SWARM INTELLIGENCE

Introduction to Swarm Intelligence

Social Behavior and Survival

- Many species benefit from living in social groups
- Sociality improves **survival and efficiency**
- Key advantages of social living:
  - Increased mating opportunities
  - Efficient food retrieval
  - Reduced predator attacks
  - Division of labor
  - Cooperative hunting

Introduction to Swarm Intelligence

Inspiration for Computational Systems

- Social behaviors inspired **computational problem-solving methods**
- Used in:
  - Optimization problems
  - Coordination strategies
  - Collective robotics (swarm robotics)
- Natural systems → models for **distributed intelligence**
- Focus on **simple agents + interaction = intelligent behavior**

What is Swarm Intelligence?

- Term "**Swarm Intelligence**" coined in late 1980s
- Introduced by **Beni (1988)** and **Beni & Wang (1989)**
- Refers to systems where:
  - Many **simple agents**
  - Interact using **local rules**
  - Operate in a shared environment
- No global control or centralized decision-making

Definitions from Literature

- **White & Pagurek (1998):**
  - Swarm Intelligence is a property of systems made of *unintelligent agents*
  - Limited individual capabilities
  - Exhibit **collectively intelligent behavior**
- **Bonabeau et al. (1999):**
  - Design of algorithms inspired by
    - Social insects
    - Animal societies
  - Focus on **distributed problem-solving**
- **Kennedy et al. (2001):**
  - Swarms can exist beyond physical space
  - Can operate in **cognitive (search) space**

Concept of a Swarm

- A **swarm** refers to a loosely structured collection of interacting agents
- Agents interact without centralized control
- Classic example: **swarm of bees**
- Concept extends to many systems:
  - Ant colonies
  - Flocks of birds
  - Traffic systems
  - Human crowds
  - Immune systems
  - Economies
- A swarm represents **collective behavior**, not just physical movement

Swarms in Abstract Space

- Swarms are not limited to physical motion
- Can exist in:
  - Cognitive space
  - Problem-solving space
  - Search and optimization space
- Focus is on **collective action**, not location
- Applicable to computational intelligence systems
- Swarm behavior applies to **both physical and abstract systems**

What is an Agent?

- An **agent** is an entity that:
  - Senses the environment
  - Acts upon the environment
- Actions may include:
  - Modifying the environment
  - Interacting with other agents
- Agents are simple, but interactions create intelligence
- Simple agents + interaction = intelligent system

Principles of Swarm Systems (Millonas, 1994)

- **Proximity:**
  - Agents must interact locally to form social links
- **Quality:**
  - Agents evaluate interactions with environment and others
- **Diversity:**
  - Variety improves response to unknown situations
- **Stability:**
  - Behavior should not change drastically due to small disturbances
- **Adaptability:**
  - Ability to adjust to environmental and population changes

Main Research Areas

- Two major research streams:
  - Systems inspired by **social insects**
  - Systems inspired by **human social behavior**
- Swarm models involve:
  - Populations of individuals
  - Direct or indirect interaction
- Interactions may:
  - Change the environment
  - Change agent behavior
- Lead to **useful emergent phenomena**

Applications of Swarm Intelligence

- Study of social insects (especially ants)
- Ant behaviors discussed:
  - Foraging
  - Clustering
  - Sorting tasks
- Applications include:
  - Combinatorial optimization
  - Data clustering
  - Robotics coordination
- Natural swarm behavior → powerful algorithms

Ant Colonies:

Collective Behavior in Nature

- Interaction among group members enables collective behavior
- Communication types:
  - Chemical (pheromones)
  - Visual signals
- Interaction with environment influences group decisions
- Same individuals → different collective outcomes under different conditions
- Observed in:
  - Social insects
  - Fish schools
  - Bird flocks
  - Mammal groups
- Collective behavior emerges from interaction, not intelligence

Ant Colony Optimization (ACO)

- Inspired by ants foraging for food
- Based on **discrete optimization problems**
- Ants find shortest paths using pheromone trails
- Artificial ants simulate this behavior
- Used in:
  - Path finding
  - Scheduling
  - Routing problems
- Simple rules lead to optimal solutions

