Nature Inspired Soft Computing
(23CS3202)

CO – 4

- Particle Swarm Optimization (PSO),
- Ant Colony Optimization (ACO),
- Artificial Bee Colony (ABC) algorithms.

AIM OF THE SESSION

To familiarize students with the concepts of Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC) algorithms .

To make students apply above algorithms on a real world problem

INSTRUCTIONAL OBJECTIVES

This unit is designed to:

1. Demonstrate Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC) algorithms and its concepts.
2. Describe the nature and features of PSO , ACO and ABC Algorithms.
3. List out the techniques of PSO , ACO and ABC Algorithms.
4. Demonstrate the process of PSO , ACO and ABC Algorithms.

LEARNING OUTCOMES

At the end of this unit, you should be able to:

1. Define the functions of PSO , ACO and ABC Algorithms
2. Summarize the techniques used for building the PSO , ACO and ABC Algorithms.
3. Describe ways to build the PSO , ACO and ABC Algorithms.

Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO)

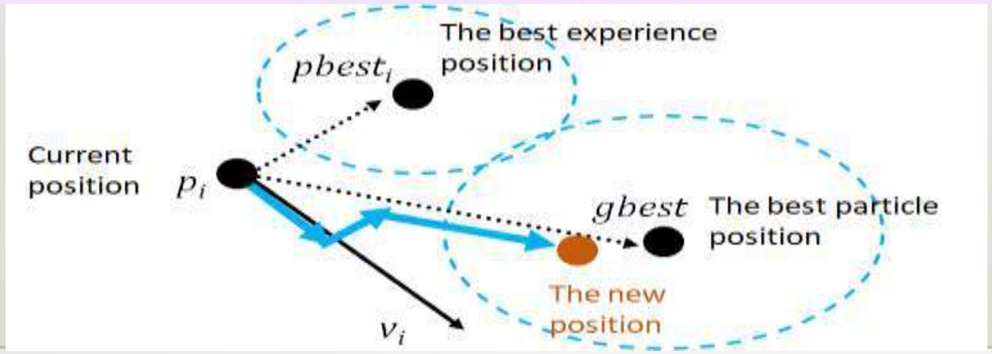
INTRODUCTION

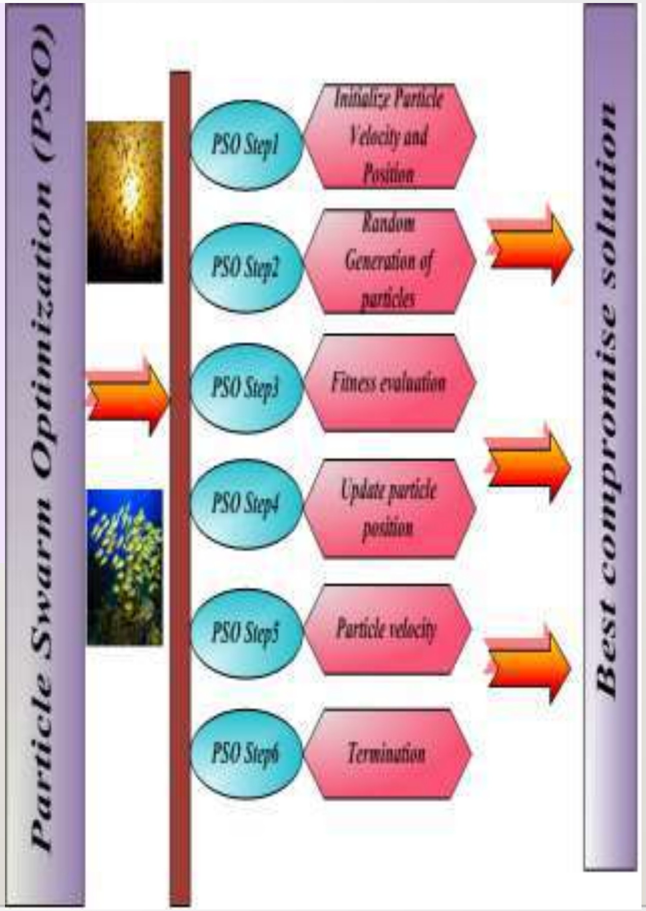
- Particle Swarm Optimization (PSO) is a population-based optimization technique inspired by the social behavior of birds and fish. It was developed by James Kennedy and Russell Eberhart in 1995 as an evolutionary algorithm to solve complex optimization problems. The main idea behind PSO is that a group of particles moves through a search space to find an optimal solution by updating their positions based on individual and collective experiences.
- PSO is widely used in engineering, machine learning, and scientific computing due to its simplicity, efficiency, and ability to handle non-differentiable, nonlinear, and multi-modal objective functions.

Particle Swarm Optimization (PSO)

ALGORITHM

- PSO optimizes a problem by iteratively improving candidate solutions. The swarm consists of particles (potential solutions), each having a position and velocity.
- The particles adjust their movement based on their own best-found position (personal best) and the best-found position of the entire swarm (global best).

