Section Week 10: Artificial Bee Colony (ABC)

Ahmed Métwalli

April 28, 2025

Artificial Bee Colony (ABC) Parameters

ABC is a metaheuristic optimization algorithm inspired by the foraging behavior of honey bees. Below are detailed explanations of each ABC parameter:

- **Colony Size:** Total number of bees (employed + onlooker + scout). More bees provide better exploration but increase computation.
- **Employed Bees:** Each bee is associated with a particular food source (solution) and searches its neighborhood.
- **Onlooker Bees:** Bees that select food sources based on shared information (better solutions have higher chances).
- **Scout Bees:** Bees that randomly search new food sources when an old source becomes abandoned.
- **Limit Parameter:** If a food source is not improved after a certain number of trials, it is abandoned.
- **Food Source Position (x_i):** Represents a candidate solution in the search space.
- **Fitness Function ($f(x_i)$):** Measures the quality of a food source (solution).

Aspect	**ABC**	**PSO**
Exploration	Strong (scouts search randomly)	Moderate (may get stuck early)
Exploitation	Good	Very strong
Parameters	Few, easy to tune	Needs careful tuning
Best for	Rugged, multimodal, discrete problems	Smooth, continuous problems
Speed	Slower	Faster

Table 1: Quick Comparison between ABC and PSO

Comparison of Bee Roles in ABC

Aspect	**Employed Bees**	**Onlooker Bees**	**Scout Bees**
Main Activity	Exploit assigned food sources and search nearby	Select food sources based on shared information and exploit them	Randomly search for new food sources
Location	Outside hive (exploring known food sources)	Inside hive (observing dances)	Outside hive (free exploration)
Selection Mechanism	Fixed assignment to a food source	Probability-based selection (better food sources more likely)	Random search without prior information
Role in Exploration-Exploitation Balance	Focused exploitation (local search)	Biased exploitation (towards best sources)	Pure exploration (diversity maintenance)
Trigger for Action	Regular update of assigned source	Based on quality information from employed bees	Triggered when food source limit is exceeded (abandonment)
Importance	Improve known solutions	Reinforce and refine promising areas	Discover entirely new solutions

Key Equations in ABC Algorithm

The Artificial Bee Colony (ABC) algorithm mainly uses three key operations based on equations:

1. Neighbourhood Search (Employed and Onlooker Bees)

Each employed or onlooker bee generates a new food source v_{ij} by modifying the current food source x_{ij} as:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

Where:

- x_{ij}: Current solution (food source) of bee i at dimension j.
- x_{kj}: Randomly selected neighbour's solution (different from i).
- $\phi_{ij} \in [-1, 1]$: Random number controlling the step size and direction.

Meaning: This formula perturbs the current solution slightly towards or away from a neighboring solution, allowing local exploration.

2. Fitness Calculation

The fitness value fit_i of a food source is calculated based on the objective function value $f(x_i)$:

$$fit_i = \begin{cases} \frac{1}{1+f(x_i)}, & \text{if } f(x_i) \geq 0 \\ 1 + |f(x_i)|, & \text{if } f(x_i) < 0 \end{cases}$$

Meaning: Higher fitness corresponds to better solutions. It ensures that fitness is always positive and inversely proportional to the objective function when $f(x) \geq 0$.

3. Probability of Selection (Onlooker Phase)

Each onlooker selects a food source based on a probability P_i proportional to its fitness:

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}$$

Where:

- fit_i: Fitness of the i-th food source.
- SN: Number of food sources (solutions).

Meaning: Better solutions have a higher chance of being selected by onlooker bees.

4. Scout Bee Update Rule

If a food source cannot be improved for a number of trials (greater than the `limit`), it is abandoned and a scout bee generates a new random solution:

$$x_{ij} = x_j^{\min} + rand(0, 1) \times (x_j^{\max} - x_j^{\min})$$

Where:

- $x_j^{\min}$ and $x_j^{\max}$: Lower and upper bounds of dimension j.
- $rand(0, 1)$: Random number between 0 and 1.

Meaning: Scout bees introduce randomness to explore completely new areas in the search space.

Summary: - **Exploration**: Achieved through scouts and random perturbations. - **Exploitation**: Achieved by employed and onlooker bees searching near good food sources. - **Balance**: Controlled by how often poor food sources are abandoned and replaced.

1 ABC vs PSO vs ACO: Combined Comparison

Feature / Stage	ABC (Bee Colony)	PSO (Particle Swarm)	ACO (Ant Colony)
Inspiration	Bee foraging and dancing	Bird flocking and fish schooling	Ant pheromone-based foraging
Representation	Food sources (solutions) and bee agents	Particles moving in continuous space	Ants traversing graph paths
Movement Rule	Random neighbor search around food sources	Update velocity based on personal/global best	Probabilistic transitions based on pheromone
Memory	Each bee remembers its food source	Each particle remembers personal and global best	Indirect memory in pheromone trails

