required

point to point based algorithm simulated annealing

population based algorithm is GA, PSO, DE

for diversity, exploration is required.

Types of encoding – binary
~ real parameter

min. cost spanning tree
create a tree with least weightage

can GA be applied? Yes

any advantage if we use GA
no, we apply greedy method although GA is a optimisation problem as there is no permutation that is global optimum & can be only given by GA

single source shortest path

all pair shortest path

in 0-1 knapsack we apply GA

randomized algorithm requires less time as compared to deterministic algorithm as randomized check only random sol^n while deterministic check each solution

in Dykstra's, GA can be applied? No

Quantum ly A-i Quantum encoding → quantum bits

we use probabilistic model ; related to Quantum theory

we use q bits which can be converted to binary bits which can be represented as $\{\alpha, \beta\}$ α^2 → probability of being in 0 state

β^2 → ——————— 1 state

$\therefore \alpha^2 + \beta^2 = 1$

if $\alpha^2 = \beta^2$

$2\alpha^2 = 1$

$\alpha^2 = \frac{1}{2}$

$\alpha = \beta = \frac{1}{\sqrt{2}}$

we cannot work on quantum bits, we have to convert it into binary → since operations can be done only on binary

$\{\alpha, \beta\} = \{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\}$

$\{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\}$ $\{\frac{1}{2}, \frac{\sqrt{3}}{2}\}$ $\{\frac{\sqrt{3}}{2}, \frac{1}{2}\}$

r_1 r_2 r_3

8 states can be represented at a time

$r_1 = .4$ $r_2 = .7$ $r_3 = .8$

$\alpha > r_1$ $r_2 > \alpha$ $r_3 < \alpha$

1 0 1

101

quantum bits can be used anywhere.

advantage → ① exploration will increase because random quantum bits are used/generated

② diversity is good but you cannot exploit the previous solution so exploitation will be minimum

③ you have to define the rule : whatever you assume mention it

if $(\alpha > r)$
    make it 1
else
    make it 0

or

if $(\alpha < r)$
    make it 0
else
    make it 1

if you cannot apply any exploitation mechanism, then ~~use~~ this is a disadvantage.

if probability $= \frac{3}{4}$

$$\alpha^2 = \frac{3}{4} \qquad \alpha = \frac{\sqrt{3}}{2}$$

if probability in 0 state is $\frac{1}{4}$

write $\left\{\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}$ ⚠ never write single value

~~single~~ multi objective optimisation . we have to deal with multiple conflicting objective optimisation

if one is increasing, another is decreasing

eg → $f_1(x) = \sin x \qquad 0 \le x \le \frac{\pi}{2}$

here $x = \frac{\pi}{4}$ $(0,1)$ $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ $(1,0)$

somebody will say 0 or some will say $\frac{\pi}{2}$
there are many optimal solution but incomparable

This will make a curve known as Pareto front. (scientist in economics)

in Pareto front, each & every solution is important but they are not comparable

eg at $x = 0$ $\quad x = \pi/4$

$f_1(x) = 0 \quad f_1(x) = \frac{1}{\sqrt{2}}$

$f_2(x) = 1 \quad f_2(x) = \frac{1}{\sqrt{2}}$

not comparable

⚠ $\max f_1(x) = \sin x$
$\min f_2(x) = \cos x$
This is single objective optimisation

Non-dominated sorting or multi objective sorting

	min $f_1(x)$	max $f_2(x)$	
x_1	1	2	
x_2	2	1	dominated set
x_3	3	4	
x_4	4	3	
x_5	5	5	
x_6	6	1	

solution which are best will create a class as we apply the ranking method and will be called Non dominating front of class 1.

Dominance check shell

x_1 will dominate x_2 if atleast in one it is good & equal in all other objective.

compare x_1 with all others

x_1 is better than x_2 in $f_1(x)$ & $f_2(x)$

so x_1 is dominating

Dominating	dominated
x_1 x_4	x_2 x_6
x_3 x_5	

x_1 & x_3 cannot be dominated by each other

after comparing everyone with x_1

choose x_3 & compare with all in that class

finalised front of class 1 — x_1, x_3, x_5

x_2, x_4, x_6 — here $x_2 \equiv x_4$, $x_6 > x_2$

as x_3 dominated x_4

now in dominated class check with x_2

x_6 dominated x_2

x_1, x_3, x_5 — front 1

x_4, x_6 — front 2

x_2 — front of class 3

complexity here is ~~n^2~~ $m \cdot n^3$

$n \rightarrow$ no of solutions
$m \rightarrow$ no of function to objective

	$f_1(x)$ min	$f_2(x)$ min	$f_3(x)$ max
x_1	1	1	3
x_2	2	2	1
x_3	3	4	4
x_4	4	3	5
x_5	5	2	6
x_6	1	2	1

comparing x_1 with x_2

$x_1\ x_1\ x_1$ so x_1 dominates x_2

dominating: x_1, x_3, x_4, x_5 dominated: x_2, x_6

$x_1\ x_1\ x_3$

so not dominating

with x_3

$x_1\ x_3\ x_4\ x_5$ $x_2\ x_6$

with x_4

$x_1\ x_3\ x_4\ x_5$ $x_2\ x_6$

x_2 with x_6

$x_1\ x_3\ x_4\ x_5$ x_6 x_2

dominating front | front 2 | front 3

if criteria not given assume, all are minimum

What will happen if we increase objective functions
fronts will be less.

optimal Non dominated front is pareto front