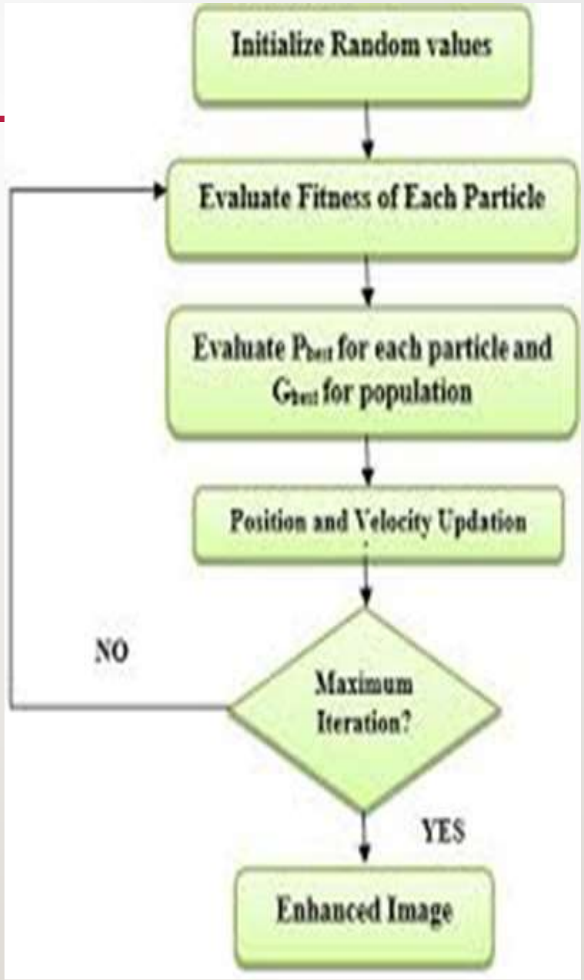




Particle Swarm Optimization (PSO)

STEPS OF PSO ALGORITHM

- 1. Initialize the swarm with random positions and velocities.
- 2. Evaluate the fitness of each particle using the objective function.
- 3. Update each particle's personal best if the new position is better.
- 4. Update the global best if any particle has found a better solution.
- 5. Update velocities and positions using the equations.
- 6. Repeat steps 2-5 until convergence or a stopping condition is met.



Particle Swarm Optimization (PSO)

MATHEMATICAL REPRESENTATION

Each particle in an N-dimensional space has:

- Position: $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$
- Velocity: $V_i = (v_{i1}, v_{i2}, ..., v_{iN})$
- Personal Best Position: $P_i = (p_{i1}, p_{i2}, ..., p_{iN})$
- Global Best Position: $G = (g_1, g_2, ..., g_N)$

The velocity update equation:

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot r_1 \cdot (p_{id} - x_{id}) + c_2 \cdot r_2 \cdot (g_d - x_{id})$$

where:

- w is the inertia weight
- c_1, c_2 are cognitive and social coefficients
- r_1, r_2 are random values in [0,1]

The position update equation:

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)$$

Particle Swarm Optimization (PSO)

TYPES OF PSO

1. Global Best PSO (Gbest-PSO)

All particles are influenced by the single best solution found.

Faster convergence but may get stuck in local minima.

2. Local Best PSO (Lbest-PSO)

Each particle is influenced by the best solution in its neighborhood.

Increases exploration and avoids premature convergence.

3. Adaptive PSO

Adjusts parameters dynamically to balance exploration and exploitation.

4. Multi-objective PSO (MOPSO)

Used for optimizing multiple conflicting objectives simultaneously.

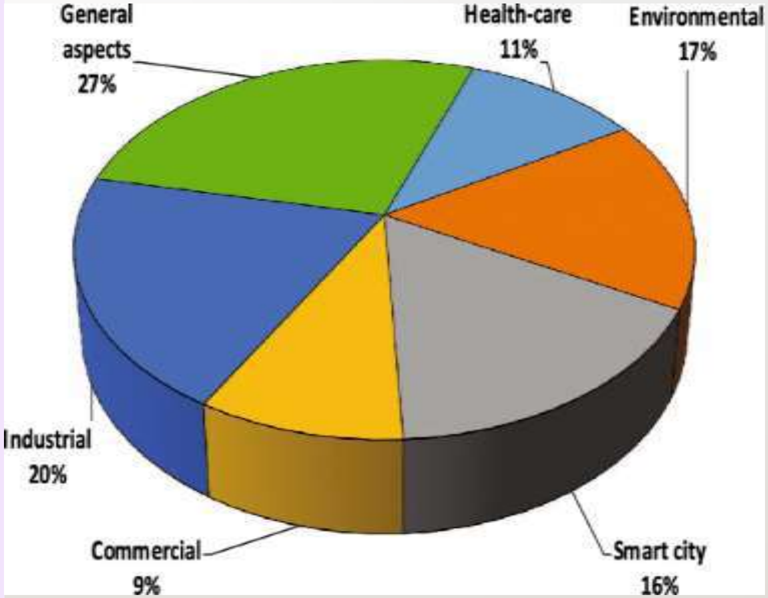
5. Hybrid PSO

Combines PSO with other optimization techniques such as Genetic Algorithms (GA) for better performance.

Particle Swarm Optimization (PSO)

APPLICATIONS OF PSO

- Engineering Optimization: Structural design, circuit design, scheduling problems.
- Machine Learning: Feature selection, hyperparameter tuning.
- Image Processing: Image segmentation, pattern recognition.
- Robotics: Path planning, swarm intelligence in autonomous robots.
- Finance: Portfolio optimization, risk management.
- Medical Diagnosis: Disease classification, bioinformatics.



