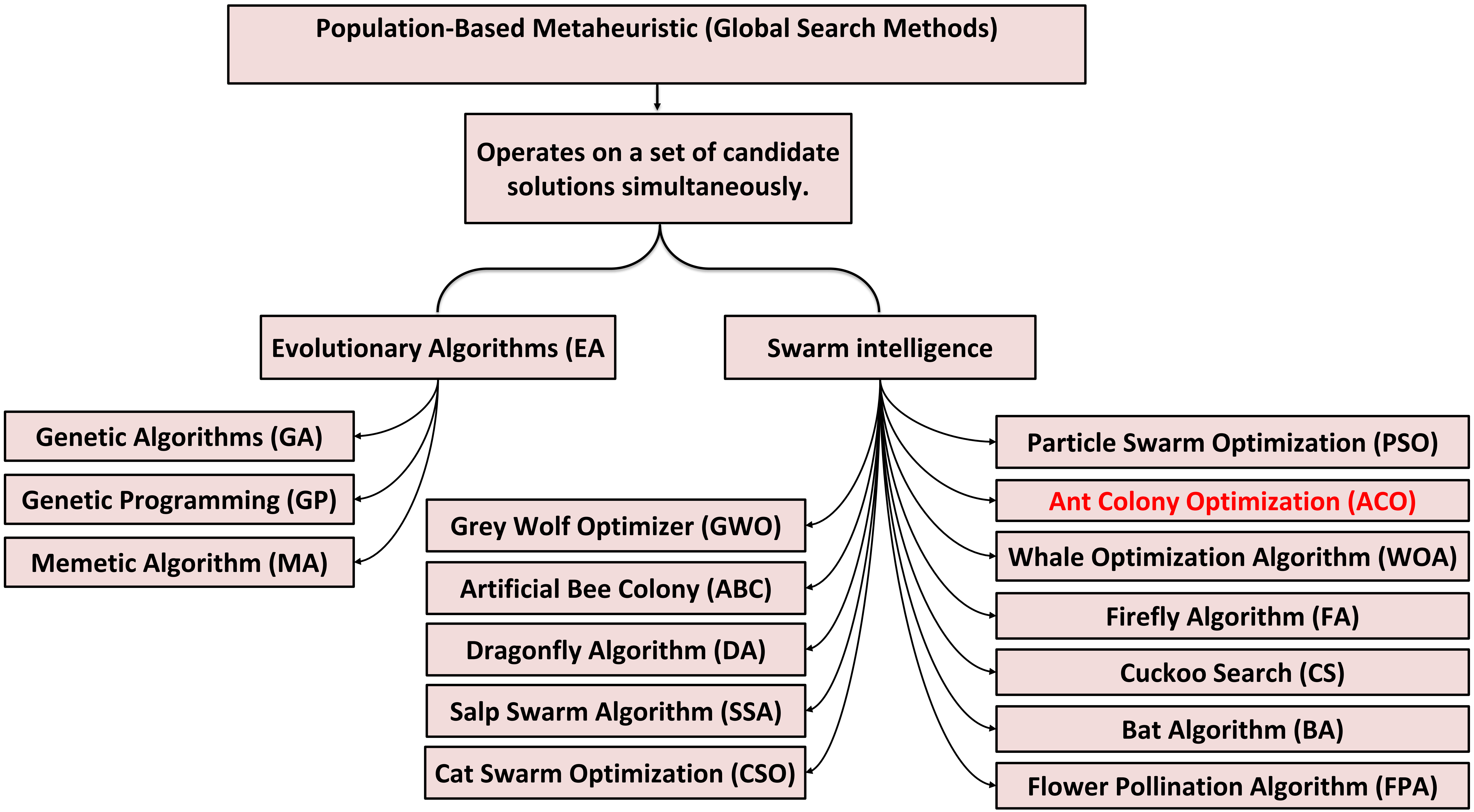
Nature Inspired Computation
DSAI 403

Assoc. Prof. **Mohamed Maher Ata**
Zewail city of science, technology, and innovation
momaher@zewailcity.edu.eg
Room: S028- Zone D

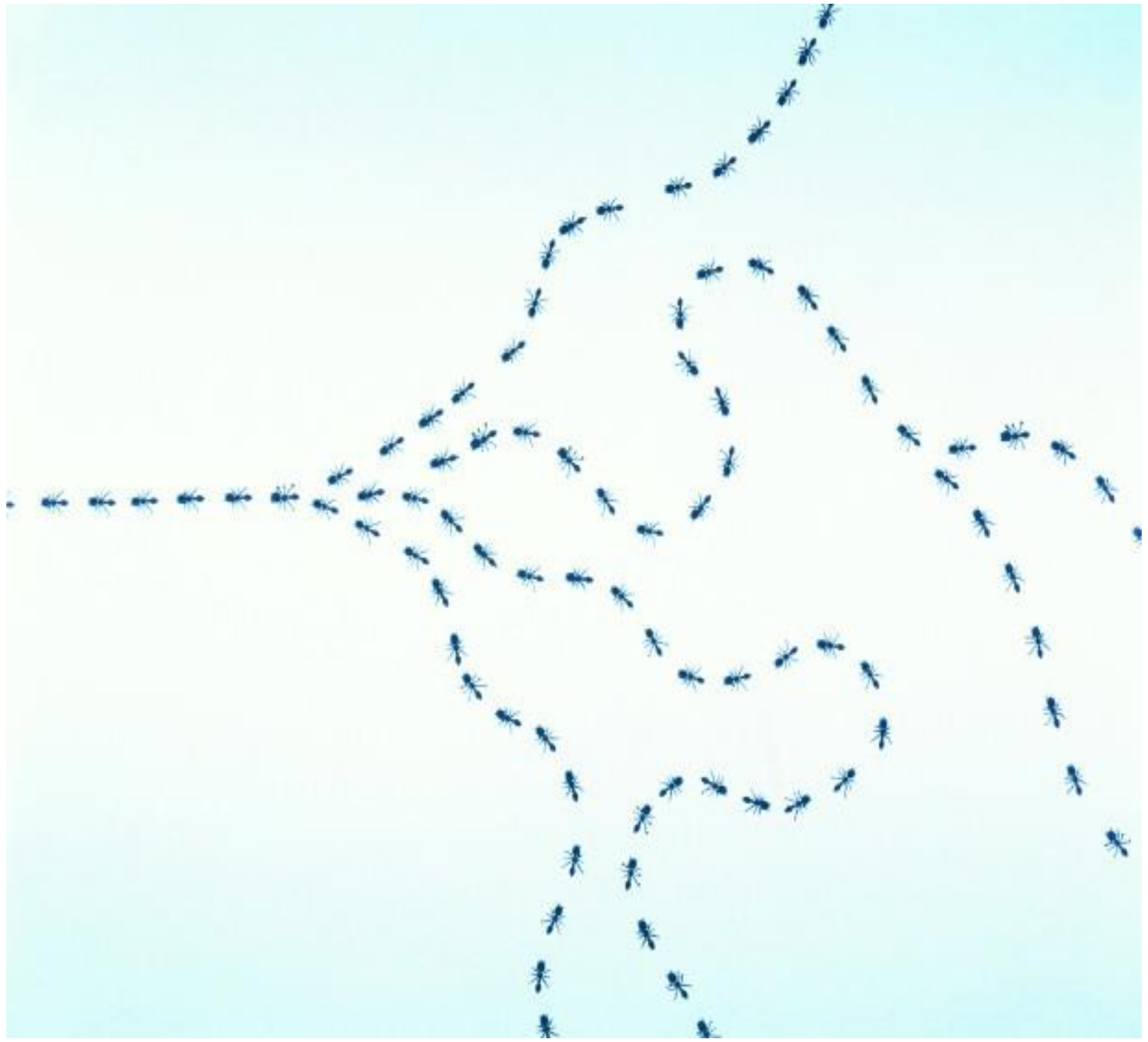
Population-Based Search (Global Search)