Ant Foraging Behavior

- Large variety of ant species exist worldwide
- Many species exhibit **similar foraging behaviors**
- Experiments show ants can:
  - Exploit rich food sources
  - Explore the environment simultaneously
  - Find the **shortest path** between nest and food
- Demonstrates a balance between **exploration and exploitation**

Ant Foraging Behavior

Exploration vs Exploitation

- **Exploitation**: Efficient use of known rich food sources
- **Exploration**: Searching for new or better food sources
- Ant colonies achieve this balance **without central control**
- This capability inspired:
  - Behavioral models of ants
  - Computational problem-solving techniques

Ant Foraging Behavior

- **From Ant Behavior to Algorithms**
- Observations of ant foraging led to:
  - Mathematical models of ant behavior
  - Development of **Ant Colony Optimization (ACO)** algorithms
- ACO algorithms solve:
  - Combinatorial optimization problems
  - Path-finding and routing problems
- Inspired directly by **real ant colonies**

Ant Foraging Behavior

Absence of Central Control

- Although ants live in colonies:
  - No leaders
  - No templates or blueprints
  - No global planning
- Collective behavior emerges from:
  - Simple local interactions
  - Individual decision-making
- Confirmed by studies (Camazine et al., 2001)

Ant Foraging Behavior

- **Role of Local Information**
- Ants communicate using **local pheromone information**
- Different trail pheromone concentrations compete
- Individual ants respond to:
  - Strength of pheromone trails
- Collective decisions emerge from:
  - Local competition among trails
  - Positive feedback mechanisms

Simple Trail Formation Experiment

- **Experimental Setup**
- Place a dish of sugar solution near an ant nest
- Initially:
    - Few forager ants discover the food
- Shortly after:
    - Recruitment attracts more ants to the food source