Particle Swarm Optimization (PSO)

ADVANTAGES OF PSO

- Simple to implement with fewer parameters to tune.
- Works well in high-dimensional search spaces.
- Requires no gradient information.
- Provides fast convergence for many problems.

Particle Swarm Optimization (PSO)

CHALLENGES OF PSO

- ✓ May converge prematurely to local optima.
- ✓ Selection of parameters (inertia weight, cognitive and social coefficients) affects performance.
- ✓ Not ideal for dynamic optimization problems without modifications.
- ✓ Sensitive to initial conditions and randomness.

Particle Swarm Optimization (PSO)

REFERENCES

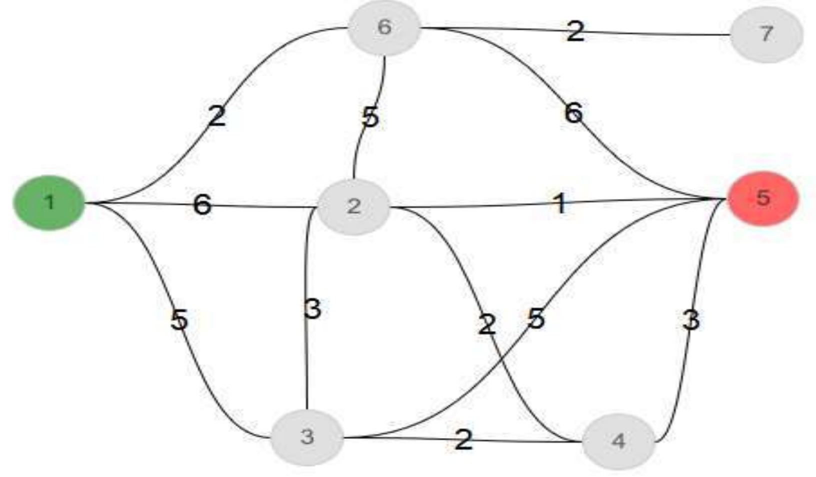
- *Kennedy, J., & Eberhart, R. (1995). "Particle swarm optimization." Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE.*
- *Shi, Y., & Eberhart, R. (1998). "A modified particle swarm optimizer." IEEE International Conference on Evolutionary Computation.*
- *Clerc, M., & Kennedy, J. (2002). "The particle swarm-explosion, stability, and convergence in a multidimensional complex space." IEEE Transactions on Evolutionary Computation.*
- *Engelbrecht, A. P. (2005). Fundamentals of Computational Swarm Intelligence. Wiley.*
- *Poli, R., Kennedy, J., & Blackwell, T. (2007). "Particle swarm optimization: An overview." Swarm Intelligence.*

Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO)

INTRODUCTION

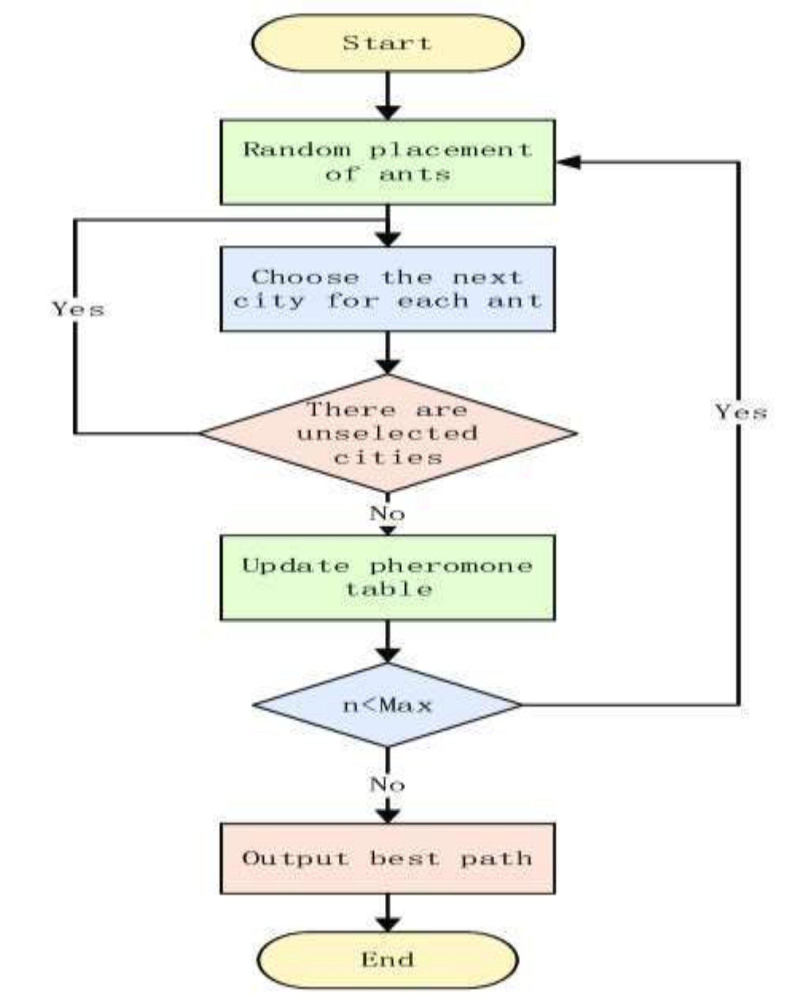
- **Ant Colony Optimization** (ACO) is a swarm intelligence-based optimization algorithm inspired by the foraging behavior of ants. It was introduced by Marco Dorigo in the early 1990s as a method to solve complex combinatorial optimization problems.
- The core idea is that ants deposit pheromones on paths while searching for food, and other ants follow these pheromone trails, reinforcing the best paths over time.
- ACO is widely used in routing, scheduling, and various optimization tasks where finding an optimal path or solution is required.