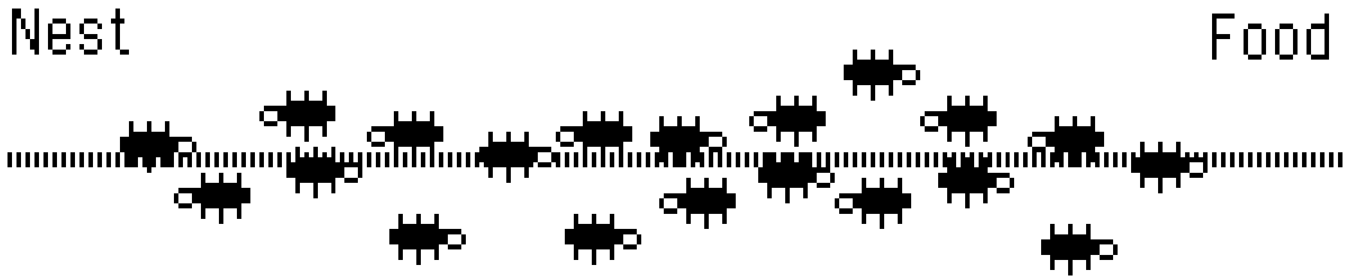
Introduction to Ant Colony Optimization (ACO)

- In nature, ants move randomly at first, but when they find food, they leave a chemical substance called ***pheromone*** on the ground.
- Other ants sense this pheromone and are more likely to follow that same path.
- Over time, shorter paths get stronger pheromone trails and this is how ants collectively find the shortest route without central control.
- In 1992, Marco Dorigo took this idea and turned it into a powerful optimization algorithm in his PhD.”
- ACO is a nature-inspired optimization algorithm based on how real ants find the shortest path between nest and food. It is widely used to solve hard optimization problems like:

  1. Routing and scheduling
  2. Feature selection and clustering

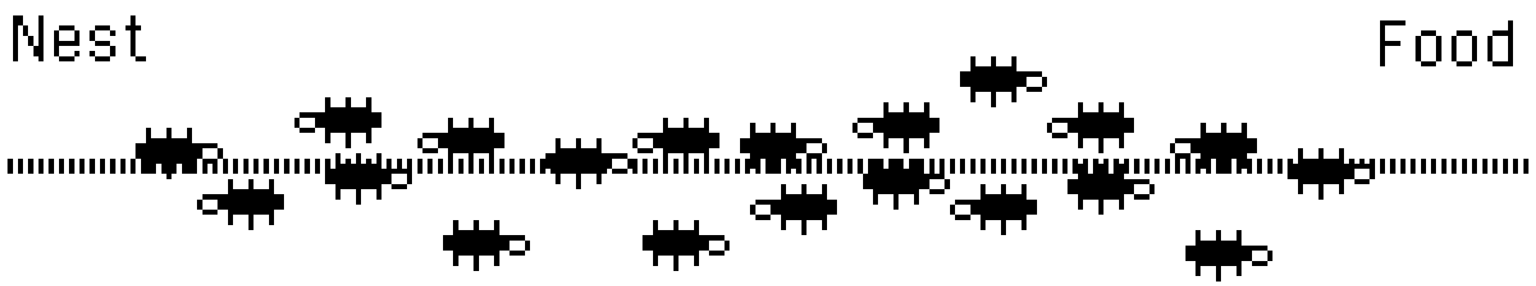


- In ACO, the nest is the starting point where all the ants begin their journey as it represents the source node in the optimization problem.