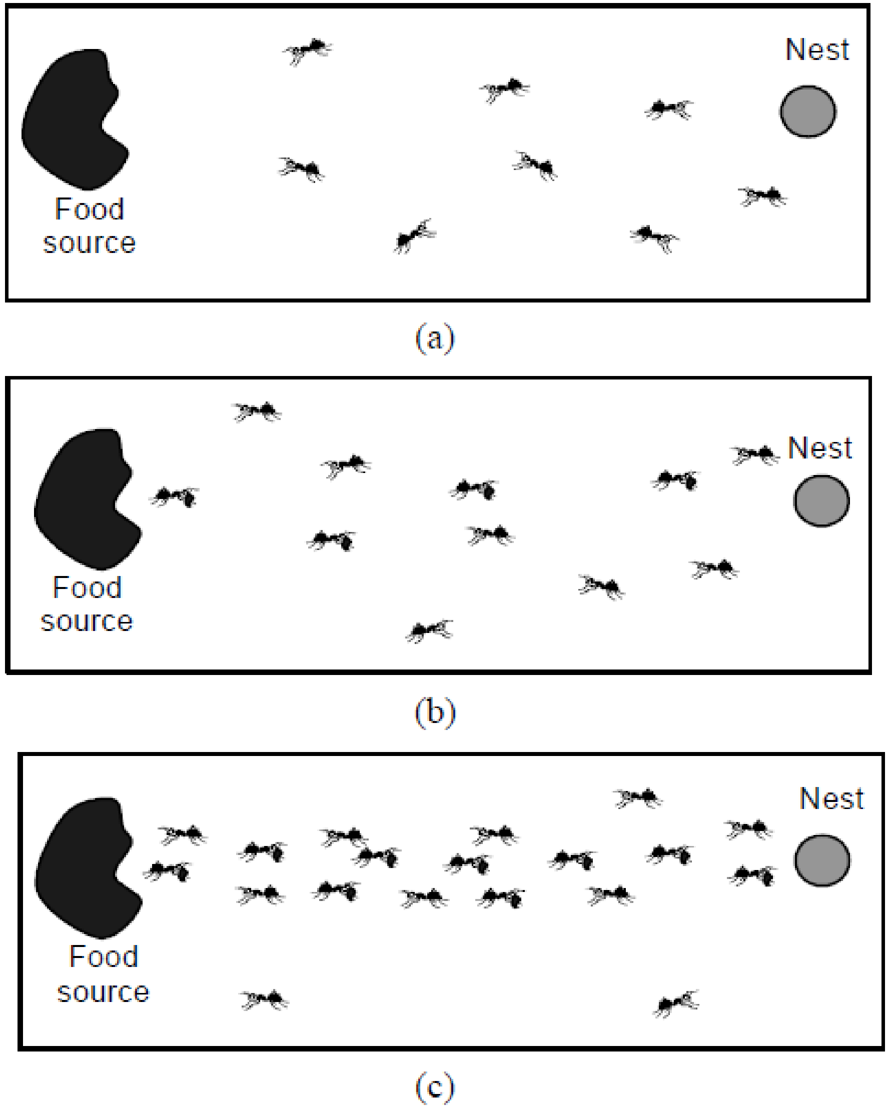

Figure 5.1: In the collective foraging of some ant species, ants recruit nestmates by releasing pheromone on the path from the food source to the nest; a pheromone trail is thus established. (a) Foraging ants. (b) A few ants find the food source and start recruiting nestmates by releasing pheromone. (c) A pheromone trail is formed.

Simple Trail Formation Experiment

- **Development of Pheromone Trails**
- Ants begin trafficking between:
  - Nest ⟷ Food source
- Movement appears organized:
  - As if ants are following a visible trail
- Trail is formed due to:
  - Pheromone laying and reinforcement
- Demonstrates **self-organized trail formation**

Simple Trail Formation Experiment

- **Evidence of Exploration**
- Not all ants strictly follow the trail
- Some ants deviate and wander
- This behavior:
  - Helps locate alternative food sources
  - Prevents over-dependence on a single trail
- Random exploration is **essential for adaptability**

Simple Trail Formation Experiment

- **Key Observations from the Experiment**
- Recruitment leads to rapid exploitation
- Pheromone trails guide collective movement
- Random deviations enable exploration
- Balance of both behaviors leads to:
  - Efficient foraging
  - Robust decision-making

Types of Recruitment Mechanisms

- **1. Mass Recruitment**
- Scout discovers food and lays a **pheromone trail**
- Nestmates detect trail and follow it
- Recruited ants reinforce trail on return
- **2. Tandem Recruitment**
- Scout physically leads **one ant at a time**
- Close contact between leader and follower
- **3. Group Recruitment**
- Scout leads a **small group**
- Uses short-range chemical attractants
- → After feeding, **recruited ants become recruiters**

Positive Feedback in Recruitment

- Recruitment is a **self-reinforcing process**
- More ants → more pheromone → more ants
- Results in rapid trail amplification
- Process slows down when:
    - Few ants remain to recruit
    - Competing food sources exist
- Forms the basis of **collective intelligence** in ant societies

Pheromone Trail Formation

- Trail formation consists of **two key stages**:
- **Trail-Laying**
  - Scout lays pheromone from food to nest
- **Trail-Following**
  - Other ants follow the pheromone trail
  - Reinforce it after food ingestion
- Pheromones are:
  - Low molecular weight chemicals
  - Produced by glands or gut
  - Subject to **slow evaporation**

Functions of Pheromone

- Pheromone serves **two major purposes**:
- **Defines the trail path**
- **Acts as an orientation signal** for ants
- Trail persists only with **continuous reinforcement**
- If food is depleted:
  - No reinforcement
  - Trail gradually disappears
- Ensures adaptability to changing environments

Shortest Path Discovery Experiment

- **Researchers:** Goss et al. (1989), Deneubourg et al. (1990)
  Species: Argentine ant (*Iridomyrmex humilis*)
- **Experimental Setup**
- Nest connected to food via a bridge
- Bridge has **two branches of different lengths**
- Ants must choose one branch each trip