Ant Colony Optimization (ACO)

ALGORITHM

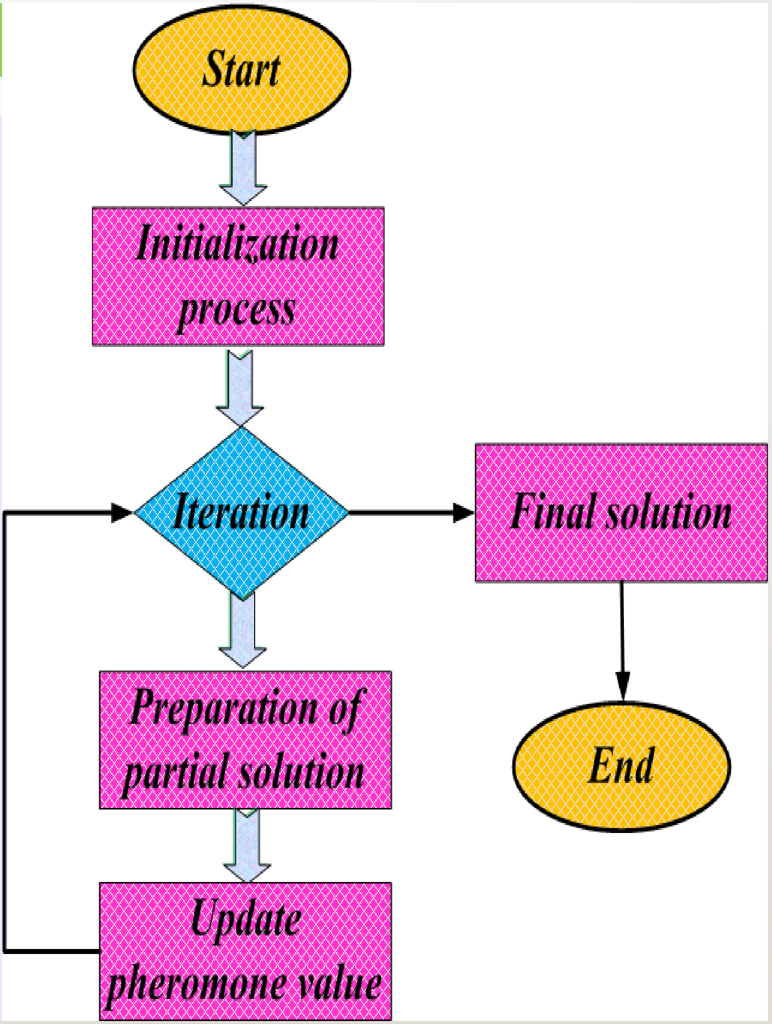
- ACO uses artificial ants to simulate the behavior of real ants in finding the shortest paths.
- Each ant constructs a solution by probabilistically choosing paths based on pheromone levels and problem-specific heuristics.
- Over iterations, pheromone updates guide the search toward optimal solutions.



Ant Colony Optimization (ACO)

STEPS OF ACO ALGORITHM

1. **Initialize** pheromone levels and algorithm parameters.
2. **Deploy ants**: Each ant starts from a random node.
3. **Solution Construction**: Ants probabilistically select the next node based on pheromone levels and heuristics.
4. **Pheromone Update**:
   - **Evaporation**: Reduce pheromone levels to avoid premature convergence.
   - **Deposition**: Reinforce paths used by successful ants.
5. **Check Stopping Condition**: Repeat until convergence or a stopping criterion is met.



Ant Colony Optimization (ACO)

MATHEMATICAL REPRESENTATION

Each solution is represented as a path in a graph $G = (V, E)$, where:

- V represents nodes (cities, locations, etc.)
- E represents edges (connections between nodes)
- τ_{ij} is the pheromone level on edge (i, j)
- η_{ij} is a heuristic value (e.g., inverse of distance)

The probability of choosing an edge (i, j) is given by:

$$P_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{k \in N_i}(\tau_{ik})^{\alpha}(\eta_{ik})^{\beta}}$$

where:

- α controls the influence of pheromone trails
- β controls the influence of heuristic information
- N_i is the set of feasible neighbors of node i

Ant Colony Optimization (ACO)

TYPES OF ACO

- **1. Ant System (AS)**

  The original ACO algorithm where all ants contribute to pheromone updates.

- **2. Elitist Ant System (EAS)**

  The best-performing solution is given extra pheromone reinforcement.

- **3. Rank-based Ant System (RAS)**

  Only top-ranking ants contribute to pheromone updates.

- **4. Max-Min Ant System (MMAS)**

  Limits pheromone levels to avoid early convergence.