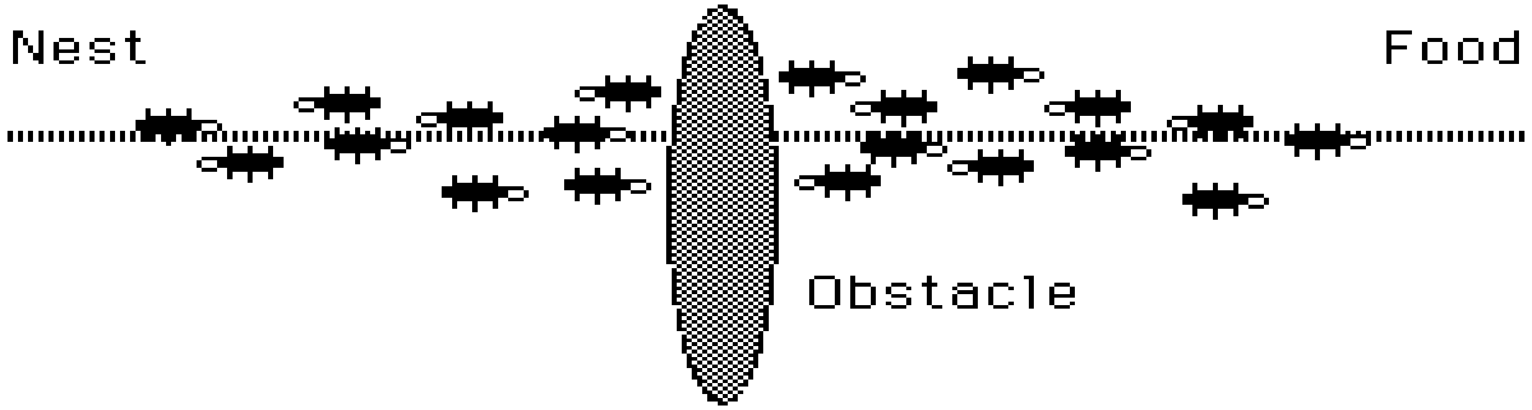


Naturally Observed Ant Behavior

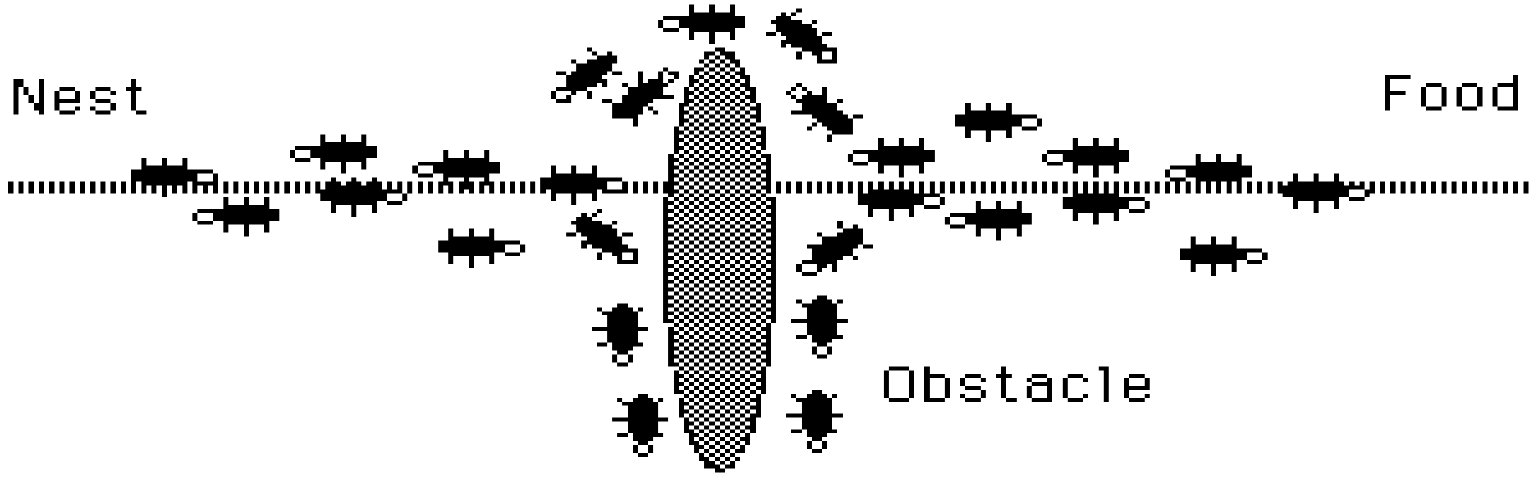
All is well in the world of the ant



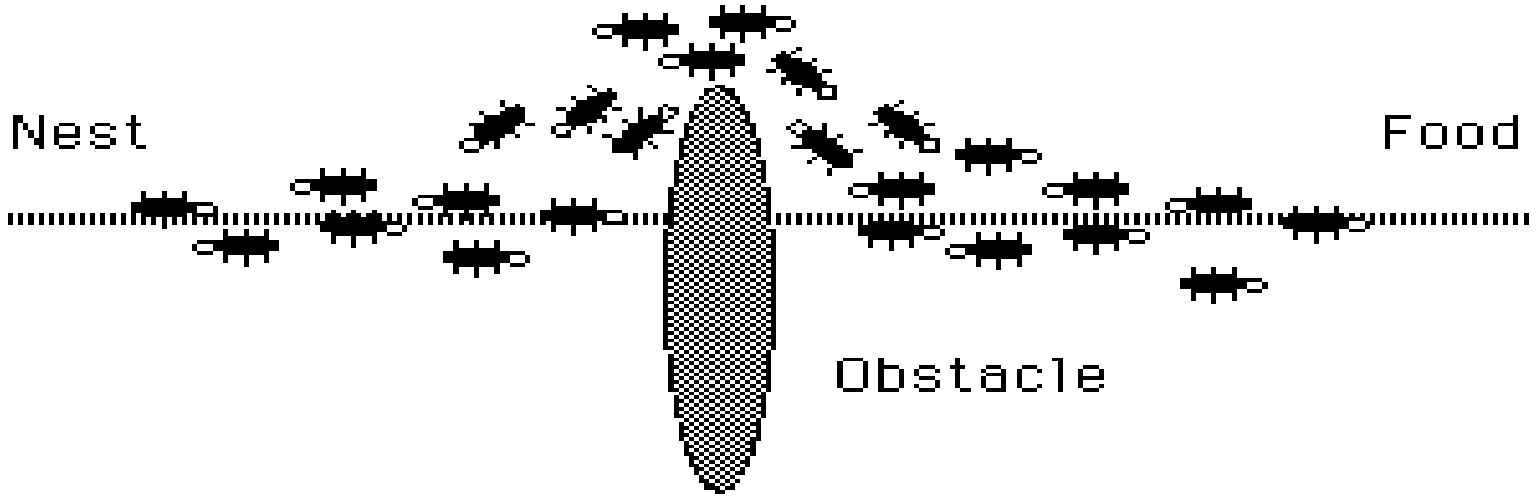
An obstacle has blocked the ant path



Where do the ant go? Everybody, flip a coin



Shorter path reinforced



How Real Ants Find the Shortest Path

- Ants move randomly at first when searching for food.
- Each ant drops a pheromone trail on its path.
- Other ants smell these pheromones and are more likely to follow stronger trails.
- Shorter paths get reinforced faster because ants travel them more often.
- Pheromone evaporation helps remove weak or longer paths over time.

Main steps:

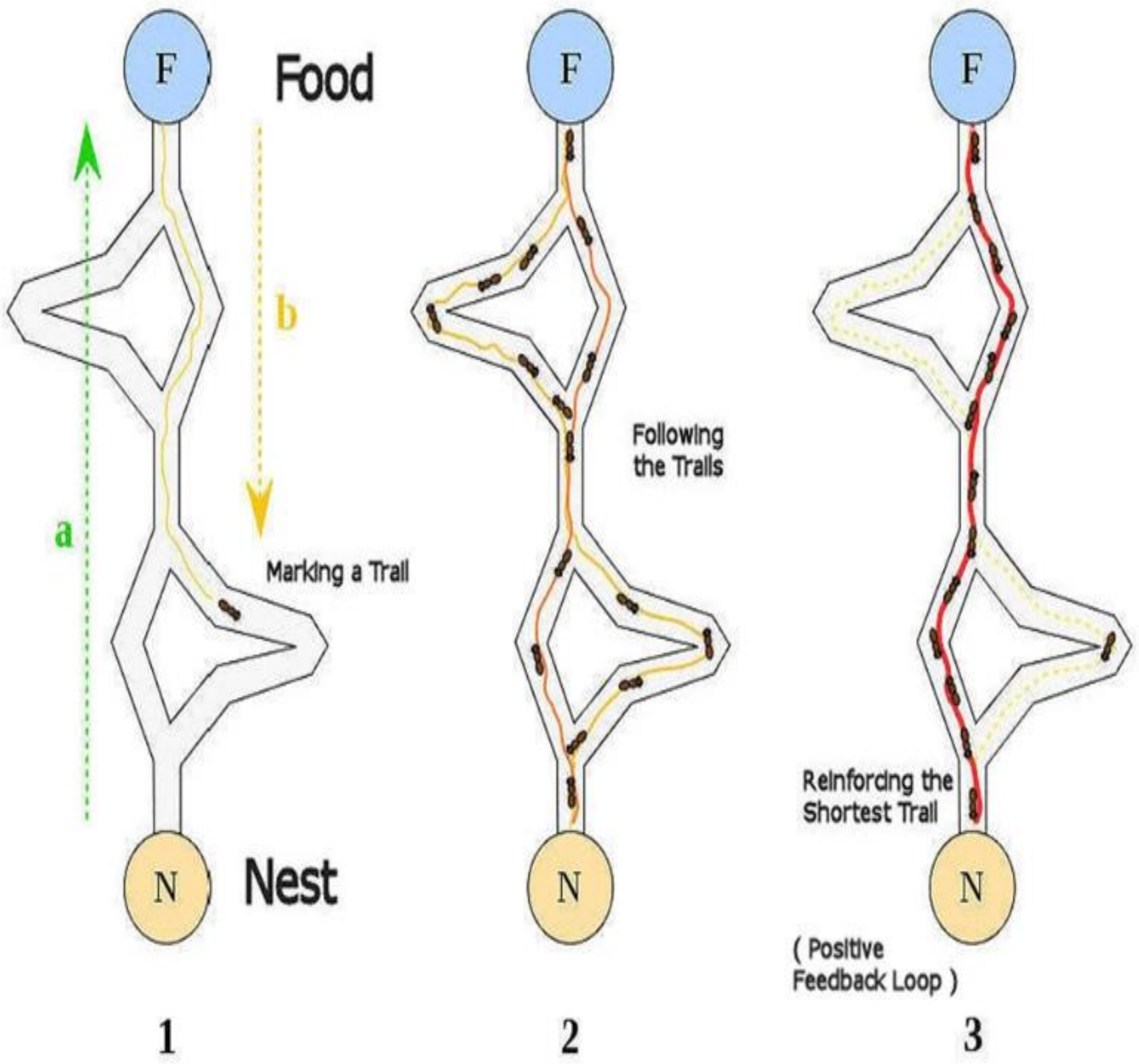
Pheromone deposit → Following → Reinforcement → Evaporation.

