Exploring Nature-Inspired Algorithms

Ant Colony Optimization Algorithm

Understanding Ant Behavior

Inspired by nature's efficiency

Our algorithm mimics **real ant foraging** behavior to solve complex optimization problems effectively and efficiently.

Key Methodology of ACO

01 Pheromone Trails

Pheromone trails guide ants in finding **optimal paths** to food sources, influencing decision-making in the colony.

02 Heuristic Information

Heuristic information helps ants evaluate paths based on **experience**, improving the efficiency of the search process.

03 Pheromone Evaporation

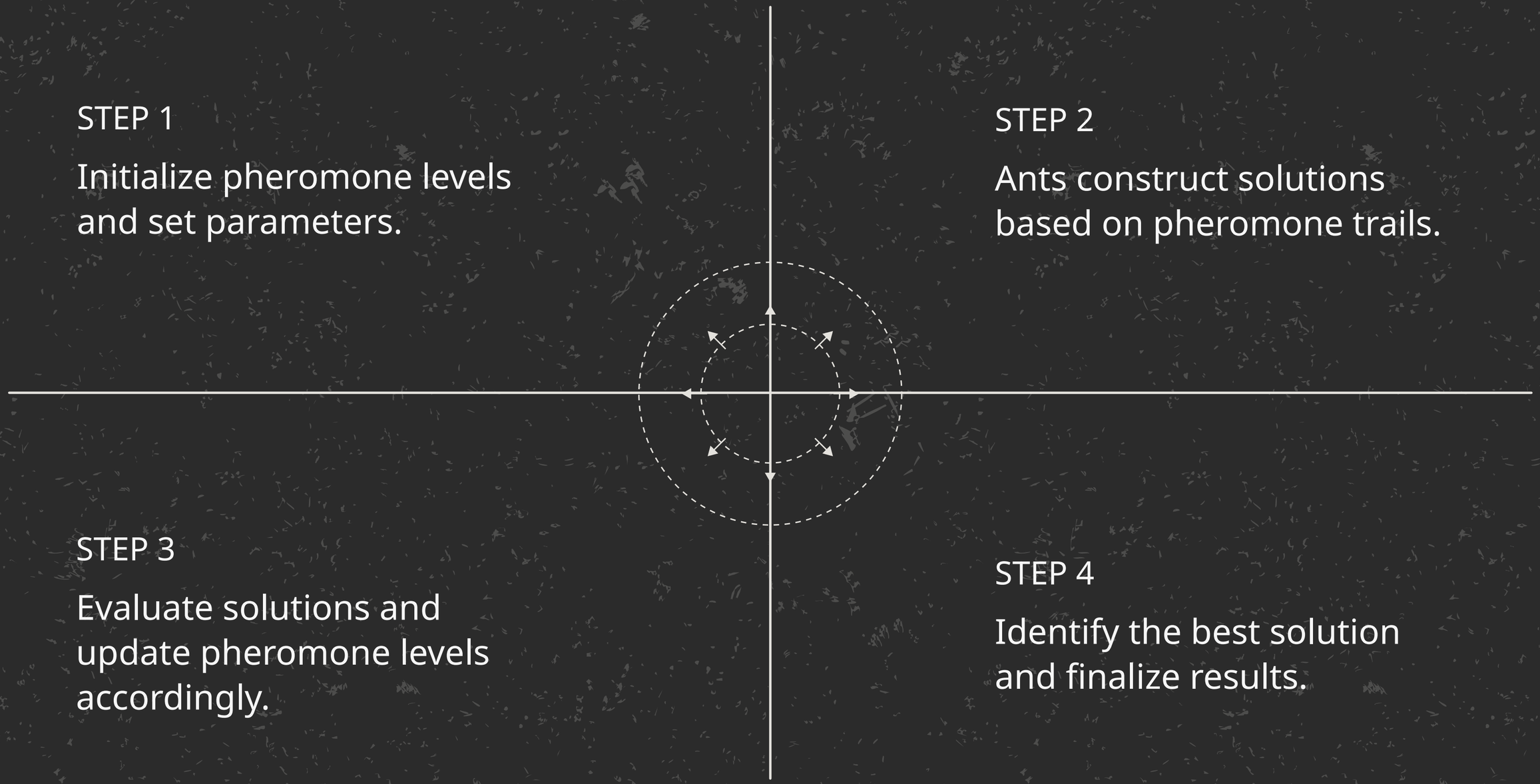
Pheromone Evaporation helps ants avoiding **long paths** being selected or getting stuck at **locally optimal solution**

STEP 1

Initialize pheromone levels and set parameters.

STEP 2

Ants construct solutions based on pheromone trails.



STEP 3

Evaluate solutions and update pheromone levels accordingly.

STEP 4

Identify the best solution and finalize results.

Used Mathimatical Equations

$$p_{ij} = \frac{(\tau_{ij}^{\alpha}) \cdot (\eta_{ij}^{\beta})}{\sum_{j \in \Lambda} (\tau_{ij}^{\alpha}) \cdot (\eta_{ij}^{\beta})}$$

Choosing next city using probability

τ: The amount of pheromone deposited on the edge

η: The travel cost (heuristic) of the edge

α *β*: Are parameters that control the influence of pheromone and heuristic information

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k}^{m} \Delta\tau_{ij}^{k}$$

Updating Pheromone

ρ: The rate of pheromone evaporation

$$\Delta\tau_{xy}^{k} = \begin{cases} Q/L_k \\ 0 \end{cases}$$

If ant K took the curve xy

Otherwise

Algorithm Analysis

Average runtime for 10 Cities: 2.88 seconds
Runtime breakdown by ant count:
1 Ants: 2.74 seconds
5 Ants: 3.34 seconds
10 Ants: 2.55 seconds
Average runtime for 20 Cities: 2.27 seconds
Runtime breakdown by ant count:
1 Ants: 2.24 seconds
5 Ants: 2.04 seconds
10 Ants: 1.99 seconds
20 Ants: 2.78 seconds

Best result for 10 cities: 152.00
Best result for 20 cities: 146.00

Conclusion

- ACO is a metaheuristic approach for solving NP hard optimization problems

- ACO efficiently solves TSP by reaching a near-optimal solution

- It has been successfully applied to a wide range of optimization problems, including scheduling, routing, and resource allocation