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 pollen.
- Pollinators such as ~~winds~~, insects, birds, bats and other animals are responsible to transfer ~~the~~ pollen from flower pollination.

Pollination.

Abiotic
(10% belongs)
Pollinators:- insects, birds, bats, animals.

Biotic
(90% belongs)
Pollinators → does not require pollinators.
winds, diffusion

Note: FPA Maintains balance between exploration and exploitation.

Metaheuristics

(Finding) exploring the new solutions in the search space

(Enhancing) exploiting the current best solution in search space

Exploration Phase → Global search → Biotic and cross-pollination (Global Pollination)

Exploitation Phase → Local search → Abiotic and self pollination (Local Pollination)

Note:
1) Too much exploration can lead to slow convergence and inefficiency.
2) Too much exploitation might cause convergence to suboptimal solution (local maxima / minima).

Mathematical model for ~~Global~~ Search in FPA

Rule 1: Global Search in FPA

Global Pollination
- Biotic Pollination
- Cross Pollination

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

- 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

Levy flights. (L)

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

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

Γ — Standard Gamma function

$\lambda = 1.5$

Rule 2 :- local Search in FPA

local Pollination

Abiotic Pollination — Self Pollination

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

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

solution/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

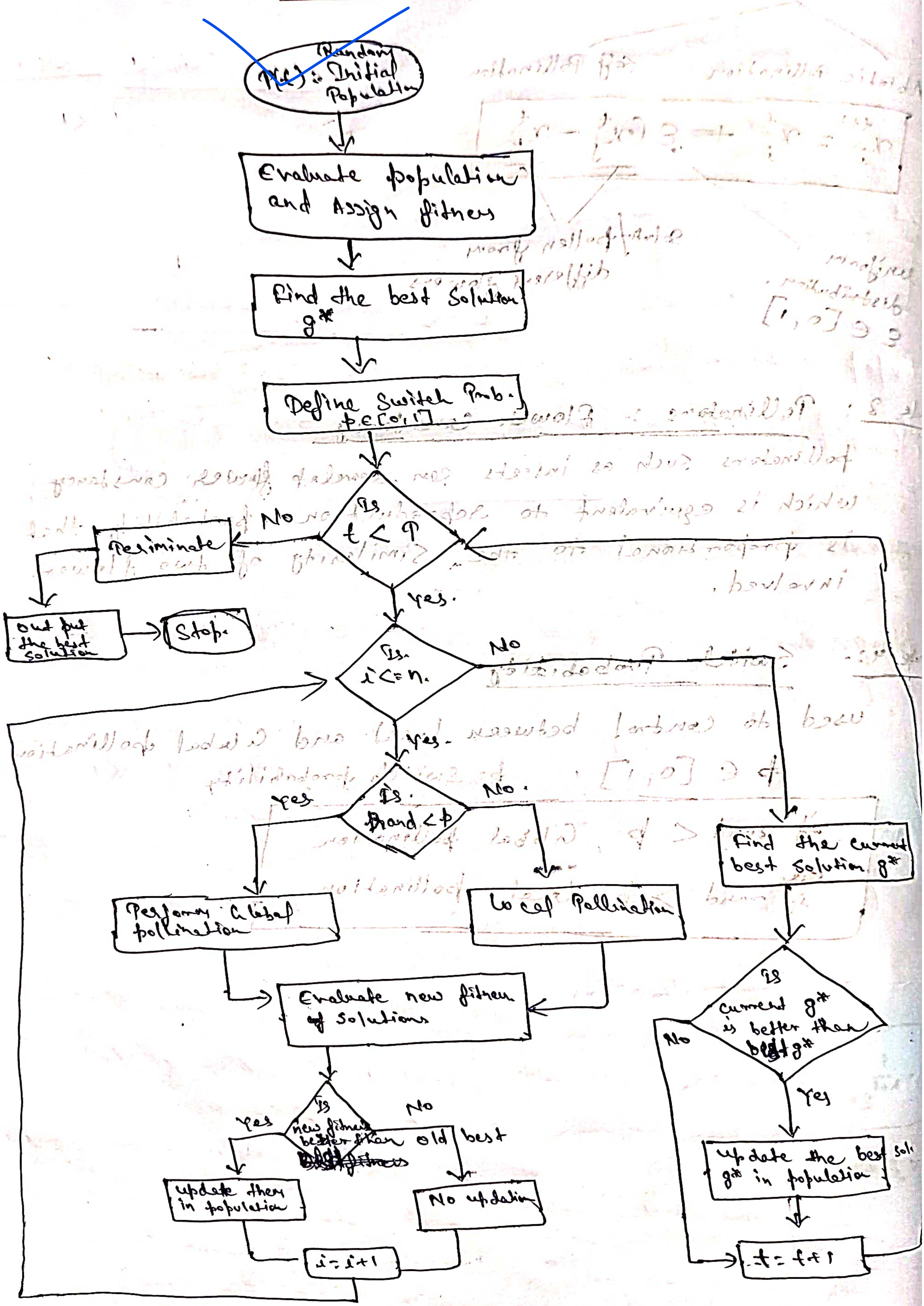
used to control between local and Global pollination.

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

if rand < p, Global pollination

if rand > p, local pollination,

Flow charts of FPA



Algorithm of FPA

1. objective min | max $f(x)$, $x = (x_1, x_2, x_3, x_4 \dots x_d)$
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 levy Distribution
9. Global pollination: $x_i^{t+1} = x_i^t + \gamma L(g^* - x_i^t)$
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 solutions 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
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 population
2. Find the best solution g^* in the initial population

Step 3 :- Define a switch probability

$p \in [0,1]$

Step 4 :- Check ~~for~~ stopping criteria.

$t < T$

Step 5 :- ~~For~~ all the n flowers in population

1) if rand < switch probability (p)
   Draw a (d-dimensional) step vector L from a Levy distribution
   Perform Global pollination: $x_i^{t+1} = x_i^t + \gamma L (g^* - x_i^t)$
2) if rand > switch probability (p)
   Draw from uniform Distribution in [0,1]
3) Evaluate New Solution, fitness
4) If new solution are better than 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