Stigmergy

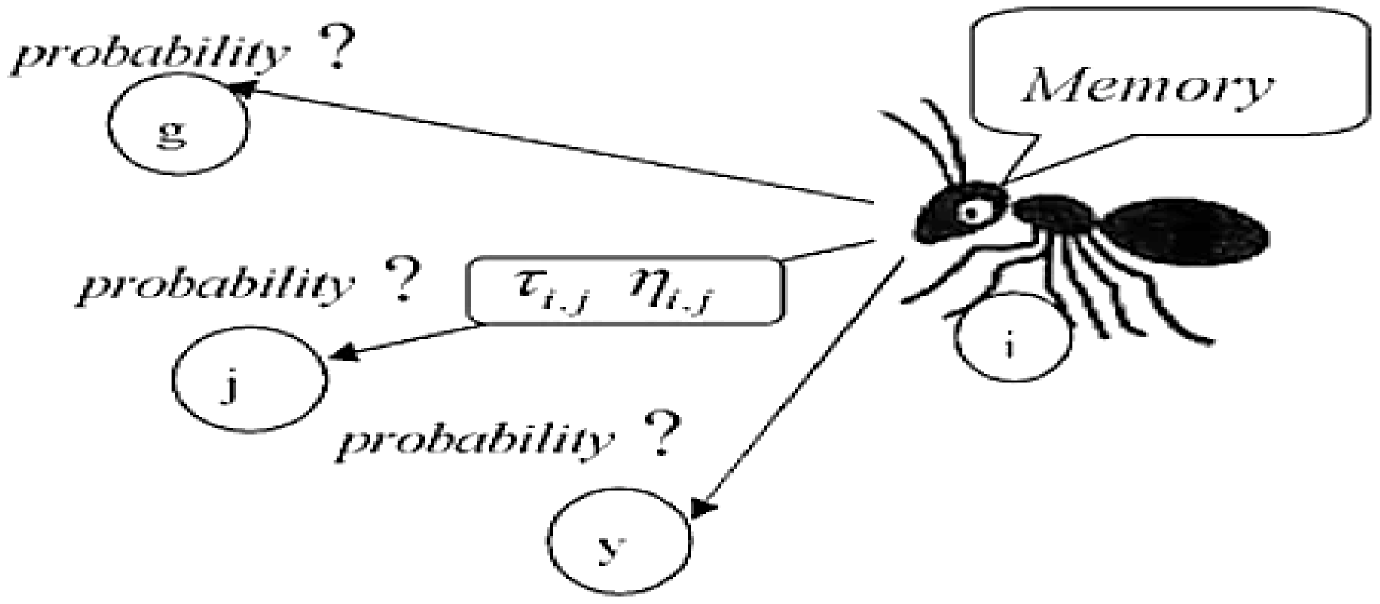
- A form of indirect communication.
- One ant changes the environment (by leaving pheromone).
- Another ant senses and reacts to that change later.
- This creates coordination without any central control.

Autocatalysis:

- A positive feedback loop.
- The more ants follow a path, the more pheromone accumulates.
- The stronger the pheromone, the more ants are attracted; reinforcing the best paths naturally.



Mathematical model of the ACO



- Define Probability Rule:
- $p_k(i,j) = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{u \in N_K(i)}[\tau_{iu}]^{\alpha}[\eta_{iu}]^{\beta}}, & if\ j \in N_K(i) \\ 0 & otherwise \end{cases}$
- Where $\eta_{i,j} = \frac{1}{d_{ij}}$

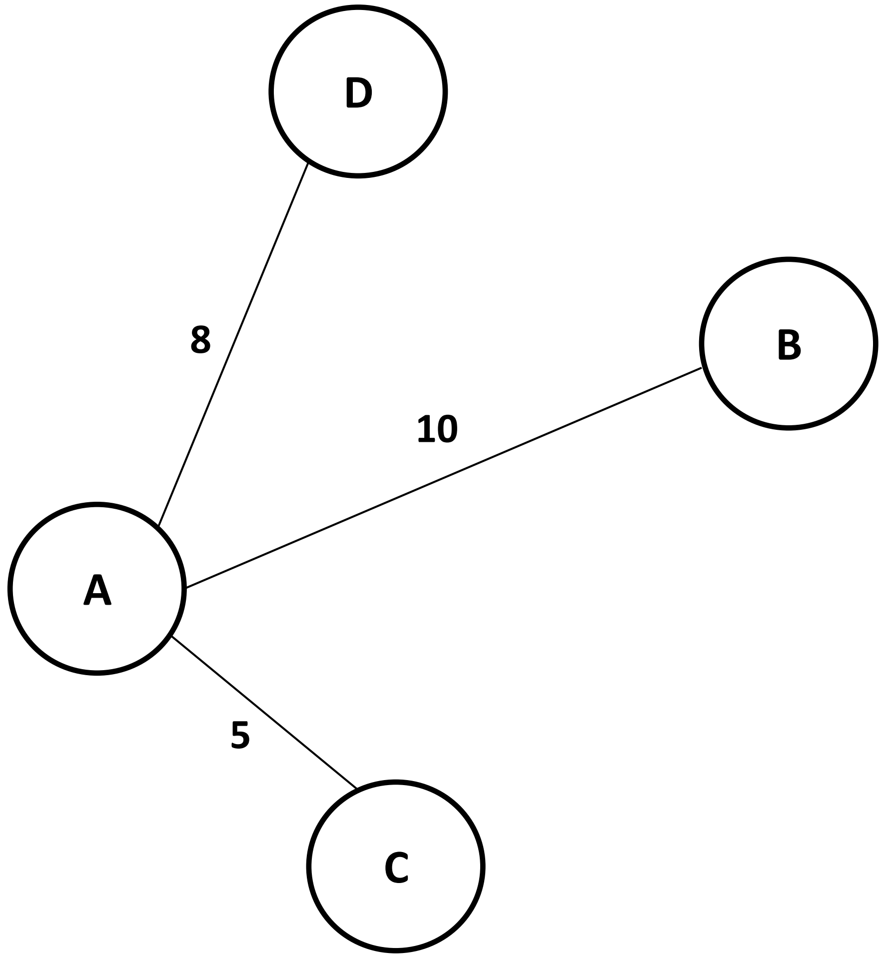
$p_k(i,j)$	▪ Probability that ant k moves from place i to place j
$\tau_{i,j}$	▪ Pheromone intensity on edge (i,j)
$\eta_{i,j}$	▪ Visibility between place i and place j
α	▪ Pheromone weight. ▪ Measure the degree of importance of pheromone (memory) ▪ Controls how strongly ants follow previous pheromone trails (exploitation).
β	▪ Visibility weight ▪ Measure the degree of importance of the distance ▪ Controls how much ants prefer shorter edges (exploration).

