University of London

Ant Colony Optimisation

Dr H.K. Lam

Department of Engineering

King's College London

Office S2.14, Strand Building, Strand Campus

Email: hak-keung.lam@kcl.ac.uk

https://nms.kcl.ac.uk/hk.lam

Nature-Inspired Learning Algorithms (7CCSMBIM)

Outline

Learning Objectives

King's College London, University of London

- To know the kind of problems that can be solved by *Ant Colony Optimisation* algorithms.
- To know how *Ant Colony Optimisation* algorithms work and their limitations.
- To apply *Ant Colony Optimisation* algorithms to shortest-path finding problems.

Introduction

Swarm Intelligence

- "Swarm intelligence is the property of a system whereby the collective behaviours of (unsophisticated) agents interacting locally with their environment cause coherent functional global patterns to emerge."

- "Swarm intelligence provides a basis with which it is possible to explore collective (or distributed) problem solving without centralised control or the provision of a global model."

Example:

- A group of fishes swim in the same direction.
- Ants work together to find food and haul back to the nest.

KING'S College LONDON
University of London

Characteristics of Social Colonies

- **Flexible:** The colony can respond to internal perturbations and external challenges.
- **Robust:** Tasks are completed even if some individuals fail.
- **Decentralised:** There is no central control in the colony.
- **Self-organised:** Paths to solutions are emergent rather than predefined.

Self-organisation

- A set of dynamical mechanisms whereby structures appear at the global level of a system from interactions of its lower-level components.
- **Four basic ingredients:**
  - **Positive feedback (amplification)**: To show the right of direction to the food source (optimal solution), to reinforce those portions of good solutions that contribute to the quality of these solutions.
  - **Negative feedback:** to introduce a time scale into the algorithm through pheromone evaporation, to prevent premature convergence (stagnation), for counter-balance and stabilisation
  - **Amplification of fluctuation:** Randomness or errors, e.g., lost ant foragers can find new food sources. An element moves more randomly to search for a solution and then amplified by a positive feedback loop.
  - **Multiple interactions:** Direct or indirect communication (e.g., modification of the environment).

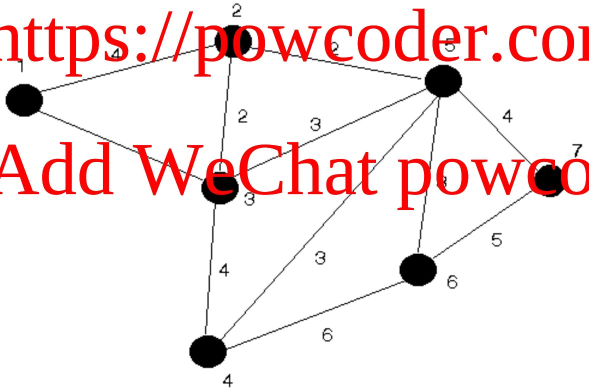
Ants

- Ants appeared on earth some 100 million years ago.
- Estimated total population: 10^{16} individuals.
- Social insects live in colonies of 30 to millions of individuals.
- Collective behaviours: foraging behaviour, division of labour, cooperative support, cemetery organisation, brood care, construction of nests, etc.
- Stimulus-response agents.
- Individual performs simple and basic action based on information of local information.
- Simple actions appear to have a large random component.

KING'S College LONDON
University of London

Ant Optimisation Algorithm

- To search for an optimal path in a graph.
- Find the shortest path between their nest and food source, without any visible, central, active coordination mechanisms.



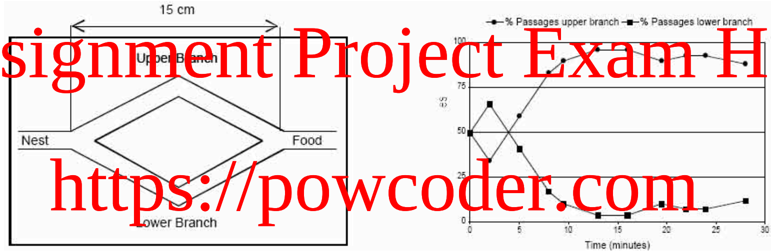
Notation

University of London

- $x^k(t)$: the solution of ant k, which is a set of nodes visited by ant k.
- $\tilde{x}(t)$: the current best path (the best solution among $x^k(t)$) at generation/iteration t, which is a set of nodes visited by the best ant.
- $\hat{x}(t)$: the global best path found from the first iteration to current iteration of the algorithm.
- $x^+(t)$: the best solution(s) giving the shortest path(s) (the iteration-best or global best ant(s)).
- t: generation/iteration number.
- ρ: evaporation rate.
- n_k: number of ants.
- n_e: number of elite ants.
- (i,j): an edge from node i to node j.
- $\tau_{ij}(t)$: pheromone concentration associated with edge (i,j) at generation/iteration t.
- $\Delta\tau_{ij}^k(t)$: the change of pheromone concentration associated with edge (i,j) at generation/iteration t.
- $\Delta\tau_{ij}^e(t)$: the change of pheromone concentration associated with edge (i,j) visited by the elite ants at generation/iteration t.
- $f(x^k(t))$: the quality of the solution of ant k.
- $f(\tilde{x}(t))$: the quality of the solution of the $\tilde{x}(t)$ (the best ant).
- $f(x^+(t))$: the cost(s) of the best solution(s) for $x^+(t)$ (the iteration-best or global best ant(s)).
- $L^k(t)$: length of the path (from the source to the destination) constructed by ant k.
- $d_{ij}(t)$: cos between edge (i,j). When t is dropped, d_{ij} is independent of generation/iteration t.
- $p_{ij}^k(t)$: transition probability of selecting the next node $j \in \mathcal{N}_i^k(t)$ by ant k and node i.
- $\mathcal{N}_i^k(t)$: the set of feasible nodes connected to node i, with respect to ant k.
- $Q > 0$: a non-zero positive constant.

The Binary Bridge Experiments

- A simple and elegant experiment to study of foraging behaviour of ants.



