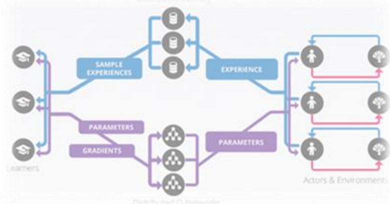
 

Binary Flower Pollination Algorithm

Min-Uk Lee

2018. 02. 02.

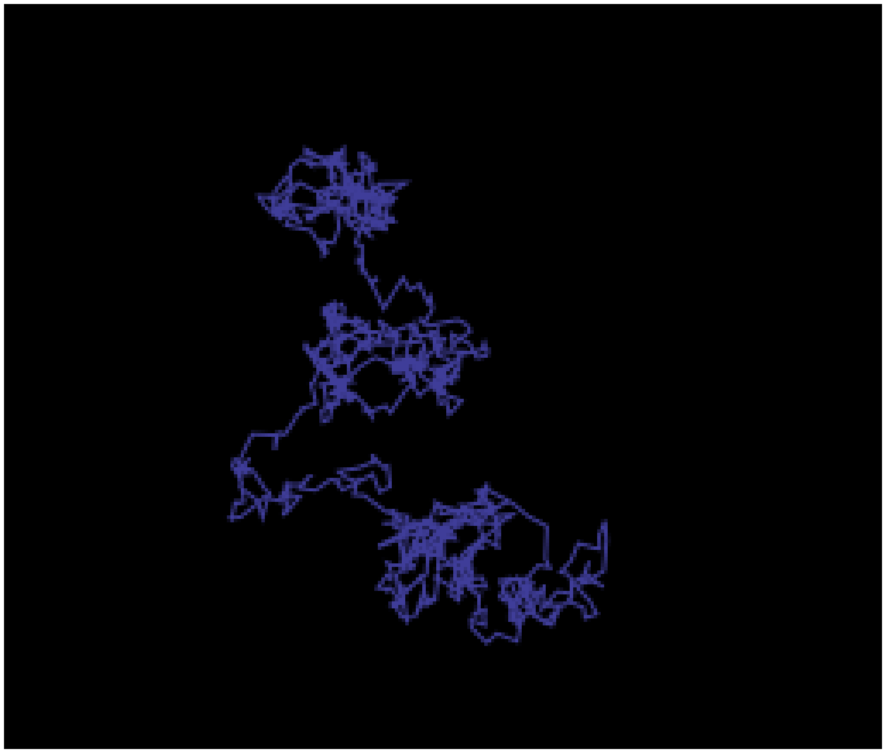
Flower Pollination

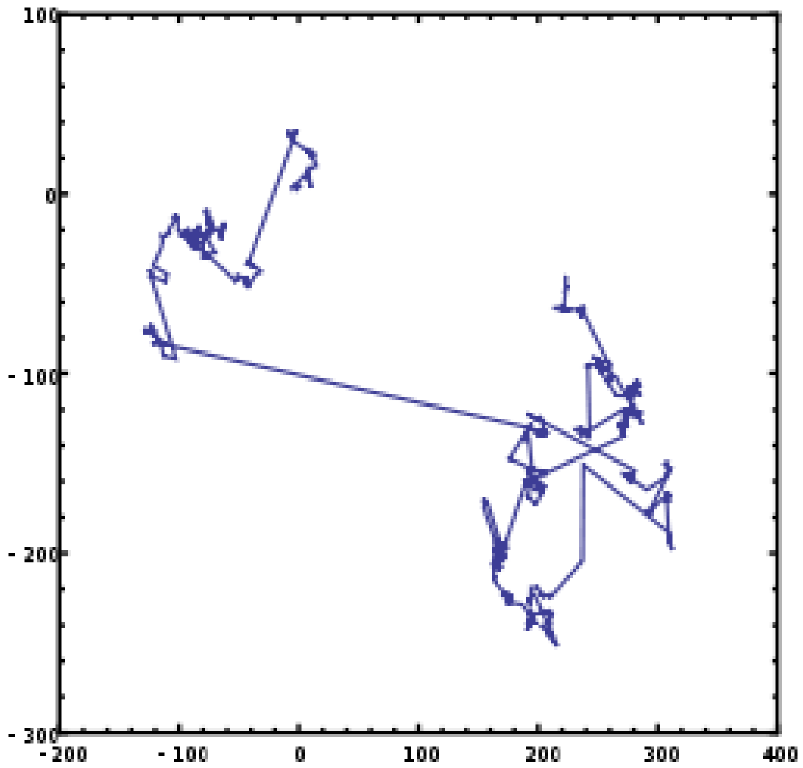
- Self-Pollination
  - No reliable pollinator

- Cross-Pollination
  - Long distance
  - Need Pollinator
    - Levy Flights

Levy Flights

- Markov Processes





Flower Pollination Algorithm Rules

1. Biotic and cross-pollination can be considered as a process of global pollination, and pollen-carrying pollinators move in a way which obeys Levy flights
2. For local pollination, abiotic pollination and self-pollination are used.
3. Pollinators such as insects can develop flower constancy, which is equivalent to a reproduction probability that is proportional to the similarity of two flowers involved.
4. The interaction or switching of local pollination and global pollination can be controlled by a switch probability $p \in [0,1]$, slightly biased towards local pollination.

