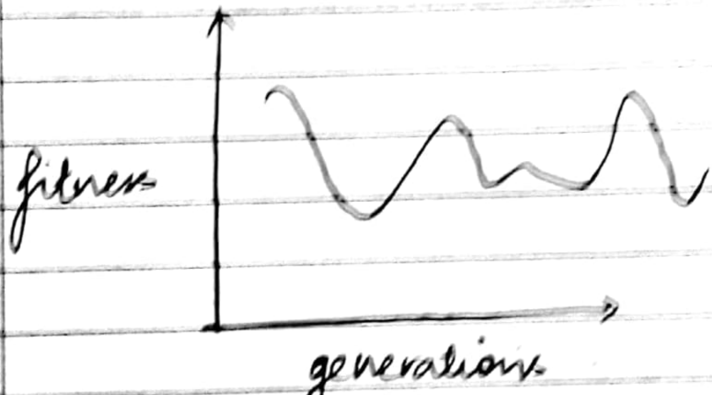
Soft computing

Genetics:

Mutation : Exploration
Cross over : Exploitation

Search space : Combination of all features

We search through this search space & find the best solution that satisfy the objective / fitness solution



→ used in drug manufacturing / research

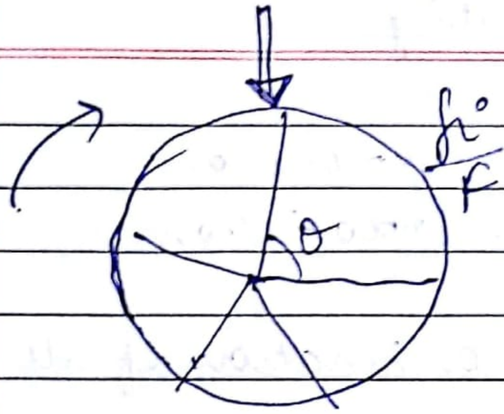
Selection :

we select who all will move onto next generation

$$F = \sum_{i=1}^{N} f_i$$

Rollerwheel selection:

Expected value $(i, t) = \dfrac{fit(i,t)}{F}$ t : time, i : individual



Roll n times.
we select the one which the header points to.

Implementⁿ:

→ Generate Probability distribution
→ Generate the cummulative distribution

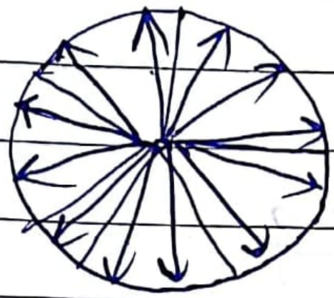
Exploitation ↑
Homogeneity of population ↑

Convergence

Pre-mature convergence

Stochastic uniform selection

↳ Randomness associated



We have multiple equi-spaced pointers. There is only one rotation required.

$P = \frac{F}{N}$

selector = Rand( ); (from 0 to F)

$\underbrace{\left\{ ptr := selector + i * P \;\Big|_{i=0}^{N-1} \right\}}_{set}$

```
for (i=0; i<N; i++)
    for( sum += fit(i,t), sum >= ptr, ptr++)
        select i;
```

for any individual:

min chance to get selected = $\lfloor fit(i,t) \rfloor$

max chance to get selected = $\lceil fit(i,t) \rceil$

Selection process is maintained at middle middle level.
→ Still there is chance of convergence

Sigma Scaling:

- also maintains a constant selection pressure
- make proportionality non-linear
- $$\text{Exp\_val}(i,t) = 1 + \frac{f(i,t) - \overline{f}(t)}{2\,\sigma(t)}$$

→ avg. fitness

standard deviation

If $\sigma(t) = 1$, exp_val $= 1.5$
Lowest exp_val $= 0$

Initially, if σ is ↑, almost all of the population will fall in it.

Later on, σ ↓ since only fit indivi. are only coming (prioritised),
⇒ exp_val ↑, for low fitness individuals, ~~too~~ the chance ↑ too.

→ selection

There may still be some sort of premature convergence

⇒ By this, we ensure variety.

However, this is also not the best way.

Anealing of glass

Boltzmann Selection

- We try to make exp-val non-linear, rather exponential

$$\text{Exp-val}(i;t) = \frac{e^{f(i;t)/T}}{\langle e^{f(i;t)/T} \rangle_t}$$

T → Temperature

average at time t

- This has the property that difference of min exp-val & max exp-val increases as you reduce the temperature

⇒ Selection pressure will increase

But

⇒ This will happen in a 'smooth' manner & I can control it.