- Ants deposit chemical pheromone while walking.
- Ants have larger probability to follow path with higher pheromone trial.
- Probability of the next ant to choose path A:

$$P_A(t+1) = \frac{(c+n_A(t))^\alpha}{(c+n_A(t))^\alpha + (c+n_B(t))^\alpha} = 1 - P_B(t+1)$$

$n_A(t)$ and $n_B(t)$: Number of ants on paths A and B, respectively.

c: degree of attraction of an unexplored branch.

α: the bias to using pheromone deposits in the decision process

KING'S College LONDON
University of London

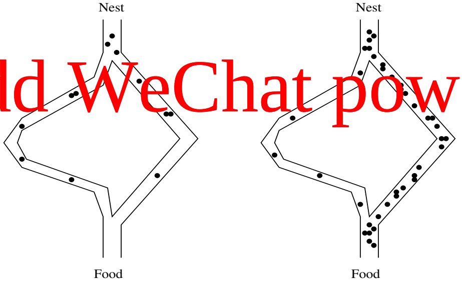
Artificial Ant Decision Process

Generate a random number $r \in [0,1]$;
Choose values of c and α;
for *each potential path A* **do**
    Calculate $P_A(t+1) = \frac{(c+n_A(t))^\alpha}{(c+n_A(t))^\alpha+(c+n_B)^\alpha}$
    if $r \le P_A$ **then**;
        Follow path A
        Break;
    end
end

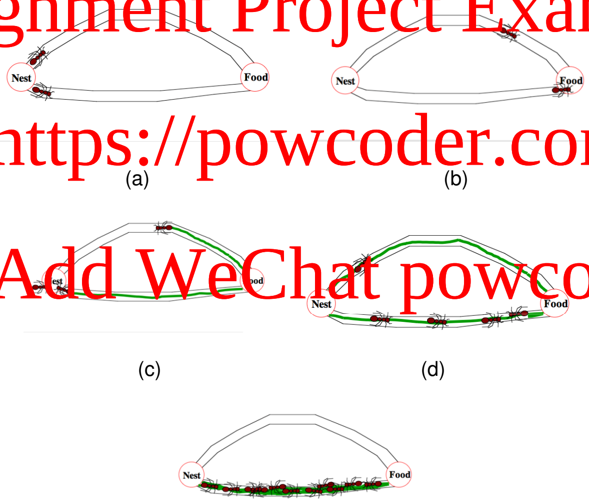
Table 1: Pseudo Code of Artificial Ant Decision Process.

Shortest path selection by forager ants

- Shortest path is selected.
- Ants return to the nest earlier.
- The pheromone on the shortest path is reinforced sooner (positive feedback)



Shortest path selection by forager ants



Stigmergy and Artificial Pheromone

- **Stigmergy** is a class of mechanisms that mediate animal-to-animal interactions.
  - A form of indirect communication mediated by modifications of the environment.
  - Some signs observed by individual trigger a specific response or action, which may reinforce or modify signals (positive or negative feedback) to influence actions of others

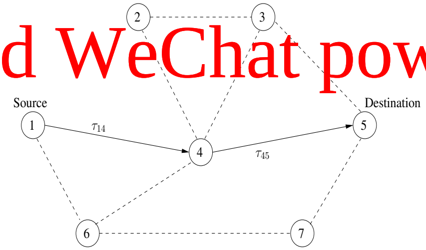
- **Two forms of stigmergy:** sematectonic and sign-based.
  - *Sematectonic stigmergy:* communication via changes in the physical characteristics of the environment.
  - *Sign-based stigmergy:* communication via a signalling mechanism, e.g., implemented via chemical compounds deposited by ants.

- **Artificial stigmergy:** "indirect communication mediated by numeric modifications of environmental states which are only locally accessible by the communicating agent". (Dorigo and Di Caro)
- *Artificial pheromone* plays the role of stigmergic variable, which encapsulate the information used by artificial ants to communicate indirectly.

Simple Ant Colony Optimisation (SACO)

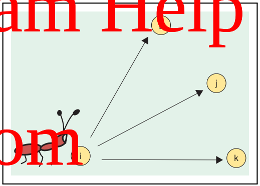
KING'S College LONDON
University of London

- (i,j): An edge from node i to node j.
- $\tau_{ij}(t)$: Pheromone concentration associated with edge (i,j) at generation/iteration t.
  - $\tau_{ij}(0)$ is assigned a small random value. Why?
- $L^k(t)$: Length of the path (from the source to the destination) constructed by ant k.



- **Transition probability** of selecting the next node $j \in \mathcal{N}_i^k(t)$ by the ant k sitting at node i (roulette wheel selection method):

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)}{\displaystyle\sum_{u \in \mathcal{N}_i^k(t)} \tau_{iu}^{\alpha}(t)} & \text{if } j \in \mathcal{N}_i^k(t) \\ 0 & \text{otherwise} \end{cases}, k = 1, \cdots, n_k$$



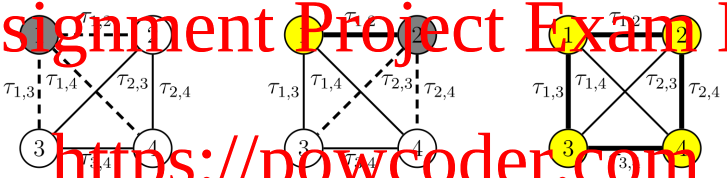
where $\mathcal{N}_i^k(t)$ is the set of feasible nodes connected to node i, with respect to the ant k; $\alpha > 0$ is a constant.

- If $\mathcal{N}_i^k(t) = \emptyset$, the predecessor to node i is included in $\mathcal{N}_i^k(t)$.
  - This may cause loops.
  - Loops are removed when the destination has been reached.

