BAT ALGORITHM

- What is BAT Algorithm?
- What is Micro bat?
- What is Echolocation?
- Bat Algorithm Rules.
- Velocity, Frequency, wavelength, loudness, pulse and position.
- How BAT Algorithm Works?
- BAT Algorithm Explanation with Example.

- Bat algorithm (BA) is a bio-inspired algorithm developed by Yang in 2010.
- BA uses a frequency-tuning technique to increase the diversity of the solutions in the population.
- BA uses the automatic zooming to try to balance exploration and exploitation during the search process by mimicking the variations of pulse emission rates and loudness of bats when searching for prey.

WHAT IS BAT ALGORITHM?

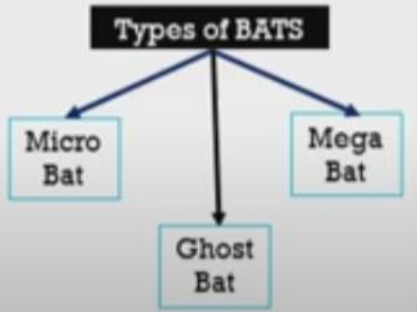
- There are about 1000 species of Bats.
- Bat algorithm is based on **Echolocation** behavior of Micro Bats.

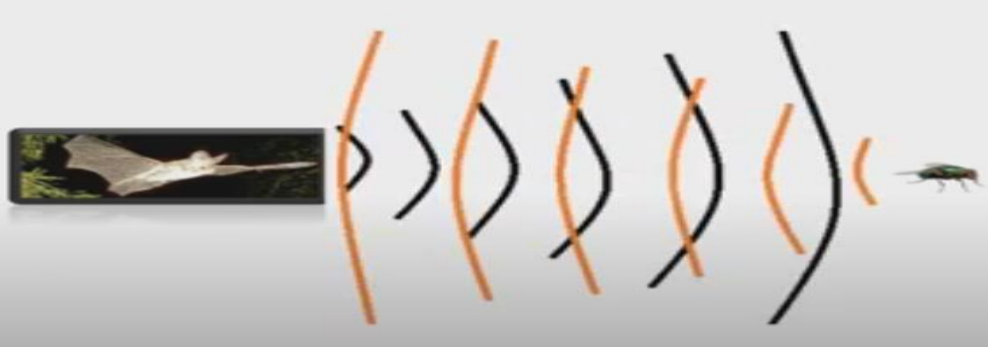
WHAT IS MICRO BAT?

- Micro Bats are **small to medium sized bats** that eat insects.
- Micro Bats used **a sonar** called **Echolocation** to **detect prey.**

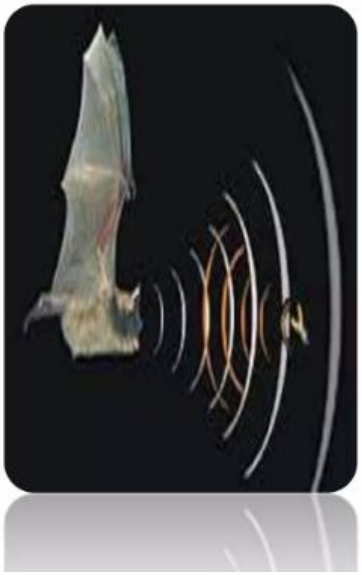
MICROBAT

Type : Flying Mammals
Size : Small to Medium
Length : 2.2 to 11cm
Wingspans : 25cm
Weight : 3g to 150g





- Microbats typically use a type of sonar, called, echolocation, to detect prey, avoid obstacles, and locate their roosting crevices in the dark.



- They can emit a very loud sound pulse and listen for the echo that bounces back from the surrounding objects.

- Their pulses vary in properties and can be correlated with their hunting strategies, depending on the species.