Typical parameter effect

- High α low β → Ants follow pheromone strongly → risk of getting stuck in one path
- low α low β → Ants focus on shortest visible edges → explore more
- Moderate α and β → Balanced exploration and exploitation (best practice)

How to use the probability Rule:

- Assume pheromone levels: $\tau_{AB} = 1$, $\tau_{AC} = 2$, $\tau_{AD} = 1.5$
- Assume: $\alpha = 1$, $\beta = 2$
- We can easily calculate visibility $\eta_{i,j} = \frac{1}{d_{ij}}$, accordingley: $\eta_{AB} = 0.1$, $\eta_{Ac} = 0.2$, $\eta_{AD} = 0.125$
- $p_k(i,j) = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{u \in N_K(i)}[\tau_{iu}]^{\alpha}[\eta_{iu}]^{\beta}}$



Steps:

1. **Compute the numerator:**

(A→B): $1^1 \times 0.1^2 = 0.01$
(A→C): $2^1 \times 0.2^2 = 0.08$
(A→D): $1.5^1 \times 0.125^2 = 0.0234375$

2. **Calculate denominator =** $\boldsymbol{sum} = 0.01 + 0.08 + 0.0234375 = 0.1134375$
3. **Find the probabilities:**

$$p_k(A,B) = \frac{0.01}{0.1134375} = 0.08815$$

$$p_k(A,C) = \frac{0.08}{0.1134375} = 0.706$$

$$p_k(A,D) = \frac{0.0234375}{0.1134375} = 0.206$$

Finally: most ants will go from A → C because it's both near and has strong pheromone.

Pheromone Update Rule

Step (1): Pheromone Evaporation Formula:

Step (2): Pheromone Deposition (Reinforcement)

Step (1): Pheromone Evaporation Formula:

$$\tau_{ij} \leftarrow (1-\rho) \times \tau_{ij}$$

Where:

- ρ is the evaporation rate, $0 < \rho < 1$
- It reduces pheromone strength over time
- Prevents unlimited accumulation and helps ants forget bad paths
- This encourages exploration of new routes

How to use this formula:

If $\tau_{12} = 1$ and $\rho = 0.3$, then:

${\tau_{12}}^{new} = (1-0.3) \times 1 = 0.7$, so pheromone trail weakness by 30%

Step 2: Pheromone Deposition (Reinforcement)

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k}$$

$$\Delta\tau_{ij}{}^{k} = \begin{cases} \dfrac{Q}{L_k}, & if\ ant\ k\ used\ edge\ (i,j) \\ 0 & otherwise \end{cases}$$

Q	▪ Pheromone constant ▪ Total pheromone amount each ant can deposit (often 1).
L_k	▪ Path length of ant k ▪ Total distance of the route; shorter path → larger pheromone deposit.
m	▪ Number of ants ▪ Each ant contributes to pheromone update
$\Delta\tau_{ij}{}^{k}$	▪ Pheromone contributed by ant k ▪ Only edges visited by the ant receive pheromone.
$iter_{max}$ (Max iterations)	▪ Number of algorithm cycles to run

Final combined update rule:

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

How to use the update rule:

Let's consider one edge $(1, 2)$:

- Previous pheromone: $\tau_{12} = 1$
- Evaporation rate: $\rho = 0.5$
- Two ants used this edge:

  Ant 1: total path $L_1 = 50$
  Ant 2: total path $L_2 = 100$

- $Q = 1$

Step (1): Evaporation: $\tau_{ij} \leftarrow (1 - \rho) \times \tau_{ij} = (1 - 0.5) \times 1 = 0.5$

Step (2): Deposition: $\Delta\tau_{ij}{}^{k} = \frac{Q}{L_k}$ → $\Delta\tau_{12}{}^{1} = \frac{1}{50} = 0.02$ **and** $\Delta\tau_{12}{}^{2} = \frac{1}{100} = 0.01$

Step (3): Update: $\Delta\tau_{12}{}^{new} = 0.5 + (0.02 + 0.01) = 0.53$

So, pheromone slightly increases because the edge was used by good paths.

Conclusion: This is the heart of learning in ACO. After each cycle, ants share what they found by depositing pheromone. If many ants used a short path, it gets a strong reinforcement. If no one used a path, pheromone evaporates. This way, over many iterations, good paths glow brighter , while bad ones disappear naturally."

Example:

- Assume 4 Nodes: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$

$$d(A, B) = 2$$
$$d(A, C) = 3$$
$$d(A, D) = 4$$
$$d(B, C) = 2$$
$$d(B, D) = 3$$
$$d(C, D) = 1$$

- Initial pheromone on every (undirected) edge: $\tau^0 = 1.0$
- ACO parameters:

$$\alpha = 1$$
$$\beta = 2$$
$$\eta = \frac{1}{d}$$

ρ (evaporation) $= 0.5$

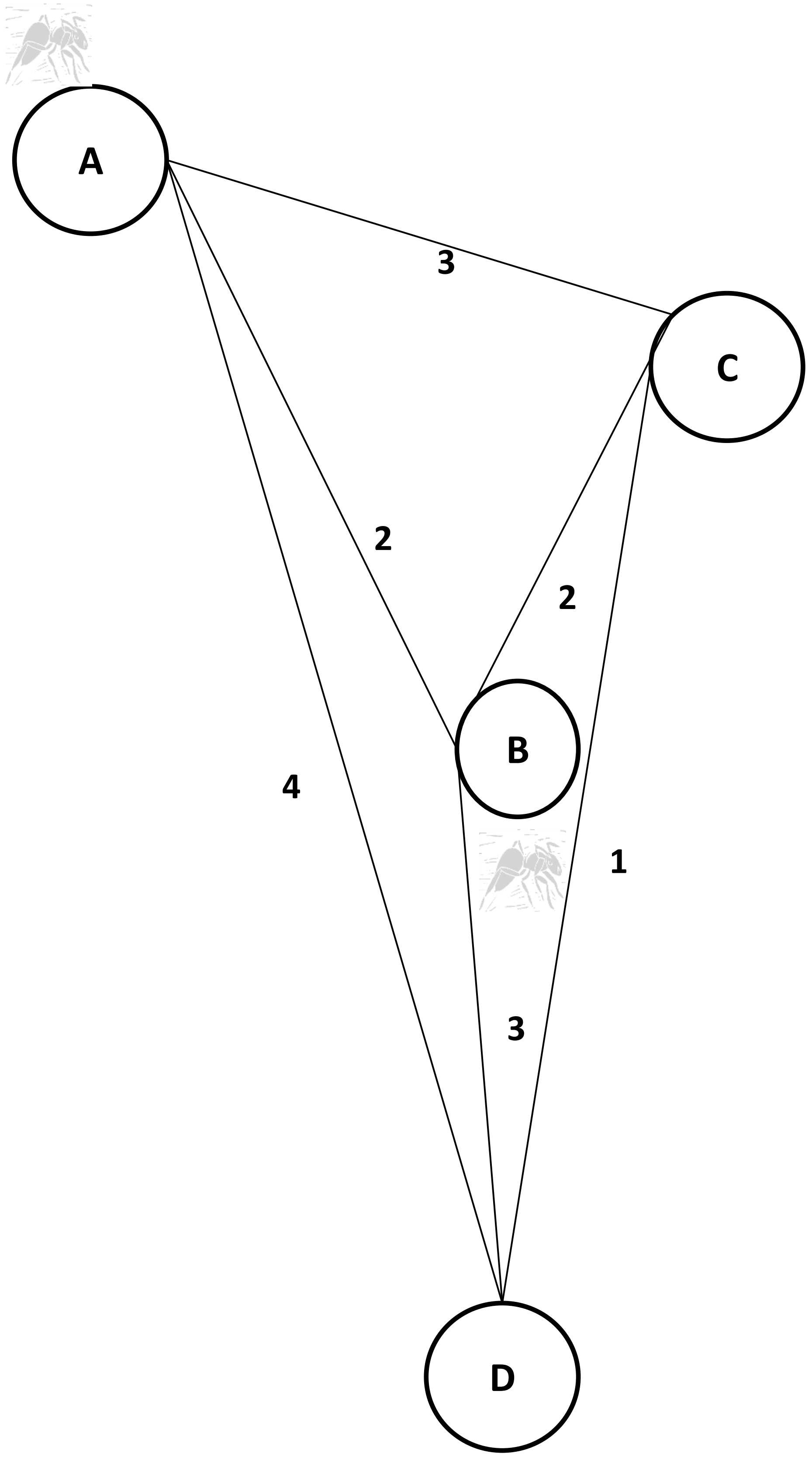
$Q = 100$

Number of ants: 2

Ant 1 starts at A

Ant 2 starts at B

We need to explore the best route



Step (0): Construct distance matrix

	A	B	C	D
A	0	2	3	4
B	2	0	2	3
C	3	2	0	1
D	4	3	1	0

Step (1): Compute visibility:

From → To	d_{ij}	$\eta = 1/d_{ij}$	η^2
A → B	2	0.5	0.25
A → C	3	0.33	0.11
A → D	4	0.25	0.625
B → C	2	0.5	0.25
B → D	3	0.33	0.11
C → D	1	1	1

Step (2): Compute Transition Probabilities and construct route subset

$p_i = \frac{[\tau_i]^\alpha[\eta_i]^\beta}{\sum_{j=1}^{N}[\tau_i]^\alpha[\eta_i]^\beta}$ given that initially all pheromones ($\boldsymbol{\tau}$) = 1.0 and α = 1, β = 2. take the random picks r = **0.50, and 0.60**

Ant 1 (starting at A):

route	τ	η^2	Probability	Cumulative Range	Random pick (r=0.5)
A → B	1	0.25	0.25 / (0.25 + 0.11+0.625) =0.59	[0.00 – 0.59]	0 < 0.5 < 0.59 (in range)
A → C	1	0.11	0.11 / (0.25 + 0.11+0.625) =0.26	[0.59 – 0.85)	Out of range
A → D	1	0.625	0.625 / (0.25 + 0.11+0.625) =0.15	[0.85 – 1]	Out of range

- So Ant1: **A → B**
- From B, the remaining nodes are C and D (A already visited)

route	τ	η^2	Probability	Cumulative Range	Random pick (r=0.6)
B → C	1	0.25	0.25 / (0.25 + 0.11) =0.69	[0.00 – 0.69]	In range
B → D	1	0.11	0.11 / (0.25 + 0.11) =0.31	[0.69 – 1)	Out of range

- So Ant1: **A → B →C**
- From C, remaining node is D (only one choice), then return to A)
- Complete Ant1 tour: **A** → **B** → **C** → **D** → **A**
- Total length $L_1 = 2 + 2 + 1 + 4 = 9$

Now, use Ant 2 (starting at B):

$p_i = \frac{[\tau_i]^{\alpha}[\eta_i]^{\beta}}{\sum_{j=1}^{N}[\tau_i]^{\alpha}[\eta_i]^{\beta}}$ given that initially all pheromones ($\boldsymbol{\tau}$) = 1.0 and α = 1, β = 2. take the random picks r = **0.60, and 0.05**

route	τ	η²	Probability	Cumulative Range	Random pick (r=0.6)
B→ A	1	0.25	0.25 / (0.25 + 0.25+0.11) =0.41	[0.00 – 0.41]	Out of range
B→ C	1	0.25	0.25 / (0.25 + 0.25+0.11) =0.41	[0.41 – 0.82)	In range
B → D	1	0.11	0.11 / (0.25 + 0.25+0.11) =0.18	[0.82 – 1]	Out of range

- So Ant2: **B → C**
- From C, the remaining nodes are A and D (A already visited)

route	τ	η²	Probability	Cumulative Range	Random pick (r=0.05)
C → A	1	0.11	0.11 / (1 + 0.11) = 0.1	[0.0 – 0.1]	In range
C → D	1	1	1 / (1 + 0.11) = 0.9	[0.1– 1)	Out of range

- So Ant2: **B → C →A**
- From A, remaining node is D (forced), then return to B)
- Complete Ant2 tour: **B** → **C** → **A** → **D** → **B**
- Total length $L_2 = 2 + 3 + 4 + 3 = 12$

Step (3): Pheromone update (evaporation + reinforcement)

Given: $\rho = 0.5, Q = 100, \tau_0 = 1.0$

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

- Evaporation part: $\tau_i \leftarrow (1-\rho) \times \tau_{ij} = (1-0.5) \times 1 = 0.5$, for all routes
- Deposit part: $\sum_{k=1}^{m} \Delta\tau_{ij}$

Ant	$\Delta\tau_i^{\ k} = \frac{Q}{L_K}$
1	$\Delta\tau_i^{\ 1} = \frac{100}{9} = 11.11$
2	$\Delta\tau_i^{\ 2} = \frac{100}{12} = 8.33$