Example: Found path with a loop: 1–4–2–3–4–5

Path after removing a loop: 1–4–5

Example



Node 1: $\mathcal{N}_1^k(t) = \{2,3,4\}$	Node 2: $\mathcal{N}_2^k(t) = \{3,4\}$	Node 4: $\mathcal{N}_4^k(t) = \{3\}$
$p_{1,2}^k(t) = \frac{\tau_{1,2}^\alpha(t)}{\tau_{1,2}^\alpha(t)+\tau_{1,3}^\alpha(t)+\tau_{1,4}^\alpha(t)}$ $p_{1,3}^k(t) = \frac{\tau_{1,3}^\alpha(t)}{\tau_{1,2}^\alpha(t)+\tau_{1,3}^\alpha(t)+\tau_{1,4}^\alpha(t)}$ $p_{1,4}^k(t) = \frac{\tau_{1,4}^\alpha(t)}{\tau_{1,2}^\alpha(t)+\tau_{1,3}^\alpha(t)+\tau_{1,4}^\alpha(t)}$	$p_{2,3}^k(t) = \frac{\tau_{2,3}^\alpha(t)}{\tau_{2,3}^\alpha(t)+\tau_{2,4}^\alpha(t)}$ $p_{2,4}^k(t) = \frac{\tau_{2,4}^\alpha(t)}{\tau_{2,3}^\alpha(t)+\tau_{2,4}^\alpha(t)}$	$p_{4,3}^k(t) = \frac{\tau_{4,3}^\alpha(t)}{\tau_{4,3}^\alpha(t)} = 1$

Note: $\tau_{i,j}(t) = \tau_{j,i}(t)$

Example: Transition probability Table (as in the Binary Genetic Algorithm)

For node 1:

Next node j	Transition Probability $p^k_{1,j}(t)$	Accumulated Transition Probability
2	$\frac{\tau^\alpha_{1,2}(t)}{\tau^\alpha_{1,2}(t)+\tau^\alpha_{1,3}(t)+\tau^\alpha_{1,4}(t)}$	$\frac{\tau^\alpha_{1,2}(t)}{\tau^\alpha_{1,2}(t)+\tau^\alpha_{1,3}(t)+\tau^\alpha_{1,4}(t)}$
3	$\frac{\tau^\alpha_{1,3}(t)}{\tau^\alpha_{1,2}(t)+\tau^\alpha_{1,3}(t)+\tau^\alpha_{1,4}(t)}$	$\frac{\tau^\alpha_{1,2}(t)+\tau^\alpha_{1,3}(t)}{\tau^\alpha_{1,2}(t)+\tau^\alpha_{1,3}(t)+\tau^\alpha_{1,4}(t)}$
4	$\frac{\tau^\alpha_{1,4}(t)}{\tau^\alpha_{1,2}(t)+\tau^\alpha_{1,3}(t)+\tau^\alpha_{1,4}(t)}$	$\frac{\tau^\alpha_{1,2}(t)+\tau^\alpha_{1,3}(t)+\tau^\alpha_{1,4}(t)}{\tau^\alpha_{1,2}(t)+\tau^\alpha_{1,3}(t)+\tau^\alpha_{1,4}(t)}=1$

Assume that $\alpha = 1$, $\tau_{1,2} = 0.5$, $\tau_{1,3} = 0.3$ and $\tau_{1,4} = 0.2$.

Next node j	Transition Probability $p^k_{1,j}(t)$	Accumulated Transition Probability
2	$\frac{0.5}{0.5+0.3+0.2}=0.5$	0.5
3	$\frac{0.3}{0.5+0.3+0.2}=0.3$	0.5 + 0.3 = 0.8
4	$\frac{0.3}{0.5+0.3+0.2}=0.2$	0.5 + 0.3 + 0.2 = 1

Generate a random number, say, $r = 0.6$. Node 3 is chosen as 0.6 is lying in between 0.5 and 0.8.

Simple Ant Colony Optimisation (SACO)

King's College London
University of London

Evaporation of Pheromone Intensity (negative feedback)

- Pheromone intensity will evaporate.
  - To force ants to explore more.
  - To prevent premature convergence.
- For each edge (i, j), pheromone intensity is reduced according to

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

where $\rho \in (0, 1)$ (0 and 1 are not inclusive) is the evaporation rate.

KING'S College LONDON
University of London

Update of Pheromone Intensity (positive feedback)

- After all ants have constructed their paths from the source to the destination, and all loops are removed, the pheromone intensity on edge (i,j) is adjusted:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t)$$

where

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{f(x^k(t))} & \text{if edge } (i,j) \text{ occurs in path } x^k(t) \\ 0 & \text{otherwise} \end{cases}$$

$x^k(t)$ is the solution of ant k,

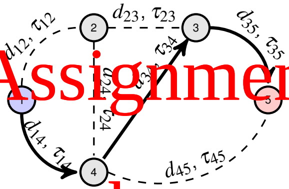
$f(x^k(t))$ is the quality of the solution,

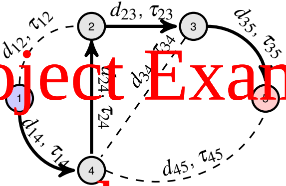
$Q > 0$ is a constant,

n_k is the number of ants.

Simple Ant Colony Optimisation (SACO)

Example: Source node: 1; Target node: 5 ($d_{ij} = d_{ji}$, $\tau_{ij} = \tau_{ji}$ and n_k = 2)

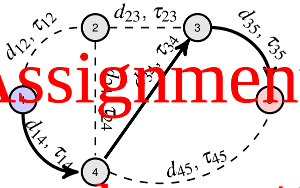


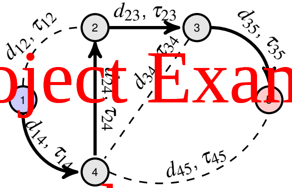


- Ant 1: $x^1(t) = \{1,4,3,5\}$
- $f(x^1(t)) = d_{14} + d_{34} + d_{35}$

- Ant 2: $x^2(t) = \{1,4,2,3,5\}$
- $f(x^2(t)) = d_{14} + d_{42} + d_{23} + d_{35}$

Example: Source node: 1; Target node: 5 ($d_{ij} = d_{ji}$, $\tau_{ij} = \tau_{ji}$ and $n_k = 2$)





- Ant 1: $x^1(t) = \{1, 4, 3, 5\}$
- $f(x^1(t)) = d_{14} + d_{34} + d_{35}$
- Ant 2: $x^2(t) = \{1, 4, 2, 3, 5\}$
- $f(x^2(t)) = d_{14} + d_{42} + d_{23} + d_{35}$

Pheromone update:

