Flower Pollination Algorithm (FPA)

- It is Nature inspired population based Metaheuristic optimization Algorithm
  - Inspired by the flowering behaviour of the plants to do pollination.
  - Mimics the pollination process in a flowering Plant
  - very superior than different other Metaheuristic o/p Algorithms

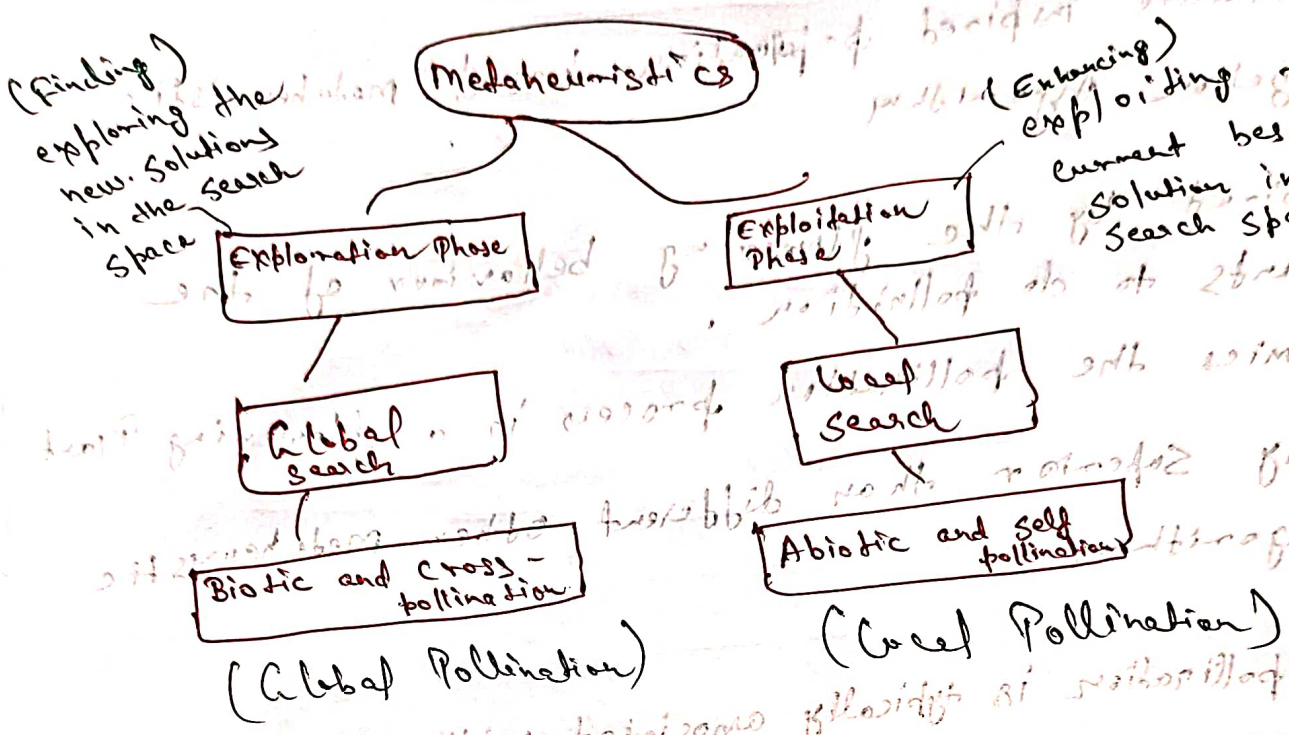
Note:

- Flower pollination is typically associated with the transfer of poller.
- Pollinators such as ~~birds~~, insects, birds, bats and other animals are responsible to transfer ~~the~~ poller for flower pollination.

Pollination.

- Abiotic (10% belongs)
  - Pollinators: insects, birds, bats, animals.
- Biotic (90% belongs)
  - Pollinators: winds, diffusion
  - does not require pollinators.

Note :- FPA Maintains balance between exploration & exploitation.