So I will first ↑ temp. Harder for

Soft Computing - 3

- Parameters are used by user to control the system.

- There may be a multi-optimization problem.

- Given a generation, what is can be suitable crossover rate or mutation rate

- We want a balance b/w exploration & exploitation

- Quality of $soln$ depends on the length of simulation

- Boltzmann required exponential Calculations. To ↑ computational efficiency, we use some other methods of selection.

* **ELITISM:** We retain the best individuals (1 or 2) of the current $gener^n$ & move directly to next $gener^n$, and leave rest for mutaty & ~~evo~~ crossovers and allow them to evolve through the next generation. We do it at every $gener^n$.

Schema theory - why GA works?

classmate
Date
Page

Rank Selection:

We give ranks to each individual & expected value is fn of this rank, rather than fitness value.

$$Exp\_val(i,t) = min + (max - min)\left(\frac{Rank(i,t) - 1}{N-1}\right)$$

- very effective method
- earlier we were finding exp_val on the abs. difference b/w fitness. That remained constant over generations.
- Until ranks of all become almost same, selection pressure remains low & almost constant

Tournament Selection:

$r = rand(0,1)$

parameter $k = 0.75$ → in our hands

We choose two random individuals

If $r < 0.75$, we select the more fit

else we select the less fit

Scanned by CamScanner

Steady state Selection

generally used with elitism

- CUCKOO SEARCH ALGORITHM
  - meta heuristic

  ⇓

- Levy's flight

  ⇓

  Cauchy distribution

- Objective function should be some minimisation or maximisation

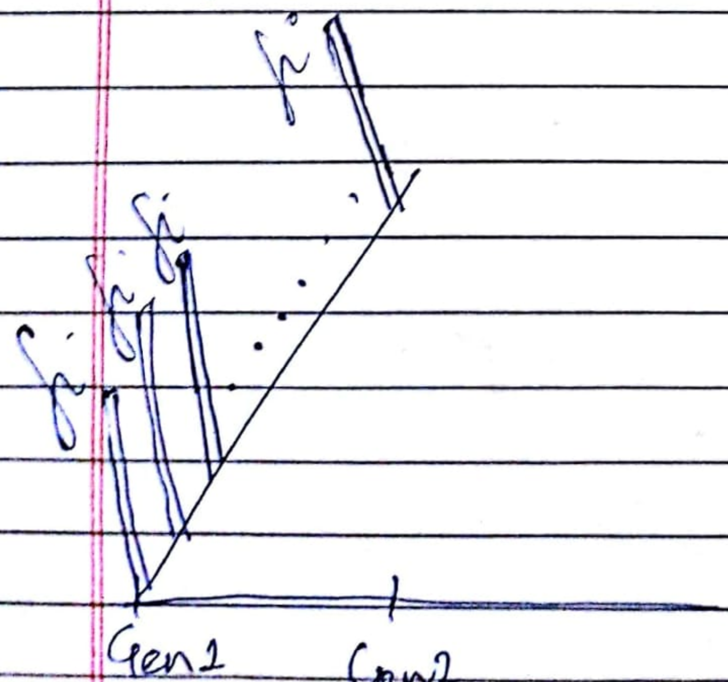
- fitness must lead to the objective function

- Determine the population size

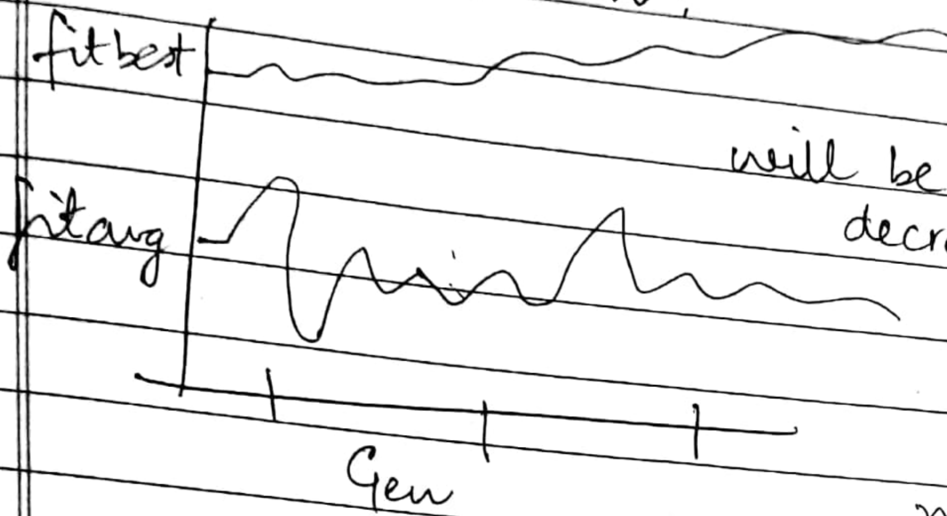
- If you have a smaller population, more evolution