Ant1 (A→B→C→D→A)
Ant2 (B→C→A→D→B)

route	Used by	Sum deposits
A-B	Ant 1	11.11
B-C	Ant 1, Ant 2	$11.11 + 8.33 = 19.44$
C-D	Ant 1	11.11
D-A=A-D	Ant 1, Ant 2	$11.11 + 8.33 = 19.44$
C-A	Ant 2	8.33
D-B	Ant 2	8.33

route	Evaporated (0.5)	Deposits	Updated τ_i
A-B	0.5	11.11	11.61
B-C	0.5	19.44	19.94
C-D	0.5	11.11	11.61
D-A=A-D	0.5	19.44	19.94
C-A	0.5	8.33	8.83
D-B	0.5	8.33	8.83

- Best route until now: A → B → C → D → A
- Ready for the next iteration

Example: We have a dataset with 5 possible features (F_1, F_2, F_3, F_4, F_5).
Our goal is to select the best subset of features that gives the highest accuracy and lowest model complexity when training a CNN or MLP.
Each feature improves model accuracy but also increases computational complexity. Our goal: Maximize Accuracy Gain while minimizing Complexity Cost

Feature	Accuracy Gain (%)	Complexity Cost
F_1	9	4
F_2	7	3
F_3	6	2
F_4	4	1
F_5	8	5

Assumptions:

Number of ants (m)	2
Pheromone (initial)→ τ_0	1
Evaporation rate (ρ)	0.5
α	1
β	2
Number of features each ant select	3
Q	1

Important note:

- In classical ACO, the heuristic information η_{ij} is defined as $\eta_{i,j} = \frac{1}{d_{ij}}$, where d_{ij} represents the distance between nodes. However, in feature selection problems, there is no physical distance; instead, the heuristic must reflect how desirable each feature is for the learning model.
- Therefore, η_i is redefined as the ratio between a feature's accuracy gain and its complexity cost ($\eta_i = \frac{Gain_i}{cost_i}$).
- This formulation ensures that features providing higher performance improvement with lower computational cost are more attractive to the ants, analogous to how shorter distances attract ants in classical optimization.

Step (1): Calculate visibility

Feature	Accuracy Gain (%)	Complexity Cost	$\eta_i = \frac{Gain_i}{cost_i}$
F_1	9	4	2.25
F_2	7	3	2.33
F_3	6	2	3.00
F_4	4	1	4.00
F_5	8	5	1.60

Step (2): Compute Probability of Selecting Each Feature

$p_i = \frac{[\tau_i]^\alpha[\eta_i]^\beta}{\sum_{j=1}^{N}[\tau_i]^\alpha[\eta_i]^\beta}$ given that initially all pheromones ($\boldsymbol{\tau}$) = 1.0 and α = 1, β = 2:

Feature	$\boldsymbol{\tau}$	$\boldsymbol{\eta}$	$\boldsymbol{\eta}^2$	$\boldsymbol{\tau} \times \boldsymbol{\eta}^2$	Probability
F_1	1	2.25	5.06	5.06	$5.06/\mathbf{38.05} = 0.133$
F_2	1	2.33	5.43	5.43	$5.43/\mathbf{38.05} = 0.143$
F_3	1	3.00	9.00	9.00	$9.00/\mathbf{38.05} = 0.2364$
F_4	1	4.00	16.00	16.00	$16.00/\mathbf{38.05} = 0.4203$
F_5	1	1.60	2.56	2.56	$2.56/\mathbf{38.05} = 0.0673$
				Sum = 38.05	**Sum = 1**

Conclusion: F4 is most attractive (≈42%), then F3 (≈23.6%), F2 (≈14.3%), F1 (≈13.3%), F5 (≈6.7%).

- Using probabilities directly makes all ants pick the highest-probability feature, causing the search to become greedy and get stuck in local minima.
- Random numbers introduce **stochastic exploration**, allowing each ant to make a unique, probabilistic decision.
- Cumulative probability converts discrete probabilities into **continuous intervals** on the 0 –1 scale.
- Each random number falls inside one interval, determining which feature the ant selects.
- This combination maintains a balance between **exploration** (discovering new features) and **exploitation** (reinforcing good ones), leading to optimal convergence.

Feature	**Probability**	**Cumulative range**
F_1	0.133	0 → 0.133
F_2	0.143	0.133 → 0.276
F_3	0.2364	0.276 → 0.5124
F_4	0.4203	0.5124 →0.9327
F_5	0.0673	0.9327→1

Step (3): ants construct feature subsets

Ant	**Assume Random numbers** r_1, r_2, r_3	Chosen features
1	0.7, 0.4, 0.2	F_4 (0.51→0.9327), F_3 (0.276→0.5124), F_2 (0.133→0.276) → **{F_4,F_3,F_2}**
2	0.3, 0.05, 0.6	F_3 (0.276→0.5124), F_1 (0→0.133), F_4 (0.5124→0.9327) → **{F_3,F_1,F_4}**
3	0.15, 0.8, 0.1	F_2 (0.133→0.276), F_4 (0.5124→0.9327), F_1 (0→0.133) → **{F_2,F_4,F_1}**

Step (4): Evaluate Fitness of Each Ant: Fitness is based on the ratio of **total accuracy gain** to **total complexity cost**: $f_k = \frac{total\ gain}{total\ cost}$

Ant	Selected Features	Total gain	Total cost	Fitness= f_k
1	{F_4, F_3, F_2}	17	6	2.83
2	{F_3, F_1, F_4}	19	7	2.71
3	{F_2, F_4, F_1}	20	8	2.50

Best subset this iteration: Ant 1 → {F_4, F_3, F_2}.

Step (5): Pheromone update (evaporation then deposit) :

$$\tau_i \leftarrow (1-\rho) \times \tau_i + \sum_{k=1}^{m} Q\, f_k$$

- Evaporation part: $\tau_i \leftarrow (1-\rho) \times \tau_{ij} = (1-0.5) \times 1 = 0.5$, for all features
- Deposit part: $\sum_{k=1}^{m} Q\, f_k$:

Ant	$\Delta\tau_i{}^k = Q\, f_k$
1	$\Delta\tau_i{}^1 = 1 \times 2.83 = 2.83$
2	$\Delta\tau_i{}^2 = 1 \times 2.71 = 2.71$
3	$\Delta\tau_i{}^3 = 1 \times 2.5 = 2.5$

feature	Used by	Sum deposits
F_1	Ant 2, Ant 3	$2.71 + 2.5 = 5.21$
F_2	Ant 1, Ant 3	$2.83 + 2.5 = 5.33$
F_3	Ant 1, Ant 2	$2.83 + 2.71 = 5.54$
F_4	Ant 1, Ant 2, Ant 3	$2.83 + 2.71 + 2.5 = 8.04$
F_5	None	0

feature	Evaporated (0.5)	Deposits	Updated τ_i
F_1	0.5	5.21	5.71
F_2	0.5	5.33	5.83
F_3	0.5	5.54	6.04
F_4	0.5	8.04	8.54
F_5	0.5	0	0.5

F_4 receives the largest pheromone boost (it appeared in all three ant solutions). F_5 receives none and remains at 0.5.