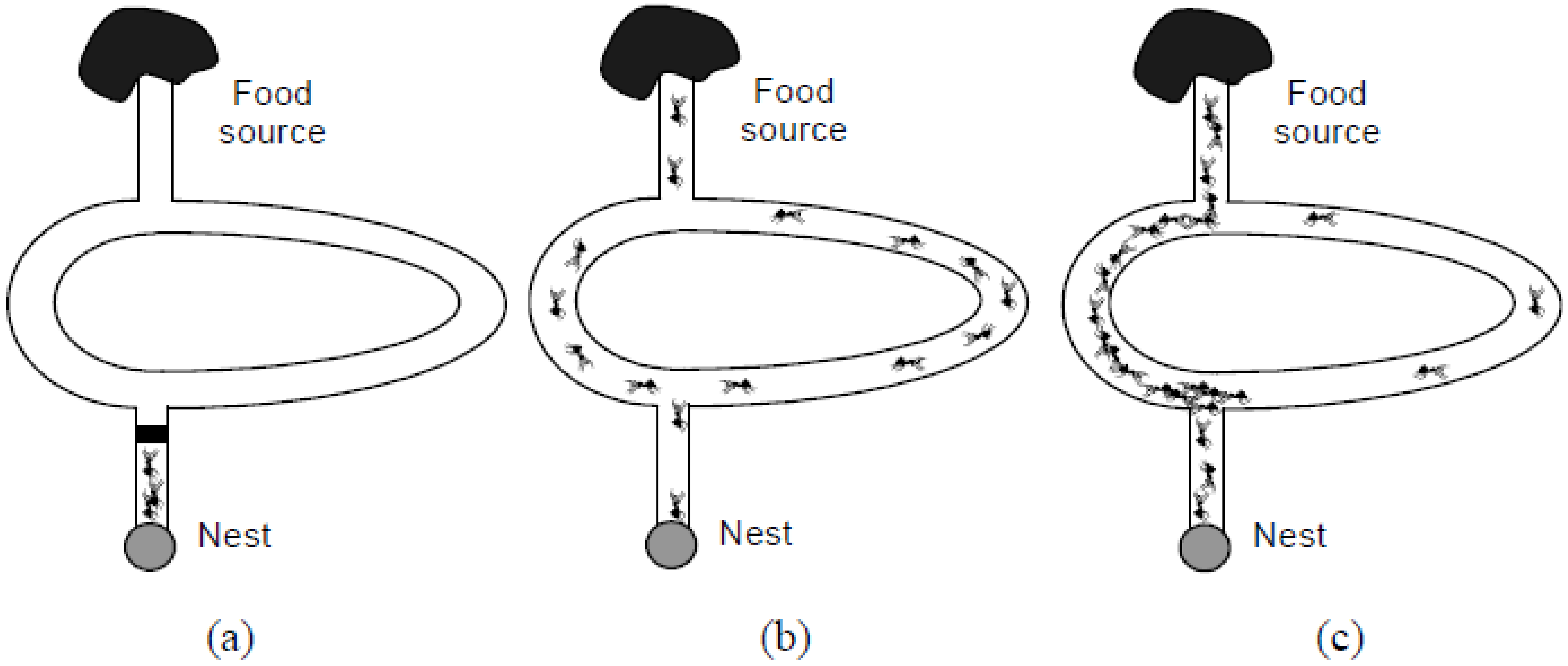

Figure 5.2: An experimental set up that can be used to demonstrate that the ant *I. Humilis* is capable of finding the shortest path between the nest and a food source. (b) The bridge is initially closed. (b) Initial distribution of ants after the bridge is open. (b) Distribution of ants after some time has passed since they were allowed to exploit the food source.

Experimental Observations

- After a short transient phase:
  - Majority of ants choose the **shorter path**
- Probability of choosing shorter path:
  - Increases with **difference in branch lengths**
- Demonstrates **collective optimization**
- Achieved without central control

Stigmergy in Ant Foraging

- Communication occurs via **environmental modification**
- Known as **stigmergy**
- Pheromone trails guide future ant behavior
- Ants indirectly influence each other’s decisions
- Core principle behind **Ant Colony Optimization (ACO)**

Role of Pheromone Evaporation

- Pheromone evaporates over time
- Longer paths:
  - Require more time
  - Lose pheromone faster
- Shorter paths:
  - Reinforced more frequently
  - Become dominant
- In ACO algorithms:
  - Evaporation is **intentionally increased**
  - Prevents premature convergence

Simple Ant Colony Optimization:

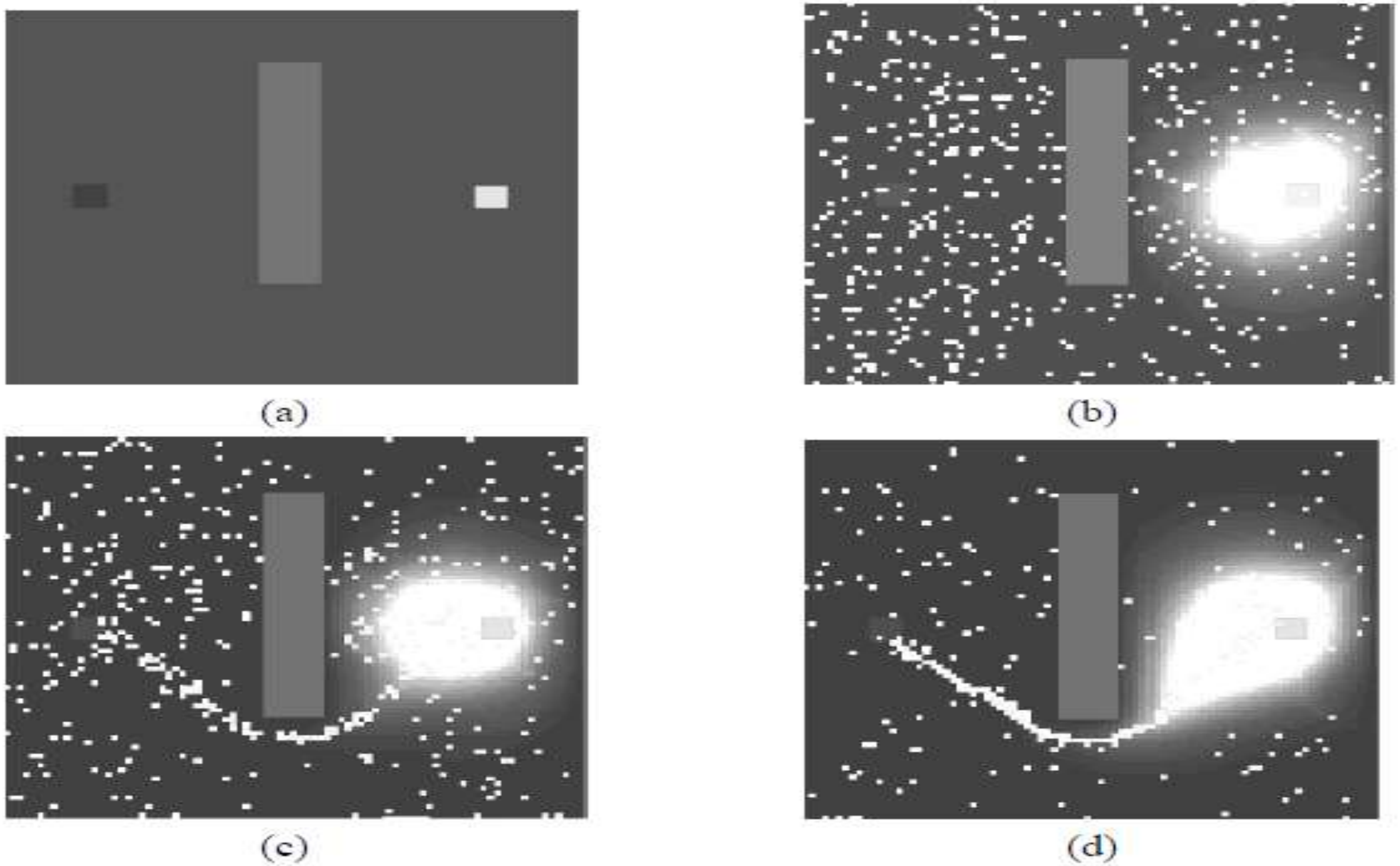

Figure 5.3: Artificial life simulation of pheromone trail laying and following by ants. (a) Environmental setup. The square on the left corresponds to the ants' nest, and the one on the right is the food source. In between the two there is an obstacle whose top part is slightly longer than the bottom part. (b) 500 ants leave the nest in search for food, and release pheromone (depicted in white color) while carrying food back to the nest. (c) The deposition of pheromone on the environment serves as a reinforcement signal to recruit other ants to gather food. (d) A strong pheromone trail in the shorter route is established.