- All bats use echolocation to sense distance, and they also know the difference between food/prey and background barriers in some magical way

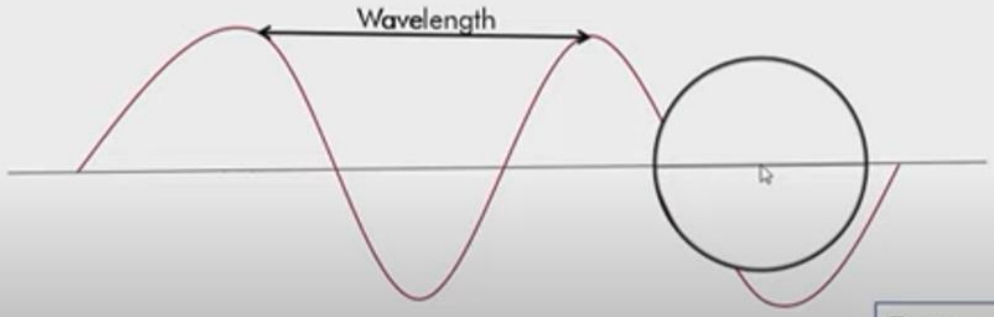
- Bats fly randomly with velocity v_i at position x_i with a frequency f_{min}, varying wavelength and loudness A_0 to search for prey.

- They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission r ϵ [0, 1], depending on the proximity of their target

BAT ALGORITHM – KEY POINTS

Sound Waves

- **Frequency:** Number of waves that passes a fixed point in unit time.



Frequency (f_i)

BAT ALGORITHM – KEY POINTS

- **Position:** A Place Where Something/someone is located.

Position (x_i)

How BAT Algorithm works?

- **Velocity:** Speed of something in a given direction.

Velocity (v_i)

How BAT Algorithm works?

- **Loudness** = Characteristics of Sound.
- **Loudness** refers to how loud or soft a sound seems to a listener.
- The **loudness** of sound is determined, by the intensity of the sound waves.

Loudness (A_i)

How BAT Algorithm works?

- **Pulse** = vibration or a wave.
- Rate of pulse emission can be speed unto about 200 pulses per second in air.

Pulse (r_i)

Bat Algorithm – (BA)...

- 1. All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers in some magical way;
- 2. Bats fly randomly with velocity V_i at position X_i with different frequency ranges $f_{[min,max]}$, varying wavelength λ and loudness A_0 to search for prey.
  - They can automatically adjust the wavelength of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target;
- 3. The loudness varies from a large (positive) A_0 to a minimum constant value A_{min}.

Bat Algorithm – BA...

- Each bat is randomly assigned a frequency between $[f_{min}, f_{max}]$, hence this algorithm is otherwise called as **frequency-tuning algorithm**
- Every bat is associated with velocity v_i^t and position x_i^t in search space at each iteration t, with respect to frequency f_i
- Hence at each iteration we need to update f_i, v_i and x_i as per the following equation
- $f_i = f_{min} + (f_{max} - f_{min})\beta$ where $\beta \in [0, 1]$
- $v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i$ where x_* is current best
- $X_i = x_i^{t-1} + v_i^t$

BAT ALGORITHM

Initialize the bat population x_i (i = 1,2,3,…n) and v_i

Initialize frequencies f_i , pulse rate r_i and the loudness A_i

Calculate fitness function. Select minimum value as the current best solution.

While (t < Max number of iterations)

Update frequency, velocity.

If (rand > r_i)

Select a solution among the best solutions & Generate a local solution around the selected best solution.

If (rand < A_i & f(x_i) > f(x_*))

Accept the new solution & Increase the r_i and reduce A_i

End if

Rank the bats and find the current best x_*

End while

Continue the LOOP until maximum number of iterations is reached.

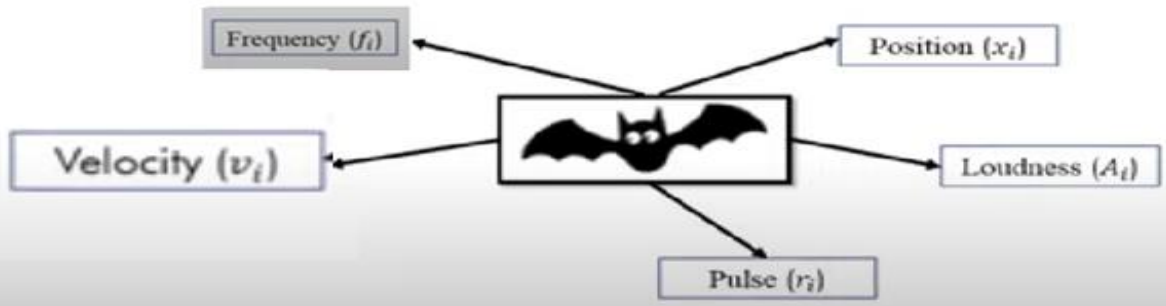
HOW BAT ALGORITHM WORKS?