Note :-

1) Too much exploration can lead to Slow Convergence and Inefficiency.
2) Too much exploitation might cause convergence to Sub optimal solution (local Maxima / Minima).

Mathematical model for ~~Global~~ Search in FPA

Rule 1 : Global Search in FPA

Global Pollination

Biotic Pollination — Cross Pollination

Levy flights. (L)

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(g^* - x_i^t)$$

- x_i^{t+1} : New / Pollen Solution
- x_i^t : Old / Pollen Solution
- γ : Scaling factor to control step size
- $L(\lambda)$: Levy distribution (Step size)
- g^* : best Solution in population

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \cdot \frac{1}{s^{1+\lambda}}$$

where $L > 0$, (s ~~$\gg$~~ s_0)

$\Gamma(\lambda)$: Standard Gamma function, $\lambda = 1.5$

Rule 2 :- Local Search in FPA

Local Pollination

Abiotic Pollination — Self Pollination

$$x_i^{t+1} = x_i^t + \varepsilon (x_j^t - x_k^t)$$

ε: uniform distribution, $\varepsilon \in [0, 1]$

x_j^t, x_k^t: pollen from different flowers

Rule 3 :- Pollinators :- Flower Constancy.

Pollinators such as insects can develop flower constancy, which is equivalent to reproduction probability that is proportional to the similarity of two flowers involved.

Rule 4 :- Switch Probability

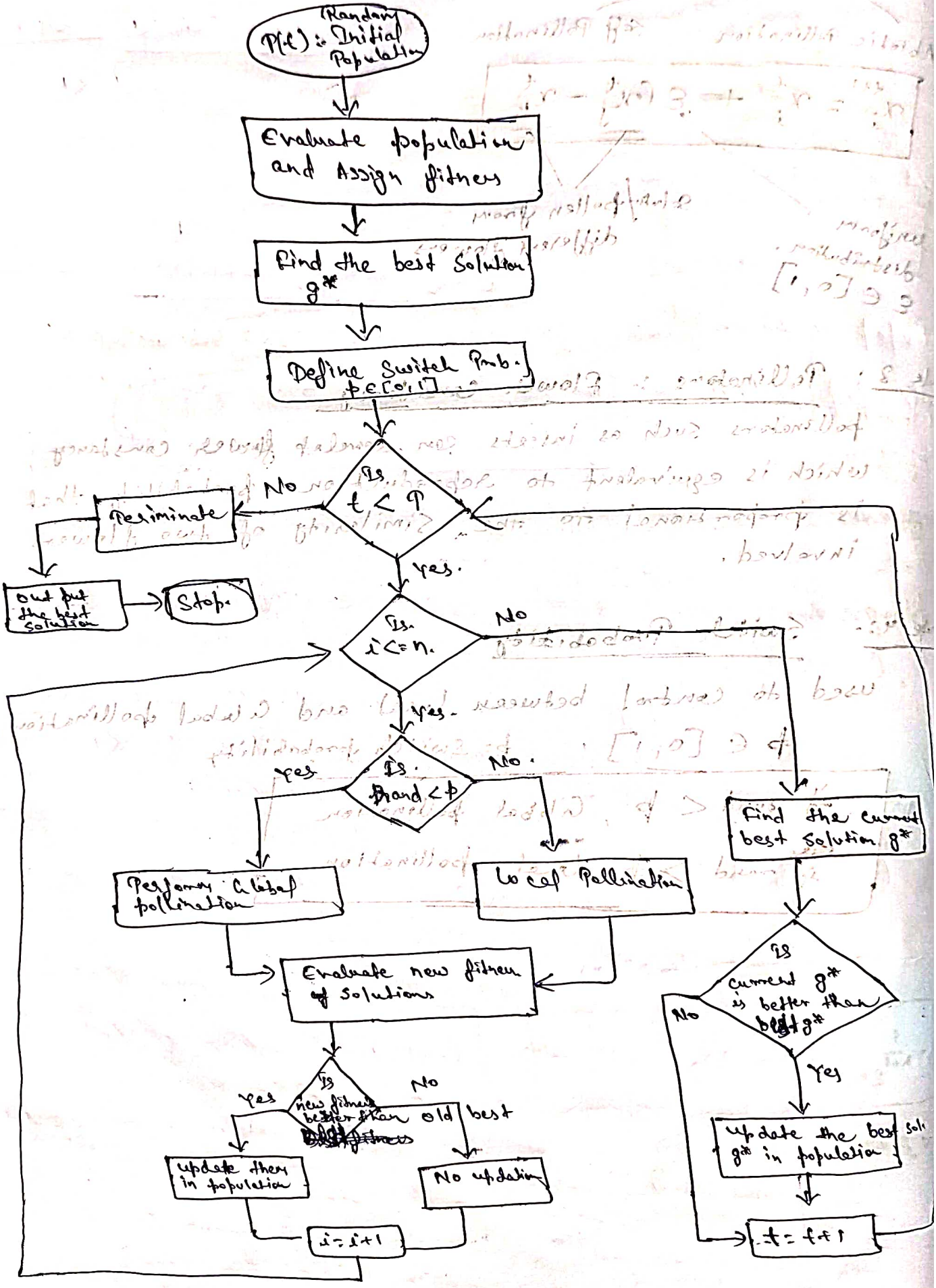
used to control between local and Global pollination.