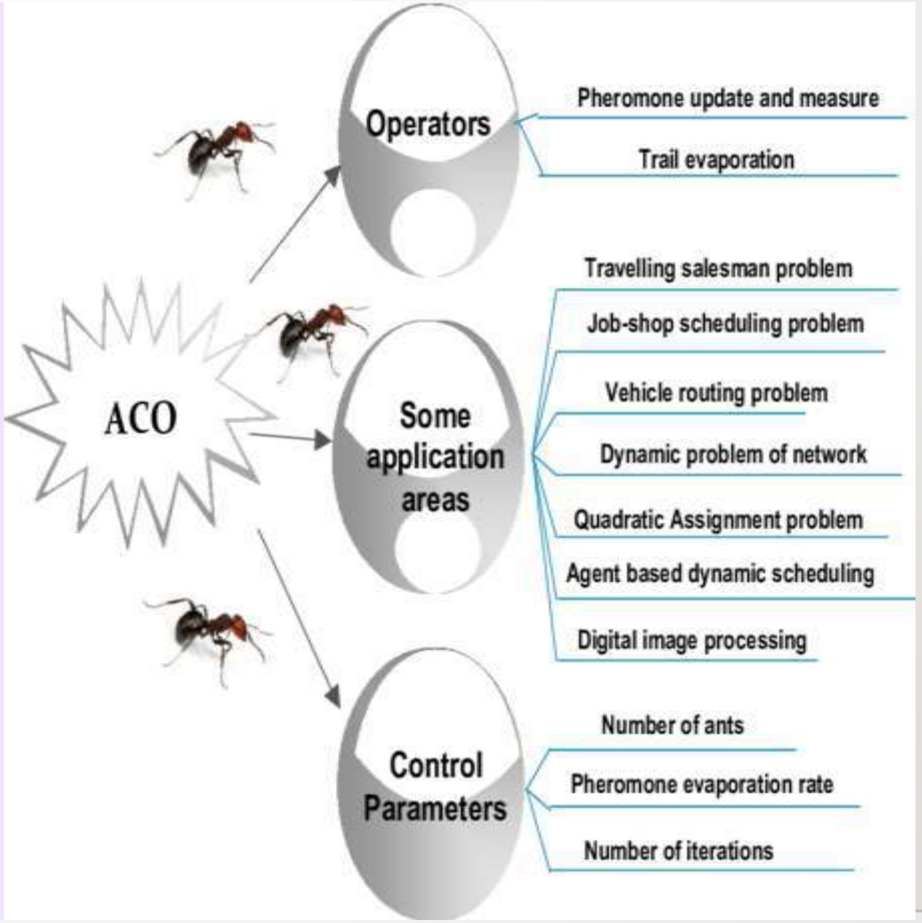
- **5. Ant Colony System (ACS)**

  Introduces local pheromone updates and stronger exploitation mechanisms.

Ant Colony Optimization (ACO)

APPLICATIONS OF ACO

- **Traveling Salesman Problem (TSP):** Finding the shortest route for a salesperson.
- **Network Routing:** Optimizing data transmission paths in networks.
- **Scheduling Problems:** Task scheduling in manufacturing and cloud computing.
- **Robotics:** Path planning for autonomous robots.
- **Bioinformatics:** Protein structure prediction and gene sequencing.
- **Logistics & Transportation:** Vehicle routing, supply chain optimization.



Ant Colony Optimization (ACO)

ADVANTAGES OF ACO

- Suitable for complex combinatorial problems.
- Adapts dynamically to changing environments.
- Provides robust and near-optimal solutions.
- Can be combined with other optimization techniques.

Ant Colony Optimization (ACO)

CHALLENGES OF ACO

- High computational cost for large problems.
- Sensitive to parameter tuning ().
- Can converge prematurely to suboptimal solutions.
- Slower convergence compared to some heuristic methods.

Ant Colony Optimization (ACO)

ADVANTAGES & DISADVANTAGES OF ACO

COMPARISON AMONG ACO-BASED ROUTING ALGORITHMS

Algorithm	Category	Advantages	Disadvantages
AntNet [113]	Proactive & Multipath & One Ant Colony	1) Robust multipath routing; 2) Automatic load balancing; 3) Adaptivity	Significant overload due to repeated path sampling
AntHocNet [114]	Proactive & Reactive & Multipath & One Ant Colony	1) Overload in AntHoc due to repeat path sampling is avoided; 2) Better delivery ratio	A high average delay is observed in the simple scenarios
MACO [129]	Multiple Ant Colony	1) Congestion in the optimal path is relieved; 2) High load balancing	Initial path selection may impact its convergence
BeeHive [84]	Honey Bee Colony	1) Load balancing enabled; 2) Quality paths being created	Routing information exchange imposes an additional traffic burden
MABC [131]	Multiple Ant-Bee Colonies	1) A better Fault-tolerance capability than MACO; 2) High load balancing	Delay caused by using multiple colonies at the beginning of failure

Ant Colony Optimization (ACO)

REFERENCES

- *Dorigo, M., Maniezzo, V., & Colorni, A. (1996). "Ant system: Optimization by a colony of cooperating agents." IEEE Transactions on Systems, Man, and Cybernetics.*
- *Dorigo, M., & Stützle, T. (2004). Ant Colony Optimization. MIT Press.*
- *Stützle, T., & Hoos, H. H. (2000). "Max-Min Ant System." Future Generation Computer Systems.*
- *Blum, C. (2005). "Ant colony optimization: Introduction and recent trends." Physics of Life Reviews.*
- *Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press.*

Artificial Bee Colony (ABC) algorithms.