- Pheromone evaporation: $\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t)$ where $\rho \in (0, 1)$
- Update according to Ants' solutions
  - $\tau_{14}(t+1) = \tau_{14}(t) + \underbrace{\frac{Q}{f(x^1(t))}}_{\text{Ant 1}} + \underbrace{\frac{Q}{f(x^2(t))}}_{\text{Ant 2}}$; $\quad \tau_{35}(t+1) = \tau_{35}(t) + \underbrace{\frac{Q}{f(x^1(t))}}_{\text{Ant 1}} + \underbrace{\frac{Q}{f(x^2(t))}}_{\text{Ant 2}}$;

    $\tau_{23}(t+1) = \tau_{23}(t) + \underbrace{\frac{Q}{f(x^2(t))}}_{\text{Ant 2}}$; $\quad \tau_{24}(t+1) = \tau_{24}(t) + \underbrace{\frac{Q}{f(x^2(t))}}_{\text{Ant 2}}$; $\quad \tau_{34}(t+1) = \tau_{34}(t) + \underbrace{\frac{Q}{f(x^1(t))}}_{\text{Ant 1}}$;

    otherwise $\tau_{ij}(t+1) = \tau_{ij}(t)$

Simple Ant Colony Optimisation Algorithm

Initialise $\tau_{ij}(0)$ to small random values; Let $t = 0$;
Place n_k ants on the origin node;

while STOP-CRIT **do**
- **for** *each ant* $k = 1, \cdots, n_k$ **do**
  - $x^k(t) = \emptyset$;
  - **While** *destination has not been reached* **do**
    - Select next node based on translation probability $p_{ij}^k(t)$;
    - Add (i,j) to path $x^k(t)$;
  - **end**
  - Remove all loops from $x^k(t)$;
  - Calculate $f(x^k(t))$;
- **end**
- **for** *each edge* (i,j) *of the graph* **do**
  - Reduce the pheromone, $\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t)$;
- **end**
- **for** *each edge* (i,j) *of the graph* **do**
  - Update $\tau_{ij}(t)$ i.e. $\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t)$;
- **end**
- $t \leftarrow t+1$;

end

Table 2: Pseudo Code of Simple Ant Colony Optimisation Algorithm.

Simple Ant Colony Optimisation Algorithm

Initialise $\tau_{ij}(0)$ to small random values; Let $t = 0$;
Place n_k ants on the origin node;

while STOP-CRIT **do**
  for *each ant* $k = 1, \cdots, n_k$ **do**
    $x^k(t) = 0$;
    While *destination has not been reached* **do**
      Select next node based on translation probability $p_{ij}^k(t)$;
      Add (i,j) to path $x^k(t)$;
    end
    Remove all loops from $x^k(t)$;
    Calculate $f(x^k(t))$;
  end
  for *each edge* (i,j) *of the graph* **do**
    Reduce the pheromone, $\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t)$;
  end
  for *each edge* (i,j) *of the graph* **do**
    Update $\tau_{ij}(t)$ i.e. $\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t)$;
  end
  $t \leftarrow t+1$;
end

Table 2: Pseudo Code of Simple Ant Colony Optimisation Algorithm.

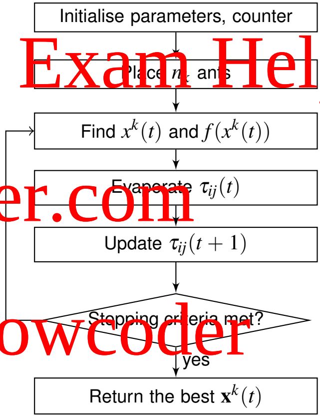


Figure 1: Flowchart of Simple Ant Colony Optimisation Algorithm.

Stopping Criteria:

- a maximum number of iterations has been exceeded.
- an acceptable solution has been found.
- all (or most) ants follow the same path.

Ant System (AS)

Ant System was developed based on SACO.

Improvements:

- Includes **heuristic information** to transition probability $p_{ij}^{k}(t)$.
- Includes a **tabu list** to the set of feasible nodes, $\mathcal{N}_i^k(t)$

  - May include only the immediate neighbours of node i.
  - May include all nodes not yet visited by ant k to prevent loops.
- **Different update strategies** for pheromone intensity.

- **Elitism** is implemented.

King's College London, University of London

Transition Probability (two methods):

- **Method 1:**

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{u \in \mathcal{N}_i^k(t)} \tau_{iu}^{\alpha}(t)\eta_{iu}^{\beta}(t)} & \text{if } j \in \mathcal{N}_i^k(t) \\ 0 & \text{otherwise} \end{cases}, k = 1, \cdots, n_k$$

  - $\tau_{ij}(t)$: pheromone intensity/concentration.
  - $\eta_{ij}(t)$: a priori effectiveness of the move from i to j (i.e. the attractiveness, or desirability, of the move).
  - $\alpha > 0$, $\beta > 0$: predefined constants.
- $\eta_{ij}(t) = \frac{1}{d_{ij}(t)}$ improves the attractiveness of the edge (i,j).
  - $d_{ij}(t)$: cost between edge (i,j).

Transition Probability (two methods):

- **Method 2:**

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\alpha\tau_{ij}(t)+(1-\alpha)\eta_{ij}(t)}{\displaystyle\sum_{u\in\mathcal{N}_i^k(t)} \alpha\tau_{iu}(t)+(1-\alpha)\eta_{iu}(t)} & \text{if } j \in \mathcal{N}_i^k(t) \\ 0 & \text{otherwise} \end{cases}, k = 1,\cdots,n_k; \alpha \in [0,1]$$

Pheromone evaporation:

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

Update of pheromone intensity/concentration:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^{k}(t), k = 1, \cdots, n_k$$

Ant System (AS)

King's College London
University of London

Update of pheromone intensity/concentration:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^{k}(t), k = 1, \cdots, n_k$$

- **Ant-cycle AS:** $\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{f(x^k(t))} \text{ if edge } (i,j) \text{ occurs in path } x^k(t) \\ 0 \text{ otherwise} \end{cases}$
  - $Q > 0$ is a constant.

- **Ant-density AS:** $\Delta\tau_{ij}^{k}(t) = \begin{cases} Q \text{ if edge } (i,j) \text{ occurs in path } x^k(t) \\ 0 \text{ otherwise} \end{cases}$

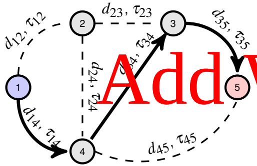
- **Ant-quantity AS:** $\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{d_{ij}(t)} \text{ if edge } (i,j) \text{ occurs in path } x^k(t) \\ 0 \text{ otherwise} \end{cases}$
  - $d_{ij}(t)$: cost between edge (i,j).