- Computational complexity = Populⁿ size * No. of generⁿ

- You need to explore the search space



random;



will be a non-decreasing function

not entirely random

→ Then we do a random walk.

— In crossover, you have to see whether the offspring does not violate any constraint.

$$\frac{w_1 OF_1}{\Sigma w} - \frac{w_2 OF_2}{\Sigma w} + \frac{w_3 OF_3}{\Sigma w}$$

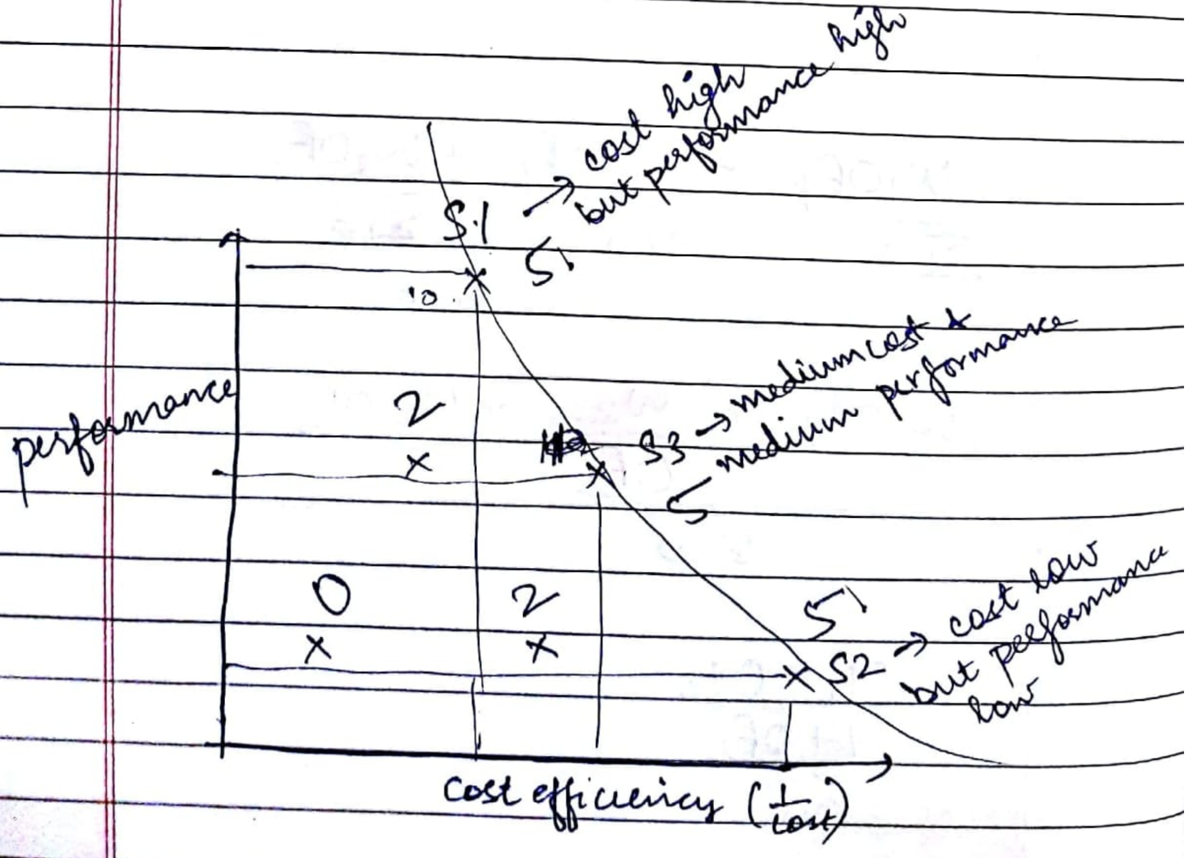
$$\frac{w_1 OF_1 + \frac{w_2}{OF_2} + w_3 OF_3}{\Sigma w}$$

$$\frac{OF_1 + OF_3}{1 + OF_2}$$

- Non-inferior solutions:
  Those solutions who are not better than other non-inferior solutions in all respects.