Simple Ant Colony Optimization:

- **Problem Definition & Movement Rule (S-ACO)**
- Assume a **connected graph**

$$G = (V, E)$$

- Goal: find the **shortest path** between:
  - Source node s
  - Destination node d
- A solution = **path on the graph**
- Path length = **number of edges traversed**

Simple Ant Colony Optimization:

- Each ant:
  - Moves **one node per iteration**
  - Uses **local pheromone information**
- **Transition Probability Rule**:
- $p_{ij}^{k}(t) = \frac{\tau_{ij}(t)}{\sum_{l \in N_i} \tau_{il}(t)} \quad (5.1)$
- Where:
- $p_{ij}^{k}(t)$:probability ant k moves from node i to node j
- $\tau_{ij}(t)$:pheromone level on edge (i, j)
- N_i :one-step neighbors of node i

Simple Ant Colony Optimization:

- **Pheromone Update & Positive Feedback**
- Each edge (i, j)has a pheromone value τ_{ij}
- Ants can:
  - **Deposit pheromone** ("mark")
  - **Sense pheromone** ("smell")
- After traversing an edge, pheromone is updated:
- $\tau_{ij}(t) \leftarrow \tau_{ij}(t) + \Delta\tau \ (5.2)$
- Where:
- $\Delta\tau$: constant pheromone amount

Simple Ant Colony Optimization:

Effect:

- Increases probability of future ants choosing same edge
- Reinforcement of good paths
- Leads to positive feedback
- Naturally favors shorter paths
- Early experiments:
    - Successfully found shortest nest-to-food paths
    - Similar to real ant laboratory experiments

Simple Ant Colony Optimization:

- **Pheromone Evaporation & Stability Improvement**
- Problem observed:
  - Algorithm becomes **unstable** for complex graphs
  - Rapid convergence to **sub-optimal paths**
- Solution introduced:
  - **Pheromone evaporation**
- Modified pheromone update rule:
- $\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) + \Delta\tau \ (5.3)$
- Where:
- $\rho \in (0, 1]$:pheromone decay rate

Simple Ant Colony Optimization:

- **Pheromone Evaporation & Stability Improvement**

Benefits:

- Prevents unlimited pheromone accumulation
- Encourages exploration
- Avoids premature convergence
- Ensures balance between:
  - Exploitation (good paths)
  - Exploration (new paths)

General-Purpose Ant Colony Optimization (ACO)

- Ant Colony Optimization (ACO) is a class of **discrete optimization algorithms**
- Inspired by **foraging behavior of real ant colonies**
  - Dorigo et al. (1999)
- Primarily developed for **discrete optimization**
- Some extensions exist for **continuous optimization**
  - Example: Bilchev & Parmee (1995)
- Research in continuous ACO is limited
- Focus here: **ACO for discrete optimization problems**

```
procedure [best] = ACO(max_it,N,τ0)
    initialize τij   //usually initialized with the same τ0
    initialize best
    place each ant k on a randomly selected edge
    t ← 1
    while t < max_it do,
        for i = 1 to N do,   //for each ant
            build a solution by applying a probabilistic
            transition rule (e-1) times.
            //The rule is a function of τ and η
            //e is the number of edges on the graph G
        end for
        eval the cost of every solution built
        if an improved solution is found,
            then update the best solution found
        end if
        update pheromone trails
        t ← t + 1
    end while
end procedure
```

Algorithm 5.1: Standard ACO for discrete optimization.

Basic Structure of a General-Purpose ACO Algorithm

- ACO runs for a **maximum number of iterations**: max_it
- Search space modeled as a **connected graph**
  - $G = (V, E)$
  - V :nodes,
  - E: edges
- Each iteration consists of **two main procedures**:

1. Solution Construction / Modification
2. Pheromone Trail Update

Basic Structure of a General-Purpose ACO Algorithm

1. Solution Construction / Modification

- A colony of **N ants**
- Each ant builds or modifies **one solution**
- All ants work **in parallel**

2. Pheromone Trail Update

- Pheromone levels on edges are updated
- Based on ants' solutions

Probabilistic Decision Making & Pheromone Update

- Solution construction is **probabilistic**
- Probability of selecting an edge depends on:
  - **Pheromone trail (τ)** on the edge
  - **Heuristic desirability (η)** of the edge

Probabilistic Decision Making & Pheromone Update

Heuristic desirability η

- Represents how attractive an edge is
- Example (shortest path problems):
- η=1/distance

- Pheromone updating depends on:
  - Evaporation rate (ρ) (as in Eq. 5.3)
  - Quality of solutions produced
- Pheromone is updated only after all ants finish
- Best solution found so far is stored as best