Step (6): Compute next-iteration selection probabilities:

Feature	τ (updated)	η	η^2	$\tau \times \eta^2$	Probability
F_1	5.71	2.25	5.06	28.89	$28.89/\mathbf{252.83} = 0.1143$
F_2	5.83	2.33	5.43	31.66	$31.66/\mathbf{252.83} = 0.1252$
F_3	6.04	3.00	9.00	54.36	$54.36/\mathbf{252.83} = 0.2150$
F_4	8.54	4.00	16.00	136.64	$136.64/\mathbf{252.83} = 0.5404$
F_5	0.5	1.60	2.56	1.28	$1.28/\mathbf{252.83} = 0.0051$
				Sum =252.83	**Sum = 1**

- F_4 now dominates (~54.0%); F_3 remains significant (~21.5%); F_2 & F_1 moderate; F_5 almost ignored.
- Using the new probabilities, ants will construct new subsets in the next iteration (with F_4 very likely).
- Repeat evaluation and pheromone updates (evaporation + deposits) each iteration.

Exploration and Exploitation In ACO

Exploration:

- Ants try different paths, even if pheromone levels are low.
- Controlled by randomness in path selection.
- Exploration: $\alpha \downarrow$, $\beta \downarrow$, $\rho \uparrow$

Exploitation

- Ants follow stronger pheromone trails, preferring high-quality paths.
- Leads to faster convergence.
- Exploitation: $\alpha \uparrow$, $\beta \uparrow$, $\rho \downarrow$

Parameter	Low Value Effect	High Value Effect	Conclusion
α (pheromone weight)	Ants ignore pheromone → explore randomly	Ants follow pheromone → exploit best paths	Low α → exploration High α → exploitation
β (heuristic weight)	Ants ignore distance info → explore	Ants prefer short visible edges → exploit heuristic	Low β → exploration High β → exploitation
ρ (evaporation rate)	Slow evaporation → pheromone lasts → exploit	Fast evaporation → pheromone fades → explore	Low ρ → exploitation High ρ → exploration

Drawbacks of ACO

1. Slow convergence	Needs many iterations to find the best path.
2. Local optimum	May get stuck in one strong path.
3. High computation	Many ants → high time and memory use.
4. Parameter sensitive	Results depend on α, β, ρ tuning.
5. Scalability issue	Becomes slow for large problems.

Step (1): Load dataset and define ACO parameters:

```python
# Load dataset
data = load_iris()
X = data.data
y = data.target
n_features = X.shape[1]

# Parameters
n_ants = 5
n_iterations = 5
alpha = 1.0        # pheromone importance
beta = 1.0         # heuristic importance (accuracy)
rho = 0.3          # evaporation rate
Q = 1.0            # pheromone deposit factor
```

Step (2): Initialize pheromone levels for each feature:

```python
pheromone = np.ones(n_features)

def evaluate(features):
    """Return accuracy using selected features"""
    if np.sum(features) == 0:
        return 0  # No features selected
    X_train, X_test, y_train, y_test = train_test_split(X[:, features == 1], y, test_size=0.3, random_state=42)
    model = LogisticRegression(max_iter=500)
    model.fit(X_train, y_train)
    pred = model.predict(X_test)
    return accuracy_score(y_test, pred)
```

Step (3): Load dataset and define ACO parameters:

```python
#  ACO loop
for iteration in range(n_iterations):
    print(f"\n=== Iteration {iteration+1} ===")

    all_solutions = []
    all_accuracies = []

    for ant in range(n_ants):
        # Each ant builds a feature subset probabilistically
        probs = pheromone ** alpha
        probs = probs / np.sum(probs)
        random_values = np.random.rand(n_features)
        features = (random_values < probs).astype(int)  # choose features probabilistically

        acc = evaluate(features)
        all_solutions.append(features)
        all_accuracies.append(acc)
        print(f"Ant {ant+1}: Features={features}, Accuracy={acc:.3f}")
```

Step (4): update phermone:

```
#  Update pheromone (evaporation + reinforcement)
pheromone = (1 - rho) * pheromone  # evaporation
best_ant = np.argmax(all_accuracies)
best_features = all_solutions[best_ant]
best_acc = all_accuracies[best_ant]

# reinforcement: add pheromone to good features
pheromone += Q * best_features * best_acc

print(f"Best Ant: {best_ant+1}, Best Accuracy: {best_acc:.3f}")
print(f"Updated Pheromone: {np.round(pheromone, 3)}")
```

Step (5): final results:

```
best_features_overall = np.argsort(-pheromone)[:np.sum(pheromone > np.mean(pheromone))]
print("\n Final Selected Features:", best_features_overall)
```

Note that: in iris dataset:

$F = [F_0\ F_1\ F_2\ \ F_3]$

Where:

F_0 = sepal length

F_1 = sepal width

F_2 = petal length

F_3 = petal width

```
=== Iteration 1 ===
Ant 1: Features=[0 1 0 1], Accuracy=0.956
Ant 2: Features=[0 0 0 1], Accuracy=1.000
Ant 3: Features=[0 1 1 0], Accuracy=1.000
Ant 4: Features=[0 0 0 1], Accuracy=1.000
Ant 5: Features=[0 0 0 0], Accuracy=0.000
Best Ant: 2, Best Accuracy: 1.000
Updated Pheromone: [0.7 0.7 0.7 1.7]

=== Iteration 2 ===
Ant 1: Features=[0 0 0 1], Accuracy=1.000
Ant 2: Features=[0 0 0 0], Accuracy=0.000
Ant 3: Features=[0 0 0 0], Accuracy=0.000
Ant 4: Features=[0 1 0 1], Accuracy=0.956
Ant 5: Features=[0 0 0 0], Accuracy=0.000
Best Ant: 1, Best Accuracy: 1.000
Updated Pheromone: [0.49 0.49 0.49 2.19]

=== Iteration 3 ===
Ant 1: Features=[0 0 0 0], Accuracy=0.000
Ant 2: Features=[0 1 0 0], Accuracy=0.556
Ant 3: Features=[0 0 0 0], Accuracy=0.000
Ant 4: Features=[0 0 0 1], Accuracy=1.000
Ant 5: Features=[0 0 1 1], Accuracy=1.000
Best Ant: 4, Best Accuracy: 1.000
Updated Pheromone: [0.343 0.343 0.343 2.533]
=== Iteration 4 ===
Ant 1: Features=[0 0 0 1], Accuracy=1.000
Ant 2: Features=[0 0 0 0], Accuracy=0.000
Ant 3: Features=[0 0 0 0], Accuracy=0.000
Ant 4: Features=[0 0 1 1], Accuracy=1.000
Ant 5: Features=[0 0 0 1], Accuracy=1.000
Best Ant: 1, Best Accuracy: 1.000
Updated Pheromone: [0.24  0.24  0.24  2.773]

=== Iteration 5 ===
Ant 1: Features=[0 0 0 1], Accuracy=1.000
Ant 2: Features=[0 0 0 0], Accuracy=0.000
Ant 3: Features=[0 0 0 0], Accuracy=0.000
Ant 4: Features=[0 0 0 1], Accuracy=1.000
Ant 5: Features=[0 0 0 1], Accuracy=1.000
Best Ant: 1, Best Accuracy: 1.000
Updated Pheromone: [0.168 0.168 0.168 2.941]

 Final Selected Features: [3]
```