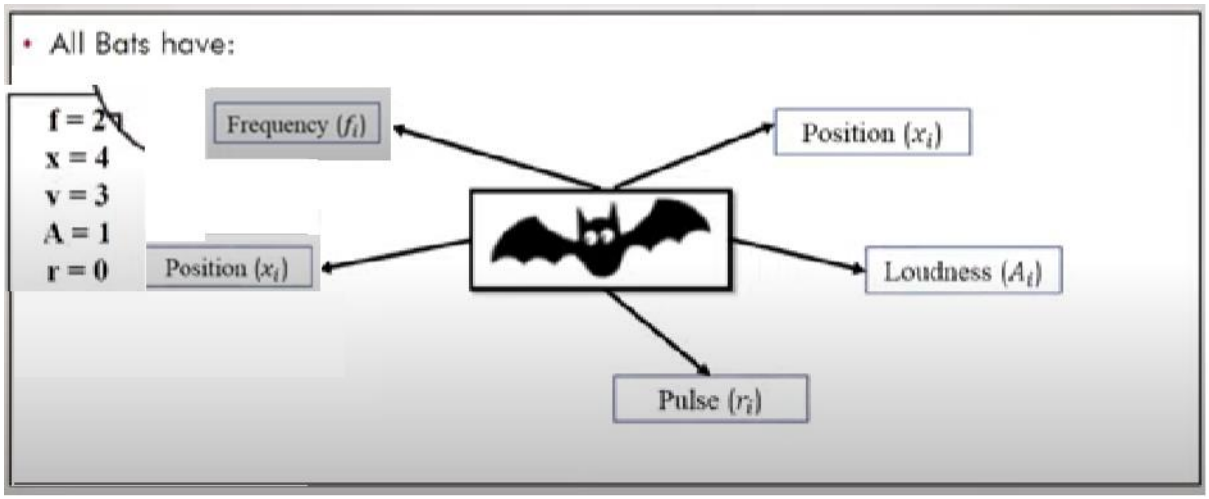
- Initialize the Bat Population x_i (i = 1,2,3…n).

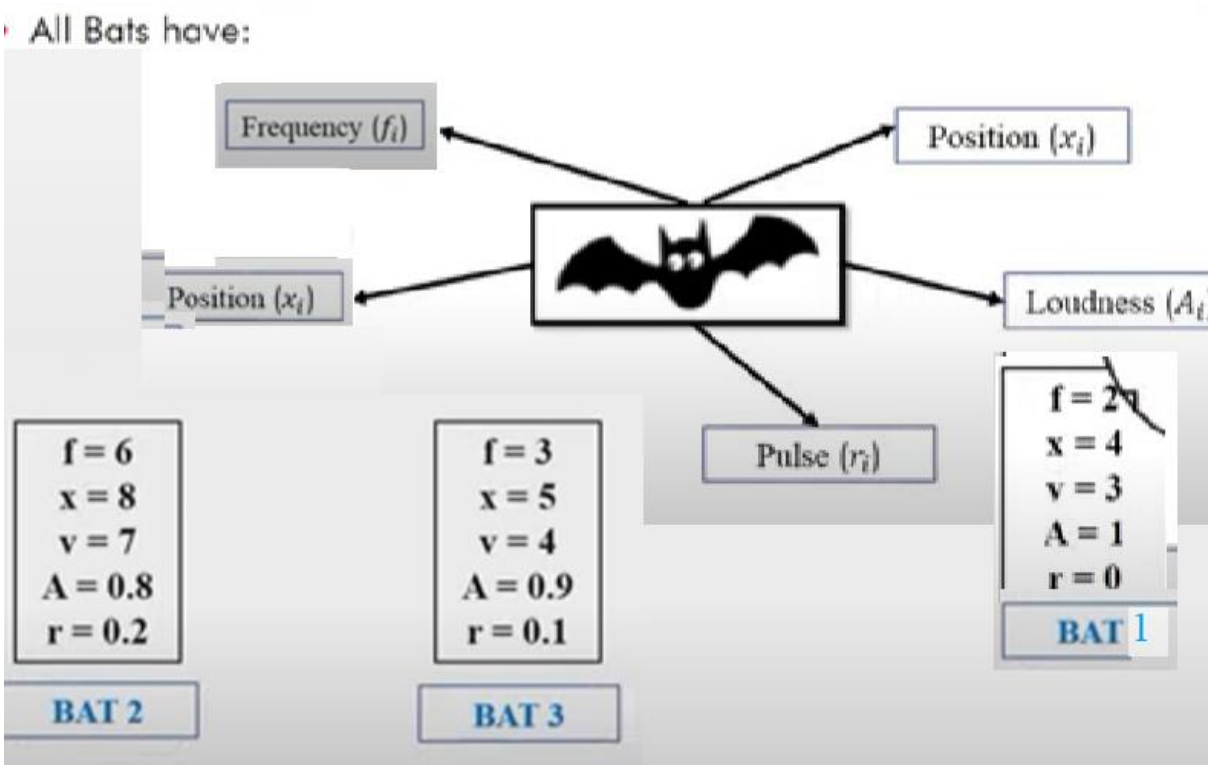
suppose we have three bats x_i (i = 1,2,3).