Exploration-Exploitation Control	Scout bees for exploration, onlooker bees for exploitation	Inertia weight and random scalars	Pheromone evaporation and heuristic bias
Convergence Control	Limit on non-improving sources triggers exploration	Balancing inertia and coefficients	Pheromone levels regulate convergence
Application Domains	Scheduling, clustering, optimization	Neural networks, continuous optimization	Routing, scheduling, combinatorial problems
Initialization	Random food sources	Random positions and velocities	Random initial placement and pheromone
Communication Mechanism	Shared probability-based selection by onlookers	Sharing global best position	Indirect via pheromone deposition
Update Mechanism	New food source generation: $$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$ where k is a randomly selected neighbor and ϕ is a random number in $[-1, 1]$	$$v_i^{t+1} = wv_i^t + c_1 r_1(p_i^{best} - x_i^t)$$ $$+ c_2 r_2(g^{best} - x_i^t)$$	$$P_{ij} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum \tau_{ik}^{\alpha}\eta_{ik}^{\beta}}$$
Termination Condition	Maximum cycles or convergence	Maximum iterations or convergence	Maximum iterations or no improvement
End Comparison	Robust search, slow convergence	Fast convergence, sensitive to local optima	Good for discrete and combinatorial tasks

2 Symbol Notation

Symbol	Description
x_i	Position of the i-th food source (solution)
$f(x_i)$	Fitness value of the i-th food source
ϕ_{ij}	Random number in range $[-1, 1]$ used to perturb the solution
k	Randomly chosen neighbor index
$limit$	Maximum trials without improvement before abandoning food source

Table 4: Notation for Symbols in ABC

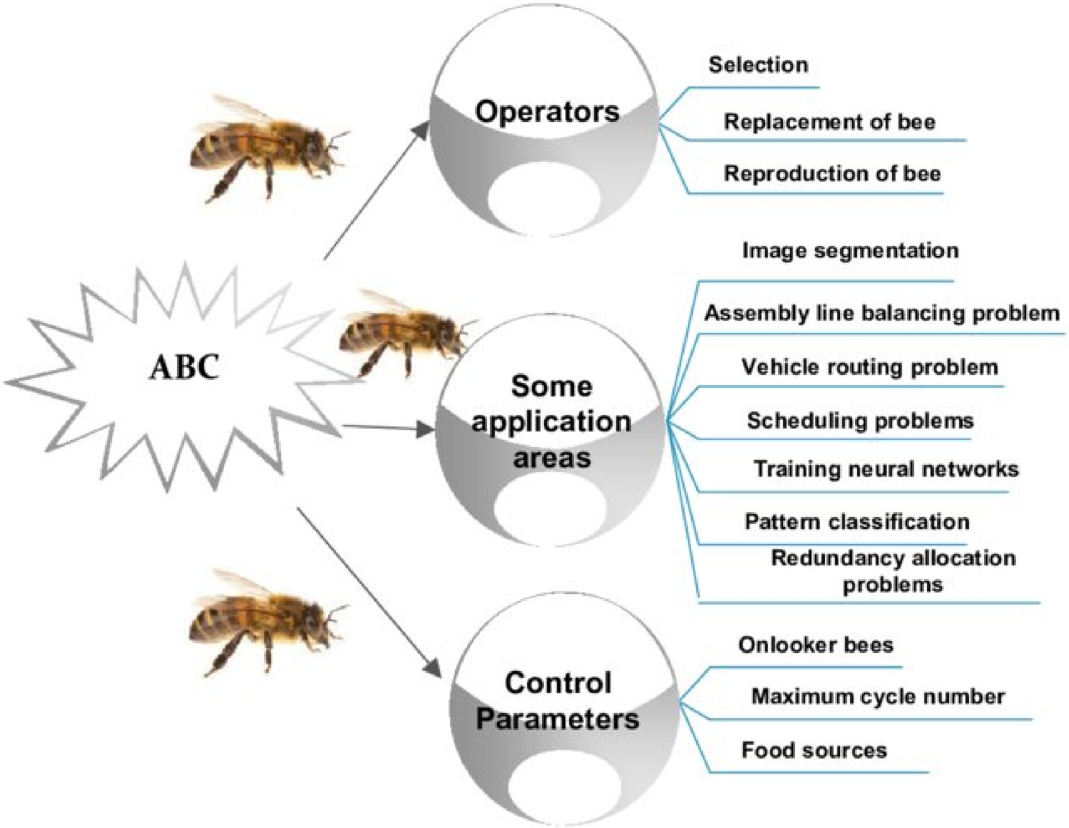

Figure 1: ABC

Worked Example: Three ABC Cycles on Rastrigin (2-D)

Problem. Minimise

$$f(\mathbf{x}) \;=\; 20 + \left[x_1^2 - 10\cos(2\pi x_1)\right] + \left[x_2^2 - 10\cos(2\pi x_2)\right], \qquad x_j \in [-5.12, 5.12].$$

Global optimum: $f(0,0) = 0$.

Assumptions Used Throughout

- **Food–source count** $(SN) = 2$. (*Comment: we keep the example tiny for hand calculation.*)
- **Dimensions** $D = 2$.
- **Bee roles:** SN employed + SN onlookers + 1 scout (implicit).
- **Limit parameter** = 3 trials. (*Comment: after 3 unsuccessful trials a source is abandoned.*)
- **Random numbers** ϕ, $rand(0,1)$ are sampled *once* per step and written explicitly.