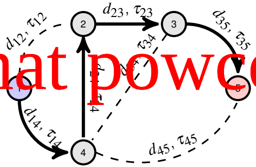
Elitist Strategy:

- $\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^{k}(t) + n_e \Delta\tau_{ij}^{e}(t)$
  - $\tilde{x}(t)$: the current best path (solution) at generation/iteration t.
  - $f(\tilde{x}(t)) = \min_{k=1,\cdots,n_k} f(x^k(t))$.
  - n_e: number of elite ants.
  - $\Delta\tau_{ij}^{e}(t) = \begin{cases} \frac{Q}{f(\tilde{x}(t))} & \text{if edge } (i,j) \in \tilde{x}(t) \\ 0 & \text{otherwise} \end{cases}$

Elitist Strategy:

- $\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^{k}(t) + n_e \Delta\tau_{ij}^{e}(t)$
  - $\tilde{x}(t)$: the current best path (solution) at generation/iteration t.
  - $f(\tilde{x}(t)) = \min_{k=1,\cdots,n_k} f(x^k(t))$.
  - n_e: number of elite ants.
  - $\Delta\tau_{ij}^{e}(t) = \begin{cases} \frac{Q}{f(\tilde{x}(t))} & \text{if edge } (i,j) \in \tilde{x}(t) \\ 0 & \text{otherwise} \end{cases}$





- Ant 1: $x^1(t) = \tilde{x}(t) = \{1,4,3,5\}$ (Assume this is the shortest path)
- $f(x^1(t)) = f(\tilde{x}(t)) = d_{14} + d_{34} + d_{35}$

- Ant 2: $x^2(t) = \{1,4,2,3,5\}$
- $f(x^2(t)) = d_{14} + d_{42} + d_{23} + d_{35}$

Ant System Algorithm

Initialise $\tau_{ij}(0)$ to small random values, and α, β, ρ, Q, $\eta_{ij}(t)$; Let $t = 0$;
Place n_k ants on the origin node;
while STOP-CRIT **do**
  for *each ant* $k = 1, \cdots, n_k$ **do**
    $x^k(t) = \emptyset$;
    While *destination has not been reached* **do**
      Select next node based on translation probability $p_{ij}^k(t)$;
      Add (i,j) to path $x^k(t)$;
    end
    Remove all loops from $x^k(t)$ if tabu list is not used;
    Calculate $f(x^k(t))$;
  end
  for *each edge* (i,j) *of the graph* **do**
    Reduce the pheromone, $\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t)$;
  end
  for *each edge* (i,j) *of the graph* **do**
    if elitist strategy is not implemented **then**
      $$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t);$$
    else
      $$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t) + n_e \Delta\tau_{ij}^e(t);$$
    end
  end
  $t \leftarrow t+1$;
end

Table 3: Pseudo Code of Ant System Algorithm.

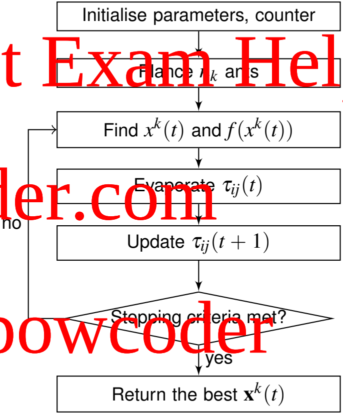
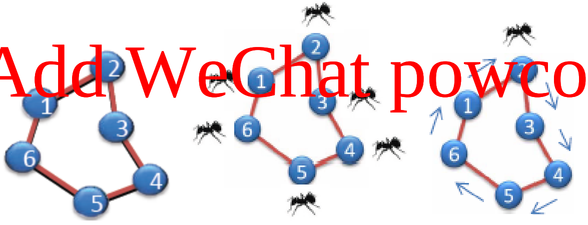


Figure 2: Flowchart of Ant System Algorithm.

Examples

Example 1 (travelling salesman problem (TSP)): Given a set of n cities, TSP requires a salesman to find the shortest route to return to the starting city while each city can be visited only once.

1. Place ants at different nodes.
2. Find the path for each ant.
3. Update pheromone intensity.
4. Go to step 2 until stopping criteria have be satisfied.

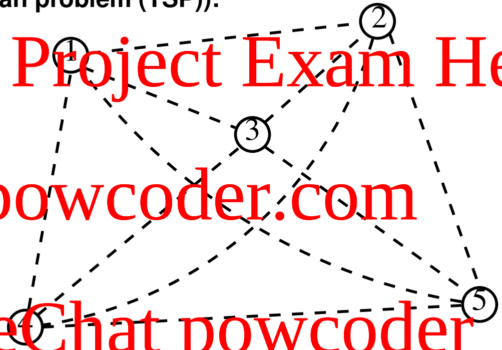


(a) (b) (c)

Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i,j)

$i \backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10	–	2	20	9
3	3	2	–	12	7
4	6	20	2	–	15
5	18	9	7	15	–



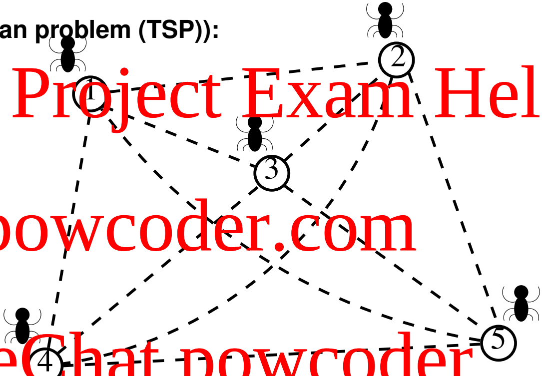
Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i, j)

$i \backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10	–	2	20	9
3	3	2	–	12	7
4	6	20	2	[illegible]	15
5	18	9	7	15	–

$\tau_{ij}(t)$ for edge (i, j)