X1, X2, X3

- All Bats have:







Bat 1 fmax=8, Bat 2 fmax=7, Bat3 fmax=5

BAT ALGORITHM STEP BY STEP

- Calculation First BAT : First iteration (t = 1)

BAT 1

f = 2
x = 4
v = 3
A = 1
r = 0

While (t < Maximum number of iterations)

While (1 < 20)
Condition TRUE

If Condition is TRUE

Generate New Solutions by adjusting frequency and updating velocity and position.

- Calculation First BAT : First iteration (t = 1)

BAT 1

f = 2
x = 4
v = 3
A = 1
r = 0

$f_i = f_{min} + (f_{max} - f_{min})$

$f_{min} = 0$
$f_{max} = 8$

f_1, v_1, x_1, r_1, A_1

NOTE: BATs can automatically adjust frequency.

BAT ALGORITHM STEP BY STEP

- Calculation First BAT : First iteration (t = 1)

BAT 1

f = 2
x = 4
v = 3
A = 1
r = 0

t = 0

$f_i = f_{min} + (f_{max} - f_{min})\,\beta$

$f_1 = 2 + (8 - 2)\,1$
$f_1 = 8$

$\beta \in (0, 1)$

f_1, v_1, x_1, r_1, A_1

$[f_{min}, f_{max}] = [0, 10]$
f_{max} depends domain size of the problem.

BAT ALGORITHM STEP BY STEP

- Calculation for First BAT : First iteration (t = 1)

BAT 1

f = 2
x = 4
v = 3
A = 1
r = 0

t = 0

f = 8
x =
v =
A = 1
r = 0

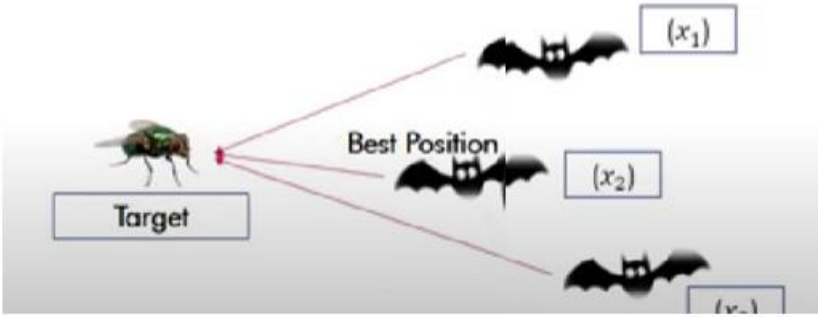
t = 1

$$v_i^t = v_i^{t-1} + (x_i^t - x_{gbest}^t) f_i$$

x_{gbest}^t = Current Global Best position.

- Example:

x_{gbest}^t = Current Global Best position.



BAT ALGORITHM STEP BY STEP

- Calculation for First BAT : First iteration (t = 1)

BAT 1

f = 2
x = 4
v = 3
A = 1
r = 0

t = 0

f = 8
x =
v =
A = 1
r = 0

t = 1

f_1, v_1, x_1, r_1, A_1

$$v_i^t = v_i^{t-1} + (x_i^t - x_{gbest}^t) f_i$$

$$v_1^1 = v_1^{1-1} + (x_1^0 - x_{gbest}^t) f_i$$

$$v_1^1 = v_1^0 + (x_1^0 - x_{gbest}^t) f_1$$

$$v_1^1 = 3 + (4 - 0)\, 8$$

=35

BAT ALGORITHM STEP BY STEP

BAT 1

- **Calculation for First BAT : First iteration (t = 1)**

f = 2
x = 4
v = 3
A = 1
r = 0

t = 0

f = 8
x =
v = 35
A = 1
r = 0

t = 1

f_1, v_1, x_1, r_1, A_1

$$x_i^t = x_i^{t-1} + v_i^t$$

$$x_1^1 = x_1^{1-1} + v_1^1$$
$$x_1^1 = x_1^0 + v_1^1$$
$$x_1^1 = 4 + 35$$
$$x_1^1 = 39$$