Initialisation (cycle 0)

Food source	Position $\mathbf{x}$	$f(\mathbf{x})$	trials
1	[3.50, −2.00]	47.25	0
2	[−1.00, 4.20]	54.05	0

Comment: positions are drawn uniformly inside the box $[-5.12, 5.12]^2$. *Fitness is simply the Rastrigin value here because all values are positive, so* $fit_i \propto 1/f_i$.

Cycle 1

1. Employed-bee phase.

- **Source 1** picks neighbour $k = 2$ and random $\phi = +0.60$:

$$\mathbf{v}_1 = \mathbf{x}_1 + 0.60\,(\mathbf{x}_1 - \mathbf{x}_2) = [6.20,\ -5.72].$$

$f(\mathbf{v}_1) = 122.60 > 47.25 \;\Rightarrow$ *reject*, $\text{trials}_1 = 1$.

Comment: Eq. (Neighbourhood Search) with $\phi \in [-1, 1]$. *Worse fitness* $\Rightarrow$ *keep old source.*

- **Source 2** picks $k = 1,\ \phi = -0.30$:
$$\mathbf{v}_2 = \mathbf{x}_2 - 0.30(\mathbf{x}_2 - \mathbf{x}_1) = [0.35,\ 2.34].$$
$f(\mathbf{v}_2) = 18.92 < 54.05\ \Rightarrow$ *accept*, $\text{trials}_2 = 0$.
Comment: better fitness replaces the current food source.

2. Onlooker-bee phase. Compute selection probabilities using $fit_i \propto 1/f_i$:

$$P_1 = \frac{1/47.25}{1/47.25 + 1/18.92} = 0.29, \qquad P_2 = 0.71.$$

Comment: Only one onlooker for simplicity; we assume it chooses the more probable Source 2.
New neighbour with $\phi = +0.45,\ k = 1$:

$$\mathbf{v}_{2'} = \mathbf{x}_2 + 0.45(\mathbf{x}_2 - \mathbf{x}_1) = [-1.07,\ 4.29], \quad f = 54.35$$

— worse, so *reject*, $\text{trials}_2 = 1$.

3. Scout check. No trials counter reached 3 $\Rightarrow$ nothing happens.

$$\boxed{\mathbf{x}_1 = [3.50, -2.00],\ f = 47.25,\ \text{trials} = 1;\ \mathbf{x}_2 = [0.35, 2.34],\ f = 18.92,\ \text{trials} = 1.}$$

Cycle 2

1. Employed phase.

- **Source 1:** $k = 2,\ \phi = -0.20 \Rightarrow \mathbf{v}_1 = [2.87, -2.87],\ f = 21.90 < 47.25$ *accept*, $\text{trials}_1 = 0$.
  Comment: improvement resets trials counter.
- **Source 2:** $k = 1,\ \phi = +0.55 \Rightarrow \mathbf{v}_2 = [0.09, 2.63],\ f = 14.53 < 18.92$ *accept*, $\text{trials}_2 = 0$.

2. Onlooker phase. With new fitnesses, $P_1 = 0.40,\ P_2 = 0.60$.
Assume two onlookers (to show diversity):

- Onlooker 1 picks Source 2, $\phi = -0.5,\ k = 1 \Rightarrow \mathbf{v}_{2'} = [0.48, -0.12],\ f = 0.26$ *accept*, $\text{trials}_2 = 0$.
  Comment: huge improvement—near global optimum!
- Onlooker 2 again picks Source 2, $\phi = +0.4 \Rightarrow \mathbf{v}_{2''} = [-0.08, -1.31],\ f = 18.02$ (worse) *reject*, $\text{trials}_2 = 1$.

3. Scout check. Still below the limit.

$$\boxed{\mathbf{x}_1 = [2.87, -2.87],\ f = 21.90;\ \mathbf{x}_2 = [0.48, -0.12],\ f = 0.26.}$$

Cycle 3 (illustrating *limit* and Scout)

Comment: we now force *Source 1 to fail three times to hit the `limit`=3.*

- Three successive neighbourhood searches with $\phi \in \{+0.7, -0.6, +0.9\}$ all worsen $f \Rightarrow \text{trials}_1 = 3$.
- **Scout rule triggers**: Source 1 abandoned; new random position sampled:
$$x_{1j} = x_j^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min}),$$
giving $[-4.8,\ 4.6],\ f = 89.1$.
Comment: fresh exploration keeps diversity alive.

Source 2 continues to improve ($f \to 0.05\ldots$); colony will converge to $(0, 0)$.

Key take-aways

1. Employed bees = local improvement; onlookers = biased reuse of best sources.
2. Scout mechanism prevents stagnation via random re-starts.
3. Using $1/f(x)$ as fitness is fine when $f(x) \geq 0$ (here true for Rastrigin in the chosen box).
4. Even a *tiny* colony (only two sources!) can locate the global basin within two cycles thanks to the selection pressure.