Flower Pollination Algorithm

- Rule 1, 3 (Cross-pollination)
  - $x_i^{t+1} = x_i^t + \gamma L(\lambda)\left(g_* - x_i^t\right)$
  - x_i^t : solution vector x_i at iteration t
  - g_* : the current best solution
  - γ : scaling factor to control the step size (hyperparameter)
  - $L(\lambda)$: Levy-flights-based step size
    - $L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}$, $(s \gg s_0 > 0)$.
    - $s = \frac{U}{|V|^{1/\lambda}}, U \sim N(0, \sigma^2), V \sim N(0,1)$
    - $\sigma^2 = \left[\frac{\Gamma(1+\lambda)}{\lambda\Gamma((1+\lambda)/2)} \frac{\sin(\pi\lambda/2)}{2^{\lambda-1/2}}\right]^{1/\lambda}$
    - This distribution is valid for large steps $s > 0$.
- Rule 2
  - $x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t)$
  - x_j^t, x_k^t :pollens from the different flowers of the same plant species.
  - ϵ : uniform distribution in [0,1]
- Rule 4
  - p: to mimic the local and the global flower pollination, the switching probability, $p = 0.8$

Flower Pollination Algorithm

Flower Pollination Algorithm (or simply Flower Algorithm)

Objective min *or* max $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, ..., x_d)$
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution $\mathbf{g}_*$ *in the initial population*
Define a switch probability $p \in [0, 1]$
while *(t <MaxGeneration)*
  for $i = 1 : n$ *(all n flowers in the population)*
    if *rand* $< p$,
      Draw a (d-dimensional) step vector L which obeys a Lévy distribution
      Global pollination via $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + L(\mathbf{g}_* - \mathbf{x}_i^t)$
    else
      Draw ϵ *from a uniform distribution in [0,1]*
      Randomly choose j and k among all the solutions
      Do local pollination via $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \epsilon(\mathbf{x}_j^t - \mathbf{x}_k^t)$
    end if
    Evaluate new solutions
    If new solutions are better, update them in the population
  end for
    Find the current best solution $\mathbf{g}_*$
end while

Figure 1: Pseudo code of the proposed Flower Pollination Algorithm (FPA).

Binary Flower Pollination Algorithm

- $S(x_i^j((t)) = \frac{1}{1+e^{-x_i^j(t)}}$
- $x_i^j(t) = \begin{cases} 1\ if\ S\left(x_i^j(t)\right) > \sigma \\ \quad 0 \quad otherwise \end{cases}$
- $\sigma \sim [0,1]$

Binary Flower Pollination Algorithm

- $x_i^{t+1} = x_i^t + \gamma L(\lambda)\left(g_* - x_i^t\right)$
  - Diffusion Process

- $x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t)$
  - Convergence Process

Binary Flower Pollination Algorithm

Algorithm 1: BFPA - Binary Flower Pollination Algorithm

input : Training set Z_1 and evaluating set Z_2, α, number of flowers m, dimension d and iterations T.
output : Global best position $\hat{g}$.
auxiliaries: Fitness vector f with size m and variables acc, $maxfit$, $globalfit$ and $maxindex$.

for *each flower i* ($\forall i = 1, \ldots, m$) **do**
    for *each dimension j* ($\forall j = 1, \ldots, d$) **do**
        $x_i^j(0) \leftarrow \text{Random}\{0,1\}$;
    $f_i \leftarrow -\infty$;
$globalfit \leftarrow -\infty$;
for *each iteration t* ($t = 1, \ldots, T$) **do**
    for *each flower i* ($\forall i = 1, \ldots, m$) **do**
        Create Z_1' and Z_2' from Z_1 and Z_2, respectively, such that both contains only features such that $x_i^j(t) \neq 0$, $\forall j = 1, \ldots, d$;
        Train OPF over Z_1', evaluate its over Z_2' and stores the accuracy in acc;
        if ($acc > f_i$) **then**
            $f_i \leftarrow acc$;
            for *each dimension j* ($\forall j = 1, \ldots, d$) **do**
                $\hat{x}_i^j \leftarrow x_i^j(t)$;
    $[maxfit, maxindex] \leftarrow max(f)$;
    if ($maxfit > globalfit$) **then**
        $globalfit \leftarrow maxfit$;
        for *each dimension j* ($\forall j = 1, \ldots, d$) **do**
            $\hat{g}^j \leftarrow x_{maxindex}^j(t)$;
    for *each flower i* ($\forall i = 1, \ldots, m$) **do**
        for *each dimension j* ($\forall j = 1, \ldots, d$) **do**
            $rand \leftarrow \text{Random}\{0,1\}$;
            if $rand < p$ **then**
                $x_i^j(t) \leftarrow x_i^j(t-1) + \alpha \oplus \text{Lévy}(\lambda)$; **else**
                    $x_i^j(t) \leftarrow x_i^j(t-1) + \varepsilon(x_i^j(t-1) - x_i^k(t-1))$;
            if ($\sigma < \frac{1}{1+e^{x_i^j(t)}}$) **then**
                $x_i^j(t) \leftarrow 1$; **else**
                    $x_i^j(t) \leftarrow 0$;

Advantage, Disadvantage

- Advantage
  - Explore a whole new feature by Levy Distribution

- Disadvantage
  - Probability-base Model
  - Unclear diffusion process
  - Too many hyper parameters $(\sigma,\ \gamma,\ \lambda,\ p,\ \epsilon)$ relative to the performance of the algorithm
  - Difficulty in adjusting the maximum number of features
  - Step Size

References

- X. S. Yang, M. Karamanoglu, X. S. He, Flower Pollination Algorithm: A Novel Approach for Multiobjective Optimization Engineering Optimization, vol. 46, Issue 9, pp. 1222-1237 (2014).