BAT ALGORITHM STEP BY STEP

- Values calculated for all BATs as:

f = 2
x = 4
v = 3
A = 1
r = 0

t = 0

f = 8
x = 35
v = 39
A = 1
r = 0

t = 1

BAT 1

f_1, v_1, x_1, r_1, A_1

f = 6
x = 8
v = 7
A = 0.8
r = 0.2

t = 0

f = 7
x = 71
v = 63
A = 0.8
r = 0.2

t = 1

BAT 2

f_2, v_2, x_2, r_2, A_2

f = 3
x = 5
v = 4
A = 0.9
r = 0.1

t = 0

f = 5
v = 29
A = 0.9
r = 0.1

t = 1

BAT 3

f_3, v_3, x_3, r_3, A_3

How BAT Algorithm works?

- Next check condition: Pulse Rate compare the pulse rate with random

If (rand > r_i)

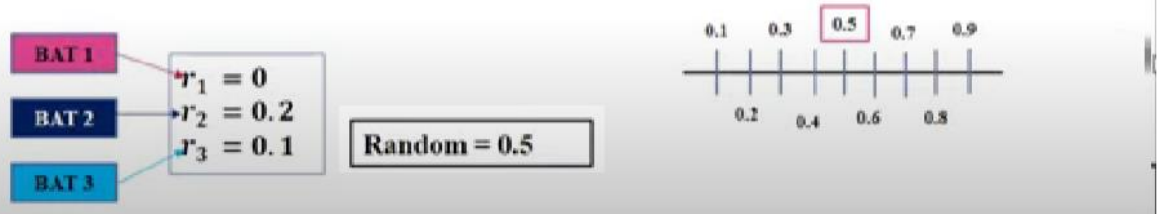
- Select a solution among the best solution.
- Generate local solution around best solution.

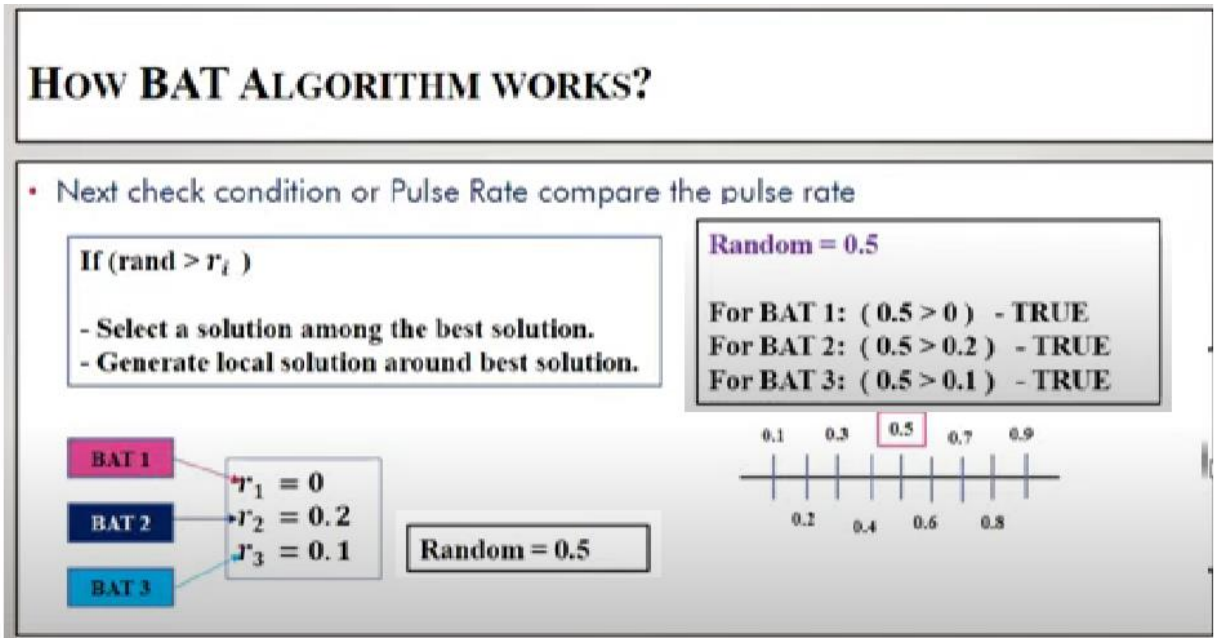
How BAT Algorithm works?