$i \backslash j$	1	2	3	4	5
1	–	0.3	1.2	0.8	0.1
2	0.3	–	1.5	0.1	0.7
3	1.2	1.5	–	0.9	0.5
4	0.8	0.1	0.9	–	0.2
5	0.1	0.7	0.5	0.2	–



$x^1(t) = \{1\}$

$x^2(t) = \{2\}$

$x^3(t) = \{3\}$

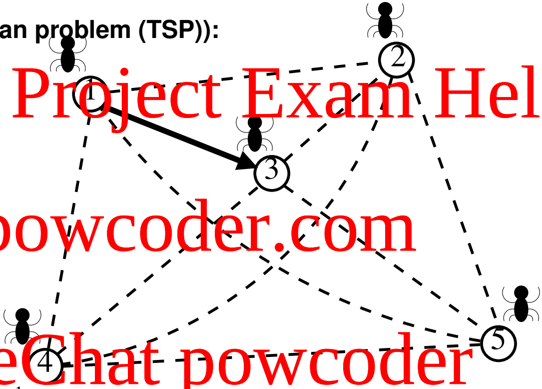
$x^4(t) = \{4\}$

$x^5(t) = \{5\}$

Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10	–	2	20	9
3	3	2	–	12	7
4	6	20	12	–	15
5	18	9	7	15	–



$\tau_{ij}(t)$ for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	0.3	1.2	0.8	0.1
2	0.3	–	1.5	0.1	0.7
3	1.2	1.5	–	0.9	0.5
4	0.8	0.1	0.9	–	0.2
5	0.1	0.7	0.5	0.2	–

$x^1(t) = \{1,3\}$

$x^2(t) = \{2,1\}$

$x^3(t) = \{3,4\}$

$x^4(t) = \{4,2\}$

$x^5(t) = \{5,4\}$

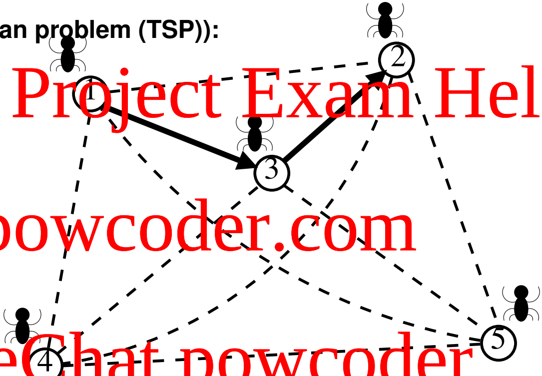
Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10	–	2	20	9
3	3	2	–	12	7
4	6	20	12	–	15
5	18	9	7	15	–

$\tau_{ij}(t)$ for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	0.3	1.2	0.8	0.1
2	0.3	–	1.5	0.1	0.7
3	1.2	1.5	–	0.9	0.5
4	0.8	0.1	0.9	–	0.2
5	0.1	0.7	0.5	0.2	–



$x^1(t) = \{1,3,2\}$

$x^2(t) = \{2,1,3\}$

$x^3(t) = \{3,4,2\}$

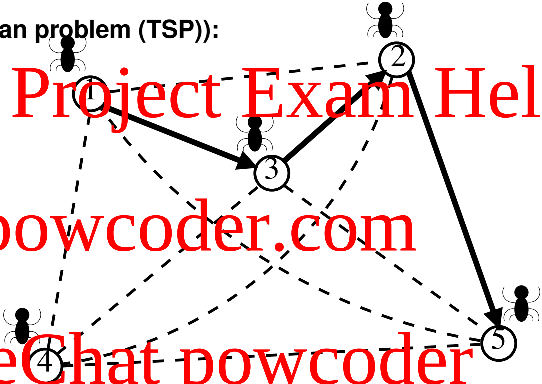
$x^4(t) = \{4,2,5\}$

$x^5(t) = \{5,4,1\}$

Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i,j)

$i \backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10	–	2	20	9
3	3	2	–	12	7
4	6	20	12	–	15
5	18	9	7	15	–



$\tau_{ij}(t)$ for edge (i,j)

$i \backslash j$	1	2	3	4	5
1	–	0.3	1.2	0.8	0.1
2	0.3	–	1.5	0.1	0.7
3	1.2	1.5	–	0.9	0.5
4	0.8	0.1	0.9	–	0.2
5	0.1	0.7	0.5	0.2	–

$x^1(t) = \{1,3,2,5\}$

$x^2(t) = \{2,1,3,5\}$

$x^3(t) = \{3,4,2,5\}$

$x^4(t) = \{4,2,5,1\}$

$x^5(t) = \{5,4,1,2\}$

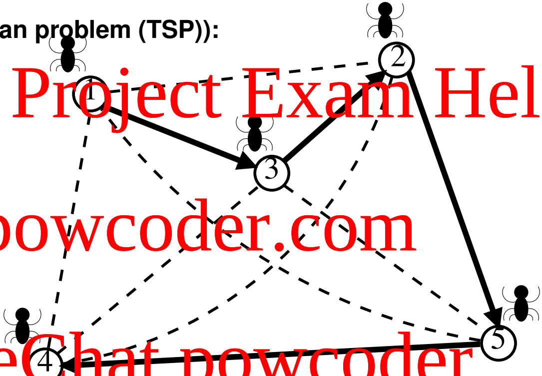
Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10	–	2	20	9
3	3	2	–	12	7
4	6	20	12	–	15
5	18	9	7	15	–

$\tau_{ij}(t)$ for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	0.3	1.2	0.8	0.1
2	0.3	–	1.5	0.1	0.7
3	1.2	1.5	–	0.9	0.5
4	0.8	0.1	0.9	–	0.2
5	0.1	0.7	0.5	0.2	–



$$x^1(t) = \{1,3,2,5,4\}$$
$$x^2(t) = \{2,1,3,5,4\}$$
$$x^3(t) = \{3,4,2,5,1\}$$
$$x^4(t) = \{4,2,5,1,3\}$$
$$x^5(t) = \{5,4,1,2,3\}$$