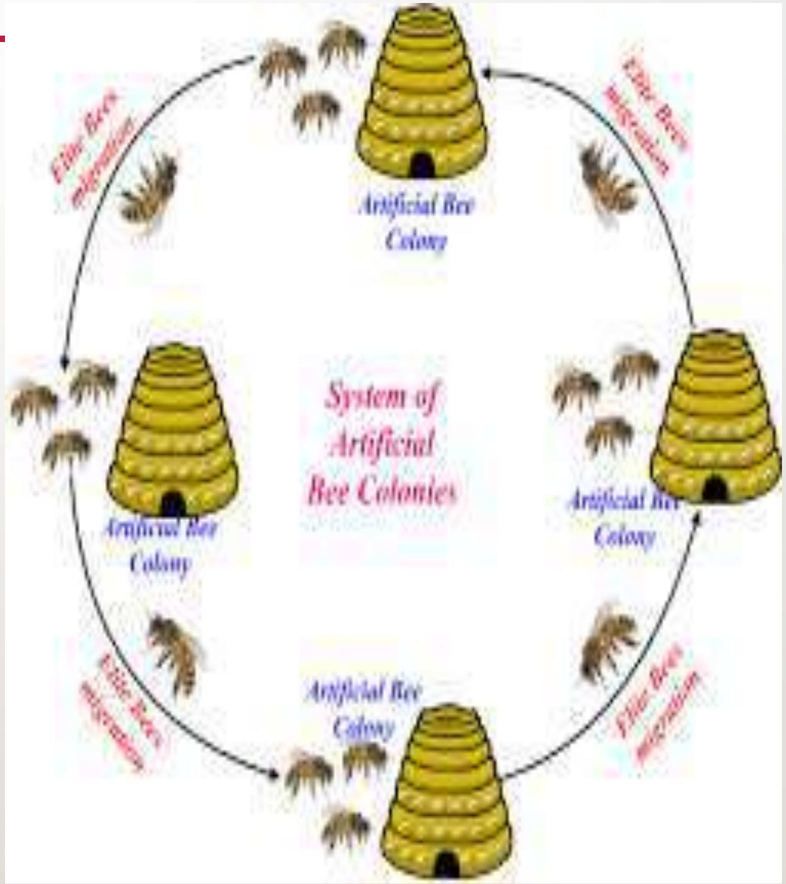
Artificial Bee Colony (ABC) algorithms.

INTRODUCTION

- The Artificial Bee Colony (ABC) algorithm is a nature-inspired optimization algorithm that simulates the food-searching behavior of honey bees. It was proposed by Karaboga in 2005 and belongs to the class of swarm intelligence algorithms.
- The ABC algorithm is a population-based optimization technique, where a population of artificial bees (agents) search for the optimal solution in the search space. The algorithm mimics the process by which bees search for food sources and communicate with each other to share information about the quality and location of food sources.
- The ABC algorithm is known for its simplicity, robustness, and efficiency in solving complex optimization problems, including both continuous and discrete optimization tasks.

Artificial Bee Colony (ABC) algorithms.

ALGORITHM



Artificial Bee Colony (ABC) algorithms.

KEY FEATURES OF THE ABC ALGORITHM:

Population-based: The algorithm works with a set of individuals (bees) that collectively explore the solution space.

Exploration and Exploitation: It balances exploration (searching new areas of the solution space) and exploitation (refining solutions near known good areas).

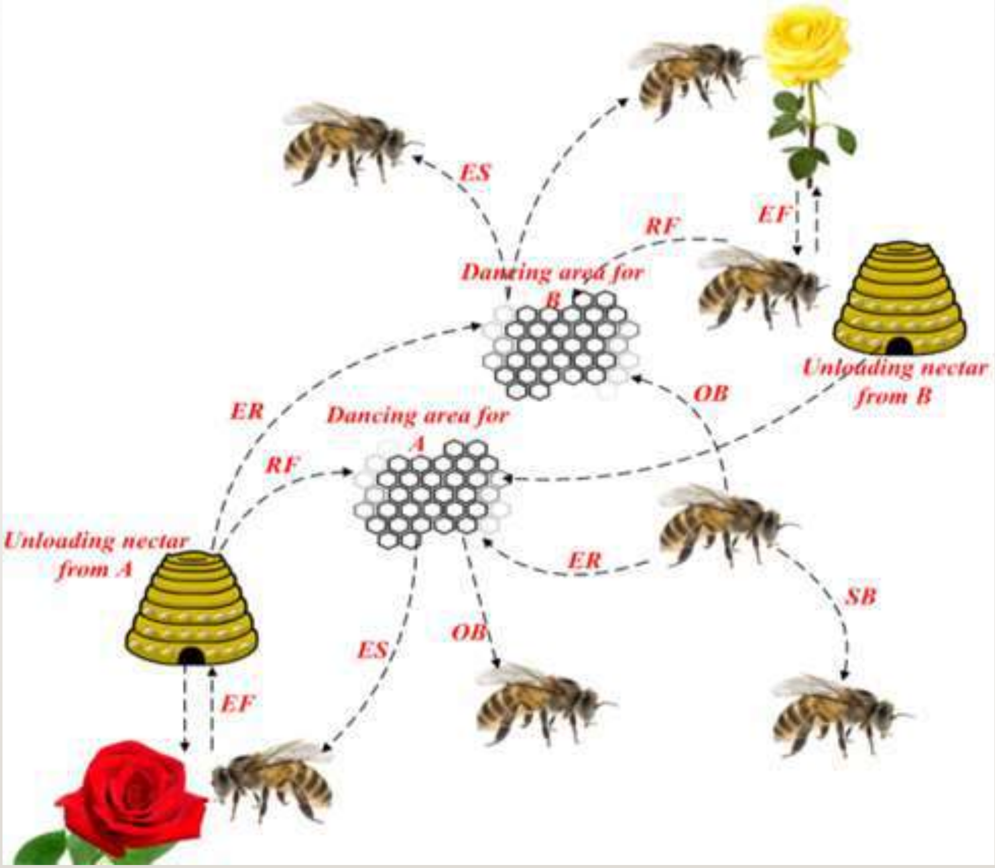
Self-adaptation: Bees can adapt their behavior based on the quality of food sources (solutions).