Applications of ACO:

- **Traveling Salesman Problem (TSP)**
  - One of the earliest ACO applications
  - NP-hard problem
  - Shortest path formulation
- **Vehicle Routing Problem (VRP)**
  - Derived by small modifications of ACO-TSP
  - Demonstrates flexibility of ACO framework

Traveling Salesman Problem (TSP) – Problem Definition

- Goal: find a **closed tour of minimum length**
- Tour must:
  - Visit **each city exactly once**
  - Return to the starting city
- Given **e cities**
- Distance between cities i and j:
- $d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \quad (5.4)$
- Cities represented as **nodes**
- Connections represented as **edges**
- Problem modeled as a graph:
- $G = (V, E)$

```
procedure [best] = AS-TSP(max_it,α,β,ρ,N,e,Q,τ0,b)
    initialize τij   //usually initialized with the same τ0
    place each ant k on a randomly selected city
    Let best be the best tour found from the beginning and
    Lbest its length
    t ← 1
    while t < max_it do,
        for i = 1 to N do,              //for every ant
            //e is the number of cities on the graph
            build tour T^k(t) by applying (e−1) times the fol-
            lowing step:
            At city i, choose the next city j with probabil-
            ity given by Equation (5.5)
        end for
        eval the length of the tour performed by each ant
        if a shorter tour is found,
            then update best and Lbest
        end if
        for every city e do,
            Update pheromone trails by applying the rule:
            τij(t+1) ← (1−ρ)τij(t) + Δτij(t) + b.Δτ^b_ij(t), where
            Δτij(t) = Σk Δτ^k_ij(t), k = 1,...N;
```

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q / L^{k}(t) & \text{if } (i,j) \in T^{k}(t) \\ 0 & \text{otherwise} \end{cases}, \text{ and}$$

$$\Delta\tau_{ij}^{b}(t) = \begin{cases} Q / L_{best} & \text{if } (i,j) \in best \\ 0 & \text{otherwise} \end{cases}.$$

```
        end for
        t ← t + 1
    end while
end procedure
```

Algorithm 5.2: Ant system for the traveling salesman problem (AS-TSP).

Ant System (AS) for TSP

- AS is an ACO-based algorithm for TSP
- Colony of N ants
- Each ant:
  - Starts from a city
  - Moves from city to city
  - Builds a complete tour
- Tours are constructed using a probabilistic transition rule
- Iterative process:
  - Iteration counter: t
  - Maximum iterations: max_it
  - $1 \leq t \leq max\_it$

Transition Rule & Tabu List

- Ant decision depends on:
  - Whether the city is **already visited**
  - **Visibility** (inverse of distance):
- $\eta_{ij} = \frac{1}{d_{ij}}$
  - **Pheromone trail** $\tau_{ij}(t)$
- Each ant maintains a **tabu list**
  - Stores cities already visited
- J_i^k :set of cities ant k can still visit from city i

Transition Rule & Tabu List

- **Transition Probability Rule:**
- $$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta}, & j \in J_i^k \\ 0, & otherwise \end{cases} \quad (5.5)$$
- α :influence of pheromone
- β: influence of visibility
- Special cases:
  - $\alpha = 0$: greedy nearest-neighbor
  - $\beta = 0$: pheromone-only behavior

Pheromone Deposit by Ants

- While traversing edges, ants **deposit pheromone**
- Pheromone deposited by ant k on edge (i, j):
- $\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L_k(t)}, & (i, j) \in T_k(t) \\ 0, & \text{otherwise} \end{cases}$ (5.6)
- Where:
- $L_k(t)$:length of tour by ant k
- $T_k(t)$:tour performed by ant k
- Q: user-defined constant
- Better (shorter) tours → **more pheromone**

Global Pheromone Update & Elitist Ants

- **Pheromone Evaporation and Update Rule:**
- $\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$ (5.7)
- Where:
- $\rho \in (0, 1]$:evaporation rate
- $\Delta\tau_{ij}(t) = \sum_{k=1}^{N} \Delta\,\tau_{ij}^{k}(t)$
- **Number of Ants (N):**
- Too many ants → reinforce poor paths
- Too few ants → weak cooperation
- Suggested value:
- $N = e\ (number\ of\ cities)$