Examples

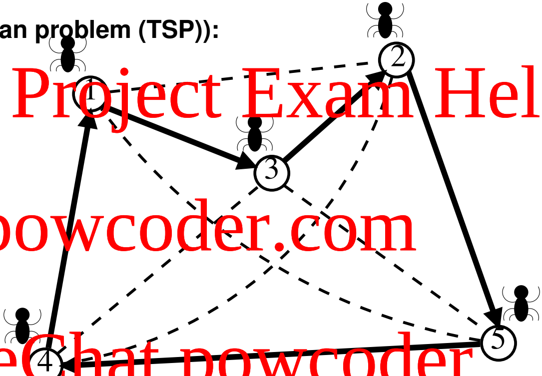
Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10	–	2	20	9
3	3	2	–	12	7
4	6	20	2	–	15
5	18	9	7	15	–

$\tau_{ij}(t)$ for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	0.3	1.2	0.8	0.1
2	0.3	–	1.5	0.1	0.7
3	1.2	1.5	–	0.9	0.5
4	0.8	0.1	0.9	–	0.2
5	0.1	0.7	0.5	0.2	–



$x^1(t) = \{1,3,2,5,4,1\}, f(x^1(t)) = d_{13} + d_{32} + d_{25} + d_{54} + d_{41} = 35$

$x^2(t) = \{2,1,3,5,4,2\}, f(x^2(t)) = d_{21} + d_{13} + d_{35} + d_{54} + d_{42} = 55$

$x^3(t) = \{3,4,2,5,1,3\}, f(x^3(t)) = d_{34} + d_{42} + d_{25} + d_{51} + d_{13} = 62$

$x^4(t) = \{4,2,5,1,3,4\}, f(x^4(t)) = d_{42} + d_{25} + d_{51} + d_{13} + d_{34} = 62$

$x^5(t) = \{5,4,1,2,3,5\}, f(x^5(t)) = d_{54} + d_{41} + d_{12} + d_{23} + d_{35} = 40$

KING'S College LONDON
University of London

Example 1 (travelling salesman problem (TSP)):

- Evaporation of Pheromone Intensity (negative feedback)

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

  - e.g., $\rho = 0.2$, $\tau_{54}(t) = (1-0.2)\tau_{54}(t) = 0.8 \times 0.2 = 0.16$

- Update of Pheromone Intensity (positive feedback)

SACO: $\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t)$, $\quad \Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{f(x^k(t))} & \text{if edge } (i,j) \text{ occurs in path } x^k(t) \\ 0 & \text{otherwise} \end{cases}$

AS: $\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t) + n_e \Delta\tau_{ij}^e(t)$

  - SACO: e.g., edge $(5,4)$, $n_k = 5$ (only ants 1, 2, 5 passed through), $Q = 1$,
    $\tau_{54}(t+1) = 0.16 + \frac{1}{35} + \frac{1}{55} + \frac{1}{40} = 0.2318$
  - AS: e.g., edge $(5,4)$, Ant-quantity AS, elitism is implemented, $Q = 1$, $n_e = 1$,
    $\tau_{54}(t+1) = 0.16 + \frac{1}{15} + \frac{1}{15} + \frac{1}{15} + 1 \times \frac{1}{35} = 0.3886$

Ant Colony System (ACS)

Ant Colony System was developed based on AS.

Improvements:

- Different transition rule, $p_{ij}^{k}(t)$.
  - Candidate lists are used to favour specific nodes.
- Different pheromone update rule, $\tau_{ij}(t)$.
  - Local and global pheromone update rules.
  - *Local update rule: pheromone evaporation.*

Ant Colony System (ACS)

Transition Probability: The k-th ant moving from node i to node j is according to

$$j = \begin{cases} \arg \max\limits_{u \in \mathcal{N}_i^k(t)} \{\tau_{iu}(t)\eta_{iu}^{\beta}(t)\} & \text{if } r \leq r_0 \\ J & \text{otherwise} \end{cases} \tag{1}$$

- $r \in [0, 1]$: a random number.
- $r_0 \in [0, 1]$: user-specified parameter; used to balance exploration and exploitation.
- $\mathcal{N}_i^k(t)$: a set of valid nodes to be visited by the k-th ant sitting at node i.
- $J \in \mathcal{N}_i^k(t)$ is a node randomly selected according to the probability:

$$p_{iJ}^k(t) = \begin{cases} \dfrac{\tau_{iJ}(t)\eta_{iJ}^{\beta}(t)}{\sum\limits_{u \in \mathcal{N}_i^k(t)} \tau_{iu}(t)\eta_{iu}^{\beta}(t)} & \text{if } J \in \mathcal{N}_i^k(t) \\ 0 & \text{otherwise} \end{cases}, k = 1, \cdots, n_k.$$

- $r \leq r_0$: the algorithm exploits by favouring the best edge.
- $r > r_0$: the algorithm explores.

Example:

- Current node: $i = 1$
- Ant 4, i.e., $k = 4$

Example:

- Current node: $i = 1$
- Ant 4, i.e., $k = 4$
- A set of valid nodes for ant 4: $\mathcal{N}_1^4(t) = \{2, 3, 7\}$
- For ant 4: $j = \begin{cases} \arg\max\{\tau_{1,2}(t)\eta_{1,2}^{\beta}(t), \tau_{1,3}(t)\eta_{1,3}^{\beta}(t), \tau_{1,7}(t)\eta_{1,7}^{\beta}(t)\} & \text{if } r \leq r_0 \\ J & \text{otherwise} \end{cases}$
  - Assume $r_0 = 0.2$, $r = 0.1$.
  - Assume $\eta_{1,2}^{\beta}(t) = \eta_{1,3}^{\beta}(t) = \eta_{1,7}^{\beta}(t)$; $\tau_{1,2}(t) = 0.22$, $\tau_{1,3}(t) = 0.33$, $\tau_{1,7}(t) = 0.44$, which node is chosen?

KING'S College LONDON
University of London

Example:

- Current node: $i = 1$
- Ant 4, i.e., $k = 4$
- A set of valid nodes for ant 4: $\mathcal{N}_1^4(t) = \{2, 3, 7\}$
- For ant 4: $j = \begin{cases} \arg\max\{\tau_{1,2}(t)\eta_{1,2}^{\beta}(t), \tau_{1,3}(t)\eta_{1,3}^{\beta}(t), \tau_{1,7}(t)\eta_{1,7}^{\beta}(t)\} \text{ if } r \leq r_0 \\ J \text{ otherwise} \end{cases}$
  - Assume $r_0 = 0.2$, $r = 0.1$.
  - Assume $\eta_{1,2}^{\beta}(t) = \eta_{1,3}^{\beta}(t) = \eta_{1,7}^{\beta}(t)$; $\tau_{1,2}(t) = 0.22$, $\tau_{1,3}(t) = 0.33$, $\tau_{1,7}(t) = 0.44$, which node is chosen?