Global search: It does not require any prior knowledge about the problem's structure, making it suitable for complex, multi-dimensional, and non-linear optimization problems.

Artificial Bee Colony (ABC) algorithms.

WORKING PRINCIPLE OF THE ABC ALGORITHM

The ABC algorithm simulates the foraging behavior of bees in a search space to find the optimal solution to a given problem. The basic steps involved in the algorithm are:



Artificial Bee Colony (ABC) algorithms.

TYPES OF BEES IN THE ALGORITHM:

- ✓ Employed Bees: These are bees that are responsible for searching the food sources and storing the information.
- ✓ Onlooker Bees: These bees observe the dances of employed bees to gather information and select food sources to exploit based on the probability of their quality.
- ✓ Scout Bees: These bees search for new, unexplored regions of the solution space to discover new food sources.

Artificial Bee Colony (ABC) algorithms.

BASIC CONCEPT OF SEARCH:

- Food Source: A potential solution in the problem space.
- Fitness Value: The quality of a food source, which is determined by an objective function. In optimization, this could be the value that needs to be minimized or maximized.
- Neighborhood Search: Bees explore the neighborhood of the current food source to improve it.

Artificial Bee Colony (ABC) algorithms.

ALGORITHM FLOW:

- Initialization: Initialize a population of food sources (solutions), each corresponding to a potential solution in the search space.
- Employed Bee Phase: Each employed bee explores a food source and evaluates its fitness. If a better food source is found (improvement in fitness), it updates the food source and stores the new position.
- Onlooker Bee Phase: Onlooker bees select a food source from the available ones, based on the quality (fitness) of the food sources. The probability of selecting a food source depends on its fitness.
- Scout Bee Phase: If a food source has not been improved by any bee for a certain number of iterations, it is abandoned. A scout bee will randomly search the solution space for new food sources.
- Termination: The algorithm terminates when a stopping criterion (such as maximum iterations or convergence) is met.

Artificial Bee Colony (ABC) algorithms.

STEPS OF THE ABC ALGORITHM

Step 1: Initialization

Define the optimization problem and initialize the population of food sources (solutions).

Each food source is represented by a solution vector, and its fitness value is calculated.

Employed bees are assigned to these food sources.

Step 2: Employed Bee Phase

Each employed bee explores the solution space around its current food source by modifying the solution slightly.

It then evaluates the fitness of the new solution and compares it with the old one.

If the new solution is better (in terms of fitness), the bee updates its position to the new solution.

Step 3: Onlooker Bee Phase

Onlooker bees observe the employed bees' dances to determine which food sources to explore.

The probability of selecting a food source is proportional to the fitness of the food source.

Each onlooker bee chooses a food source and explores it further, just like employed bees.

Step 4: Scout Bee Phase

If a food source has not been improved for a certain number of iterations, it is abandoned.

A scout bee will search randomly for a new food source in the solution space.

Step 5: Termination

The algorithm terminates if a predefined stopping criterion is met, such as reaching a maximum number of iterations or finding a satisfactory solution.

Artificial Bee Colony (ABC) algorithms.

TYPES OF ARTIFICIAL BEE COLONY ALGORITHMS

The basic ABC algorithm can be adapted and modified to suit various types of optimization problems. Some of the notable variations include:

1. **Basic ABC Algorithm:** The standard ABC algorithm as described above.

2. **Improved ABC Algorithm:** Modifications are made to the basic ABC algorithm, such as introducing local search strategies or hybridizing with other algorithms like genetic algorithms or particle swarm optimization.

3. **Parallel ABC Algorithm:** In this variation, multiple populations of bees work in parallel, and the results are combined to improve the search efficiency.

4. **Discrete ABC Algorithm:** This version is designed to solve combinatorial optimization problems, where the solution space consists of discrete values, such as the traveling salesman problem.

5. **Multi-objective ABC Algorithm:** The algorithm is modified to handle problems with multiple conflicting objectives.

Artificial Bee Colony (ABC) algorithms.