- Step1: features
  Step2: constraints
  Step3: threshold
  Step4: objective function

$OF = \frac{w_1}{\sum w_i} f_1 + \frac{w_2}{\sum w_i} f_2$



S1, S2, S3 are non-inferior solutions.

Through weighted mean, I will get the averagely best soln S3. S1 and S2 will be lost.
I would rather have a set of non-inferior solutions

↳ $\boxed{\text{pareto-optimal solution}}$

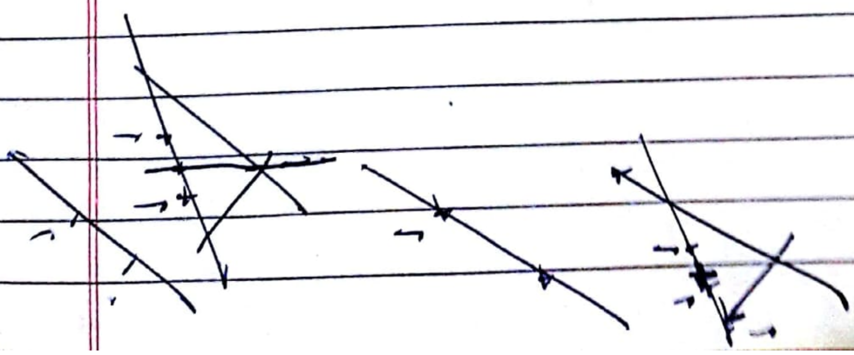
Rank: No. of solutions which are not superior in all dimensions

FORAGING BY ANIMALS:

Ant family:
↓
release feromone
other ants sense pheromone & follow path with high ~~feromone~~ pheromone

⇓

$\boxed{\text{Ant Colony Optimisation}}$



Ant Colony Optimisation

Two parameters

Heuristic η ⟷ Pheromone τ

- greedy
- local search (finding the local best)

- based on experience
- cooperation (kinda global search)

Combinatorial Optimisation Problem :

$$\langle S, C, \overset{\text{optimisation fn}}{f} : S \rightarrow R_0^+ \rangle$$

S: set of features $= \{x_i\}$ $i = 1$ to N
C: set of constraints
f: Objective function

Scanned by CamScanner

Solution ≡ Set of assigned features

DATE
PAGE

Features have values which belong to some domain.

Feasible solution: S_C → Set of features which satisfy certain constraints

Define global minima/max → you may not reach it but still we define it, because that is the ultimate goal
$\quad \hookrightarrow S^*$

$$f(S_C^*) \leq f(S_i) \quad \forall S_i \in S_C$$

global optima

TSP using ACO :

Initialize
  Set Parameters.
  Initialise pheromone levels for some paths
Do
  Begin
    Construct Ant Solution → give random starting points to ants
      → Greedy / Probabilistic
    Synchronize (optional)
    Update pheromone
  End
Till (Stop criteria)
  ↳ convergence criteria
    ↓
  In ACO, it is ~~almost~~ often 'n' iterations

Scanned by CamScanner

Probability of choosing a next node, given a set of neighbours

DATE
PAGE

$$P(C_{ij})\Big/_{S^P} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{c_{i-l} \in S^P} \tau_{il}^{\alpha} \eta_{il}^{\beta}}$$

S^P : neighbours

τ_{ij} → Pheromone trail b/w i & j

η_{ij} → the local factor (min distance)

$$\eta \propto \frac{1}{L_{ij}} = \frac{Q}{L_{ij}}$$

Synchronise :

In GAs, solutions interact with each other using crossovers. That is not directly present here.

In ACO, there may also be communication b/w ants.
∴ This optional step.

$$\text{Update: } \tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho \cdot \sum F(s)$$

ρ → evaporation rate

$F(s)$ → fitness fn of all the solutions

$$s \in \{\ \}_{upd} \Big|_{c_{ij} \in s}$$

What should this set be?
- All the ants who travelled from i→j
- or only the fittest ones who travel the i→j

when you want to consider the one jiska argument has max value

DATE
PAGE

$$S_{upd} = \underset{s \in S_{iter}}{\operatorname{argmax}} (F_s)$$