$p \in [0, 1]$ p = switch probability

if rand $< p$, Global pollination

if rand $> p$ local pollination.

Flow chart of FPA



Algorithm of FPA

1. objective min / max $f(x)$, $x = (x_1, x_2, x_3, x_4, \ldots, x_d)$
2. Initialize a population of n flowers / pollen gametes with random solutions.
3. Find the best solution g^* in the initial solution
4. Define a switch probability $p \in [0, 1]$
5. while ($t <$ Max Generation)
6. for $i = 1 : n$ (for all n flowers in the population)
7. if rand $< p$
8. Draw a (d-dimensional) step vector L from a
9. Global pollination: $x_i^{t+1} = x_i^t + \gamma L (g^* - x_i^t)$ levy Distribution
10. else
11. Draw ε from a uniform distribution in $[0, 1]$
12. Do local pollination $x_i^{t+1} = x_i^t + \varepsilon (x_j^t - x_k^t)$
13. endif.
14. Evaluate new solutions
15. If new solution are better, update them in population
16. endfor
~~17. end while~~
17. Find the current best solution g^*
18. if current g^* are better, update best solution g^* in population
19. $t = t + 1$
20. end while
21. Produce the best solution g^* as output.

FPA Algorithms steps

Step 1 :- Initialization

1) Initialize Parameters : n, $t=0$, T, ϕ, β

2) Initialize Population
   ⊢ n flowers/pollen gametes with random solution

Step 2 :- Evaluate fitness $f(x_i^t)$

1) Calculate the fitness value of each flowers in populat

2. Find the best solution g^* in the initial population

Step 3 :- Define a switch probability

$\phi \in [0,1]$

Step 4 :- Check for stopping criteria.

$t < T$

Step 5 :- For all the n flowers in population

1) if rand < Switch probability (ϕ)
   Draw a (d- dimensional) step vector L from a (levy distribution)
   Perform Gobal pollination : $x_i^{t+1} = x_i^t + \gamma L (g^* - x_i^t)$

2) if rand > Switch probability (ϕ)
   Draw from uniform Distribution in $[0,1]$

3) Evaluate New Solutions fitness

4) If new solution are better then old solution, then update, or else ~~does~~ no need to update

Step 6 :- Find the current best solution g^* in population

1) Find the current best solution g^* in population

2) If current best solution g^* in the population best ~~g*~~ solution g^* is better the ~~old~~

3) update best solution g^* in

4) $t = t+1$

Step 7 :- Produce g^* output