Complexity Analysis

Time Complexity

O(N × I)

- N = number of bats (candidate solutions)
- I = number of iterations

Each bat per iteration performs:

- Frequency and velocity update → O(1)
- Position update and boundary check → O(1)
- Fitness evaluation (Euclidean distance) → O(1)
- Acceptance check and possible local random walk → O(1)

So, the algorithm's total work scales linearly with both the number of bats and the number of iterations.

Space Complexity

O(N)

Each bat stores:

- Position (x, y)
- Velocity vector
- Frequency value
- Fitness (implicitly or explicitly)

Overall space required is linear in the number of bats.

Notes

- Efficient for real-time or embedded optimization in discrete environments like grid-based drone routing.
- Frequency tuning and loudness/pulse control provide a balance between exploration and exploitation.
- Can be extended with:
    - **Dynamic targets or obstacles**

    - **Time-window constraints**
    - **Energy-aware flight modeling**
- More stable convergence than pure random methods due to controlled exploration.