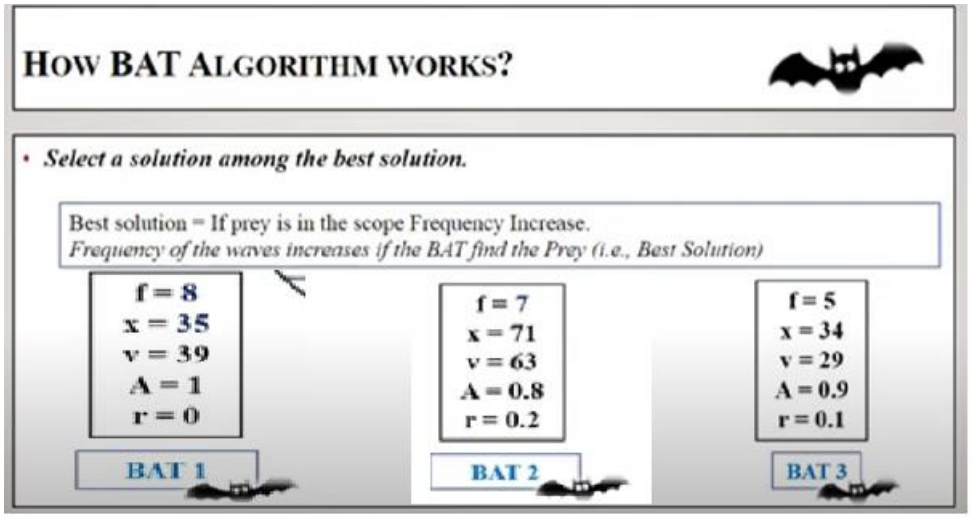
- Next check condition or Pulse Rate compare the pulse rate

If (rand > r_i)

- Select a solution among the best solution.
- Generate local solution around best solution.







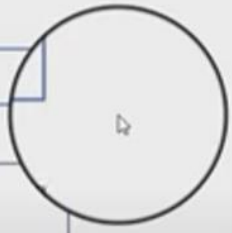
BAT-1 Best Solution

How BAT Algorithm works?

- Generate Local Solution Around Best Solution

$$x_{new} = x_{old} + \varepsilon A^t$$

$$x_{new} = x_{old} + \varepsilon A^1$$
$$x_{new} = 35 + 1.1$$
$$x_{new} = 36$$
$$\varepsilon = [-1, 1]$$



f = 8
x = 35
v = 39
A = 1
r = 0

BAT 1

t = 1

How BAT Algorithm works?

- The loudness decreases as the BAT moves closer to its prey and Pulse rate Emission Increases.

$$if\ (rand < A_i\ \&\ f(x_i) < f(x_*))$$

$$(0.5 < 1\ \ \&\ \ 0.2 < 5)$$

Accept the Solution.
Increase r_i *and Reduce* A_i

f = 8
x = 36
v = 39
A = 1
r = 0

BAT 1

How BAT Algorithm works?

- The loudness decreases as the BAT moves closer to its prey and Pulse rate Emission Increases.

Increase r_i *and Reduce* A_i

$$r_i^{t+1} = r_i^0 [1\text{-}e^{\gamma t}]$$

f = 8
x = 36
v = 39
A = 1
r = 0

BAT 1

$\alpha = \gamma = 0.9$

How BAT Algorithm works?