Global Pheromone Update & Elitist Ants

- **Elitist Ants Concept:**
- Elitist ants reinforce **best tour found so far**
- Extra pheromone added:
- $\frac{b \cdot Q}{L_{best}}$
- b :number of elitist ants
- L_{best} :length of best tour

Vehicle Routing Problem (VRP) – Problem Definition

- Given a **fleet of identical vehicles** with **uniform capacity**
- A **single common depot**
- Several **customer demands** (geographically scattered)
- Objective:
- Find a **set of routes with minimum total cost**
- Constraints:
- Each route **starts and ends at the depot**
- Each customer is:
  - Served **only once**
  - Served by **only one vehicle**

Vehicle Routing Problem (VRP) – Problem Definition

- VRP is closely related to **TSP**
- VRP ≈ solving **multiple TSPs**
- Common start/end node (depot)
- Additional **capacity constraints**

Graph Representation of VRP

- VRP represented as a **complete weighted directed graph**:
- $G = (V, E, w)$
- Nodes:
- $V = \{v_0, v_1, \ldots, v_N\}$
- v_0 :depot
- $v_1 \ldots v_N$:customers
- Edges:
- $E = \{(v_i, v_j) : i \neq j\}$

Graph Representation of VRP

- Weights:
  - $w_{ij} \geq 0$:distance / time / cost from $v_i$to v_j
- Customer attributes:
  - Demand: $d_i \geq 0$
  - Service time: $\delta_i \geq 0$
- Depot values:
- $d_0 = \delta_0 = 0$

VRP Constraints & Solution Construction

- **Constraints**
- Every customer visited **exactly once**
- Each vehicle route:
  - Total demand $\leq D$ (vehicle capacity)
  - Total route length (including service times) $\leq L$
- All routes begin and end at the **depot**

ACO-Based Solution Construction

- One ant is placed on **each node**
- Ants construct routes by:
  - Sequentially visiting customers
- If selecting a new city violates:
  - Capacity constraint D, or
  - Route length limit L
- Then:
  - Ant returns to the **depot**
  - A **new route** is started

Transition Rule & Heuristic Desirability

- VRP solved using **general-purpose ACO (Algorithm 5.1)**
- Ant movement governed by:
  - **Pheromone trail** τ_{ij}
  - **Heuristic desirability** η_{ij}
- Transition probability uses Equation (5.5) (same as AS-TSP)

Transition Rule & Heuristic Desirability

- **Heuristic Desirability (Savings Function)**

- $\eta_{ij} = w_{i0} + w_{0j} - g \cdot w_{ij} + f \cdot | w_{i0} - w_{0j} |$ *(5.8)*
- Where:
- w_{i0} :distance from customer i to depot
- w_{0j} :distance from depot to customer j
- w_{ij} :distance between customers
- g, f: user-defined parameters

Ranked Elitist Pheromone Update for VRP

- Standard AS:
  - All ants deposit pheromone
- Improved VRP-AS (Bullnheimer et al., 1999b):
  - **Only best-ranked ants (elitist ants)** update pheromone
- Pheromone update rule:
- $$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^{r}(t) + \sigma \cdot \Delta\tau_{ij}^{+}(t) \qquad (5.9)$$
- Where:
- ρ: evaporation rate
- σ: number of elitist ants

Rank-Based Pheromone Contribution

- $\Delta\tau_{ij}^{r}(t) = \sum_{\mu=1}^{\sigma-1} \Delta\, \tau_{ij}^{\mu}(t)$ *(5.10)*
- $\Delta\tau_{ij}^{\mu}(t) = \begin{cases} \frac{\sigma-\mu}{L_\mu(t)}, & \textit{if edge } (v_i, v_j) \textit{ is used} \\ 0, & \textit{otherwise} \end{cases}$
- $L_\mu(t)$:route length of the μ-th best ant
- Best-so-far route reinforced as if **σ elitist ants** used it:
- $\Delta\tau_{ij}^{+}(t) = \frac{1}{L^{+}}$

VRP Summary:

- VRP = multi-tour TSP with capacity & length constraints
- ACO adapts naturally using:
  - Feasibility checks
  - Savings-based heuristic
  - Ranked elitist pheromone updates
- Results in better scalability and performance than basic AS