APPLICATIONS OF ABC ALGORITHM

The ABC algorithm has been successfully applied to a wide range of optimization problems across various fields. Some of the notable applications include:

- **Optimization Problems:** The ABC algorithm is used in continuous optimization problems like function optimization, parameter tuning, and machine learning model training.
- **Engineering Design:** In fields like structural design, mechanical engineering, and electrical engineering, ABC has been applied to optimize designs, such as optimizing the shape of a wing or the configuration of a circuit.
- **Feature Selection:** ABC is used to optimize the selection of features in machine learning models, improving the performance by reducing the dimensionality of the input data.
- **Image Processing**: The algorithm has been applied to image segmentation, image denoising, and other computer vision tasks.
- **Routing Problems:** The ABC algorithm is used to solve various routing problems, including the Traveling Salesman Problem (TSP) and Vehicle Routing Problems (VRP).
- **Robotics:** ABC has been employed in optimizing the path planning and trajectory generation for robots.
- **Data Clustering:** ABC has been used to optimize clustering techniques, such as k-means or fuzzy c-means, to find the best clustering centers.

Artificial Bee Colony (ABC) algorithms.

ADVANTAGES OF THE ABC ALGORITHM

Simplicity: The algorithm is simple to implement and requires minimal parameters for configuration.

Global Search: ABC performs a global search of the solution space, which helps in avoiding local minima in optimization problems.

Efficiency: The algorithm is efficient in solving optimization problems with large search spaces.

Robustness: ABC is robust and can handle noisy and complex problem environments effectively.

Versatility: It can be applied to a wide range of optimization problems, including continuous, combinatorial, and multi-objective problems.

Artificial Bee Colony (ABC) algorithms.

CHALLENGES AND LIMITATIONS

Premature Convergence: Like many other swarm intelligence algorithms, the ABC algorithm can suffer from premature convergence, where the population of bees converges to suboptimal solutions early in the search process.

Parameter Sensitivity: The performance of ABC is sensitive to the choice of parameters such as the number of bees, the number of iterations, and the limit on the number of unimproved solutions.

High Computational Cost: In problems with very large search spaces or very complex fitness functions, the algorithm can be computationally expensive.

Exploration-Exploitation Trade-off: Balancing exploration and exploitation is critical. If exploration is overemphasized, the algorithm may fail to refine the solutions, while excessive exploitation may lead to premature convergence.

Artificial Bee Colony (ABC) algorithms.

CONCLUSION

The Artificial Bee Colony (ABC) algorithm is a powerful optimization tool inspired by nature's honey bee foraging behavior. It has shown great promise in solving various complex optimization problems across different fields. Despite its simplicity and robustness, there are challenges related to parameter tuning, premature convergence, and computational cost, but ongoing research into hybrid approaches and modifications is addressing these issues. ABC continues to be an active area of research due to its flexibility and efficacy in finding solutions to real-world problems.

Artificial Bee Colony (ABC) algorithms.

REFERENCES

- *Karaboga, D. (2005). "An Idea Based on Honeybee Swarm for Numerical Optimization." Technical Report-TR06, Erciyes University, Engineering Faculty.*
- *Karaboga, D., & Basturk, B. (2007). "On the Performance of Artificial Bee Colony (ABC) Algorithm." Proceedings of the IEEE Swarm Intelligence Symposium, 2007.*
- *Kennedy, J., & Eberhart, R. (1995). "Particle Swarm Optimization." Proceedings of the IEEE International Conference on Neural Networks, 1995.*
- *Cui, Z., & Zhang, W. (2014). "Modified Artificial Bee Colony Algorithm for Continuous Optimization." Procedia Computer Science, 30, 1197–1205.*
- *Xue, Y., & Zhang, L. (2015). "Artificial Bee Colony Algorithm for Optimization Problems." Springer Science & Business Media.*

Terminal Questions .

1. Apply the PSO algorithm to optimize a simple mathematical function, such as the Rosenbrock function. Describe the steps involved in the algorithm and discuss its convergence.
2. Using PSO, solve an optimization problem with constraints (e.g., a resource allocation problem). How does the algorithm handle constraint satisfaction, and how do you implement it?
3. Implement the PSO algorithm to optimize a machine learning model's hyperparameters (e.g., support vector machine parameters). Discuss the impact of PSO optimization on the model's performance.
4. Describe how you would modify the standard PSO algorithm to improve its exploration capabilities. Provide examples of how this would help in solving complex optimization problems.
5. Apply the Ant Colony Optimization (ACO) algorithm to solve the Traveling Salesman Problem (TSP). Explain how ants build their paths and how pheromone updates affect the algorithm's convergence.
6. Demonstrate the use of ACO in solving a scheduling problem. Explain how pheromone updates can guide the algorithm to find an optimal or near-optimal schedule.

Terminal Questions .

7. Using ACO, optimize a network routing problem. Discuss how the algorithm adapts to dynamic changes in the network and finds the shortest paths.
8. How would you modify the standard ACO algorithm to handle multi-objective optimization problems, such as optimizing both cost and time in a transportation problem? Provide an example of how this would work.
9. Implement the Artificial Bee Colony (ABC) algorithm to solve a function optimization problem. Discuss the different phases of the algorithm (employed bees, onlooker bees, and scout bees) and their role in finding the optimal solution.
10. Apply the ABC algorithm to a real-world problem, such as optimizing the design of an electrical circuit or structural optimization. Explain how the algorithm can be adapted for this application.
11. Compare the performance of ABC with PSO in terms of convergence rate and solution quality when solving a complex optimization problem, such as the parameter optimization for a neural network.
12. Discuss the impact of the number of bees (population size) and the number of iterations on the performance of the ABC algorithm in solving optimization problems. How would you determine the optimal values for these parameters?

THANK YOU