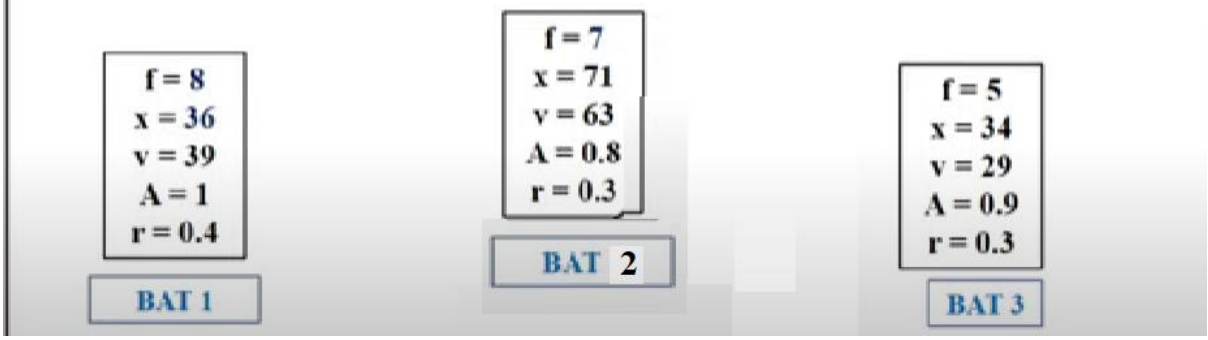
- RANK the BAT and Find Current BEST.

f = 8
x = 36
v = 39
A = 1
r = 0.4

BAT 1

How BAT Algorithm works?

- The loudness decreases as the BAT moves closer to its prey and Pulse rate Emission Increases.



Pros and Cons of BA

- The Bat Algorithm (BA) is accurate and very efficient algorithm to solve complex problems.
- Efficient to solve multi-stage, multi-machine, multi-product scheduling problems, and also the NP-hard problems
- The nature of automatic zooming, effective parameter control, the frequency turning and echolocation are great things to solve wide range of problems with quick time in promising optimal solution.
- The disadvantage of this algorithm is, it converge very quickly at early stage, and also convergence rate will slow down.
- In large scale applications the accuracy is limited, and no defined mathematical analysis to link the parameters with convergence rate.

Cuckoo search algorithm

•A method of global optimization based on the behavior of cuckoos was proposed by Yang & Deb (2009).

•The original "cuckoo search (CS) algorithm" is based on the idea of the following :-

➢ How cuckoos lay their eggs in the host nests.

➢How, if not detected and destroyed, the eggs are hatched to chicks by the hosts.

➢How a search algorithm based on such a scheme can be used to find the global optimum of a function.

Behavior of Cuckoo breeding

•The CS was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of host birds.

•Some cuckoos have evolved in such a way that female parasitic cuckoos can imitate the colors and patterns of the eggs of a few chosen host species.

•This reduces the probability of the eggs being abandoned and, therefore, increases their reproductivity .

•If host birds discover the eggs are not their own, they will either throw them away or simply abandon their nests and build new ones.

•Parasitic cuckoos often choose a nest where the host bird just laid its own eggs.

•In general, the cuckoo eggs hatch slightly earlier than their host eggs.

•Once the first cuckoo chick is hatched, his first instinct action is to evict the host eggs by blindly propelling the eggs out of the nest.

•This action results in increasing the cuckoo chick's share of food provided by its host bird .

•Moreover, studies show that a cuckoo chick can imitate the call of host chicks to gain access to more feeding opportunity.

Characteristics of Cuckoo search

•Each egg in a nest represents a solution, and a cuckoo egg represents a new solution.

•The aim is to employ the new and potentially better solutions (cuckoos) to replace not-so-good solutions in the nests.

• In the simplest form, each nest has one egg.

•The algorithm can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions

The CS is based on three idealized rules:

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest

- The best nests with high quality of eggs (solutions) will carry over to the next generations

- The number of available host nests is fixed, and a host can discover an alien egg with probability $p \in [0,1]$.

- In this case, the host bird can either throw the egg away or abandon the nest to build a completely new nest in a new location.

Lèvy Flights

- In nature, animals search for food in a random or quasi-random manner.

- Generally, the foraging path of an animal is effectively a random walk because the next move is based on both the current location/state and the transition probability to the next location.

- The chosen direction implicitly depends on a probability, which can be modeled mathematically.

•A Lévy flight is a random walk in which the step-lengths are distributed according to a heavy-tailed probability distribution.

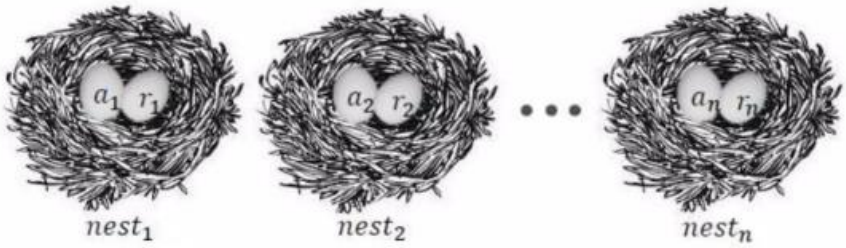
•After a large number of steps, the distance from the origin of the random walk tends to a stable distribution.

