**1.Initialize Parameters:** Initialize the number of honey badgers (population size *N*) and their respective positions.  
Define maximum number of iterations   
Initialize parameters:

where *r1 is* a random number between 0 and 1, and  *are* the lower and upper bounds of the search domain

**2.Evaluate Initial Population:** Evaluate the fitness of each honey badger position  *using* the objective function and assign it to

Save the best position and assign its fitness to

**3. Main Loop (Repeat until termination criteria are met)**

Update the decreasing factor *α*

For each honey badger *i:*

**Calculate Intensity** :

where is a random number between 0 and 1.

**4. Position Update**:

If *r<0.5* (digging phase):

where *r3, r4, r5 are* random numbers between 0 and 1, *F* is a flag determined by

F=1 if r6 ≤0.5 otherwise F= -1 and r6 is a random number between 0 and 1

If *r≥0.5* (honey phase):

where *r7 is* a random number between 0 and 1.

**5. Evaluate New Position**: Evaluate the new position and assign its fitness to *fnew* .

If *fnew ≤ fi*

Set

If *fnew≤fprey*

Set

**6. Termination:** Stop if the maximum number of iterations is reached and return the best position and its fitness

The Honey Badger Algorithm (HBA) is a nature-inspired metaheuristic optimization algorithm. It mimics the intelligent foraging behavior of the honey badger, which is known for its fearless and strategic hunting techniques. The HBA algorithm operates through two main phases: the digging phase and the honey phase.

* **Digging Phase**: This phase emulates the badger's behavior of sniffing and digging to locate its prey. It allows for extensive exploration of the search space, enabling the algorithm to avoid local optima and explore various potential solutions.
* **Honey Phase**: In this phase, the badger follows the honey guide bird to directly locate beehives. This phase is more exploitative, focusing on intensifying the search around the best-known solutions to refine and improve them.

The HBA algorithm is characterized by its ability to balance exploration and exploitation dynamically through controlled randomization techniques and intensity calculations based on the distance to the prey. This balance ensures diversity in the population and enhances the algorithm's ability to converge towards global optima efficiently.

Reference:

<https://mealpy.readthedocs.io/en/latest/pages/models/mealpy.swarm_based.html#module-mealpy.swarm_based.HBA>