Algorithm 1 Cuckoo search algorithm

1: Set the initial value of the host nest size n, probability $p_a \in [0,1]$ and maximum number of iterations Max_{itr}.
2: Set $t := 0$. {Counter initialization}.
3: **for** $(i = 1 : i \leq n)$ **do**
4: Generate initial population of n host $x_i^{(t)}$. {n is the population size}.
5: Evaluate the fitness function $f(x_i^{(t)})$.
6: **end for**
7: **repeat**
8: Generate a new solution (Cuckoo) $x_i^{(t+1)}$ randomly by Lévy flight.
9: Evaluate the fitness function of a solution $x_i^{(t+1)}$ $f(x_i^{(t+1)})$
10: Choose a nest x_j among n solutions randomly.
11: **if** $(f(x_i^{(t+1)}) > f(x_j^{(t)}))$ **then**
12: Replace the solution x_j with the solution $x_i^{(t+1)}$
13: **end if**
14: Abandon a fraction p_a of worse nests.
15: Build new nests at new locations using Lévy flight a fraction p_a of worse nests
16: Keep the best solutions (nests with quality solutions)
17: Rank the solutions and find the current best solution
18: Set $t = t + 1$. {Iteration counter increasing}.
19: **until** $(t < Max_{itr})$. {Termination criteria satisfied}.
20: Produce the best solution.

The following steps describe the main concepts of Cuckoo search algorithm

Step1. Generate initial population of n host nests.



(ai,ri) : a candidate for optimal parameters

Step2. Lay the egg (ak',bk') in the k nest.

- K nest is randomly selected.
- Cuckoo's egg is very similar to host egg.

Where

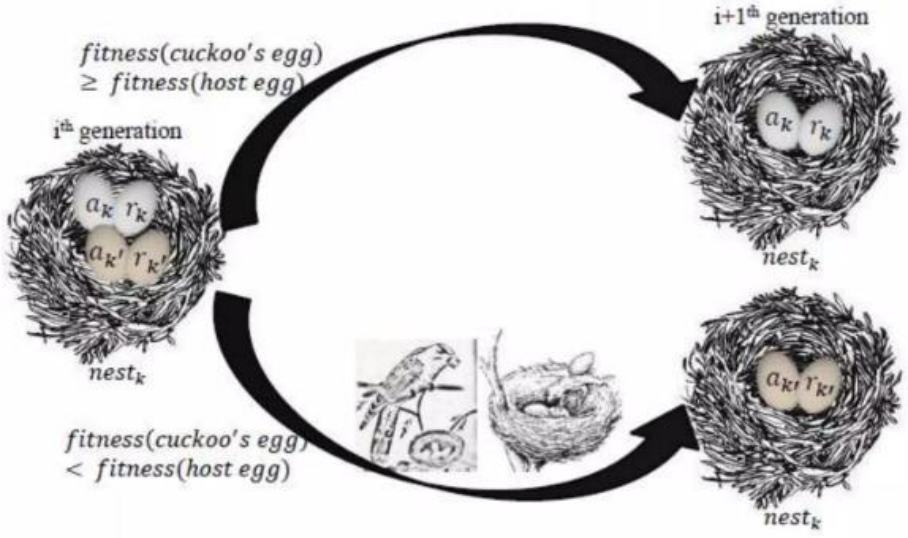
$ak'=ak+Randomwalk$(Lèvy flight)ak

$rk'=rk+Randomwalk$(Lèvy flight)rk

Step3. Compare the fitness of cuckoo's egg with the fitness of the host egg.

Root Mean Square Error (RMSE)

Step4. If the fitness of cuckoo's egg is better than host egg, replace the egg in nest k by cuckoo's egg.



Step5. If host bird notice it, the nest is abandoned and new one is built. (p <0.25) (to avoid local optimization)

Iterate **steps 2 to 5** until termination criterion satisfied

Application of the CS Algorithm

- Engineering optimization problems
- NP hard combinatorial optimization problems
- Data fusion in wireless sensor networks
- Nanoelectronic technology based operation-amplifier
- (OP-AMP)
- Train neural network
- Manufacturing scheduling