- For ant 4: $p_{1J}^4(t) = \begin{cases} \dfrac{\tau_{1J}(t)\eta_{1J}^{\beta}(t)}{\tau_{1,2}(t)\eta_{1,2}^{\beta}(t)+\tau_{1,3}(t)\eta_{1,3}^{\beta}(t)+\tau_{1,7}(t)\eta_{1,7}^{\beta}(t)} & , J \in \mathcal{N}_1^4(t) = \{2, 3, 7\} \\ 0 \text{ otherwise} \end{cases}$
  - Probability of choosing nodes 2, 3, 7 for ant 4 (sitting at node 1 currently) are $p_{12}^4(t)$, $p_{13}^4(t)$ and $p_{17}^4(t)$, respectively.
  - Probability is 0 for choosing nodes other than nodes 2, 3, 7.

KING'S College LONDON
University of London

Local and Global Update Rules:

- **Local update rule:** Pheromone concentrations are updated for **all edges** (evaporation).

$$\tau_{ij}(t) \leftarrow (1-\rho_L)\tau_{ij}(t) + \rho_L \tau_0$$

Ant Colony System (ACS)

Local and Global Update Rules:

- **Local update rule:** Pheromone concentrations are updated for **all edges** (evaporation).

$$\tau_{ij}(t) \leftarrow (1-\rho_L)\tau_{ij}(t) + \rho_L \tau_0$$

  - $\rho_L \in (0,1)$ (0 and 1 are not inclusive): a user-specified parameter.
  - $\tau_0 > 0$: a small constant specified by user.

Why 0 and 1 are not allowed in ρ_L?

Why τ_0 is not allowed to be 0?

Local and Global Update Rules:

- **Global update rule:** Reinforcement of pheromone concentrations is allowed on the edges of the **best** path.

$$\tau_{ij}(t+1) = (1-\rho_G)\tau_{ij}(t) + \rho_G \Delta\tau_{ij}(t)$$

Local and Global Update Rules:

- **Global update rule:** Reinforcement of pheromone concentrations is allowed on the edges of the **best** path.

$$\tau_{ij}(t+1) = (1-\rho_G)\tau_{ij}(t) + \rho_G \Delta\tau_{ij}(t)$$

- $\Delta\tau_{ij}(t) = \begin{cases} \frac{1}{f(x^+(t))} & \text{if } (i,j) \in x^+(t) \\ 0 & \text{otherwise} \end{cases}$.
- $\rho_G \in (0,1)$ (0 and 1 are not inclusive, why?): a user-specified parameter.

- $x^+(t)$: the best solution(s) giving the shortest path(s).
  - **iteration-best strategy:** $x^+(t)$ represents the best path found during the current generation/iteration t, denoted as $\tilde{x}(t)$.
  - **global-best strategy:** $x^+(t)$ represents the best path found from the first iteration to the current generation/iteration t of the algorithm, denoted as $\hat{x}(t)$.
- $f(x^+(t))$ denotes the cost(s) of the best solution(s).

KING'S College LONDON
University of London

Ant Colony System Algorithm

Initialise $\tau_{ij}(0)$ to small random values, and β, ρ_L, ρ_G, r_0, τ_0, $\eta_{ij}(t)$; Let $t = 0$;
Place n_k ants on the origin node; Initialize the best solution $x^+(t) = \{\}$, $f(x^+(t)) = 0$
while STOP-CRIT **do**
  for *each ant* $k = 1, \cdots, n_k$ **do**
    $x^k(t) = \emptyset$;
    While *destination has not been reached* **do**
      Select next node based on equation (1);
      Add (i,j) to path $x^k(t)$;
    end
    Remove all loops from $x^k(t)$ if tabu list is not used;
    Calculate $f(x^k(t))$;
  end
  for *each edge* (i,j) *of the graph* **do**
    Apply local update rule: $\tau_{ij}(t) \leftarrow (1-\rho_L)\tau_{ij}(t) + \rho_L \tau_0$;
  end
  Update the global best solution $\hat{x}(t)$ and its cost $f(\hat{x}(t))$;
  for *each edge* $(i,j) \in x^+(t)$ **do**
    Update global update rule: $\tau_{ij}(t+1) = (1-\rho_G)\tau_{ij}(t) + \rho_G \Delta\tau_{ij}(t)$;
  end
  $t \leftarrow t+1$;
end

Table 4: Pseudo Code of Ant Colony System Algorithm.

Ant Colony System (ACS)

Ant Colony System Algorithm

Initialise $\tau_{ij}(0)$ to small random values, and β, ρ_L, ρ_G, r_0, τ_0, $\eta_{ij}(t)$; Let $t = 0$;
Place n_k ants on the origin node; initialize the best solution $x^+(t) = \{\}$, $f(x^+(t)) = 0$
while STOP-CRIT **do**
  for *each ant* $k = 1, \cdots, n_k$ **do**
    $x^k(t) = \emptyset$;
    While *destination has not been reached* **do**
      Select next node based on equation (1);
      Add (i,j) to path $x^k(t)$;
    end
    Remove all loops from $x^k(t)$ if tabu list is not used;
    Calculate $f(x^k(t))$;
  end
  for *each edge* (i,j) *of the graph* **do**
    Apply local update rule: $\tau_{ij}(t) \leftarrow (1-\rho_L)\tau_{ij}(t) + \rho_L \tau_0$;
  end
  Update the global best solution $\hat{x}(t)$ and its cost $f(\hat{x}(t))$;
  for *each edge* $(i,j) \in x^+(t)$ **do**
    Update global update rule: $\tau_{ij}(t+1) = (1-\rho_G)\tau_{ij}(t) + \rho_G \Delta\tau_{ij}(t)$;
  end
  $t \leftarrow t+1$;
end

Table 4: Pseudo Code of Ant Colony System Algorithm.

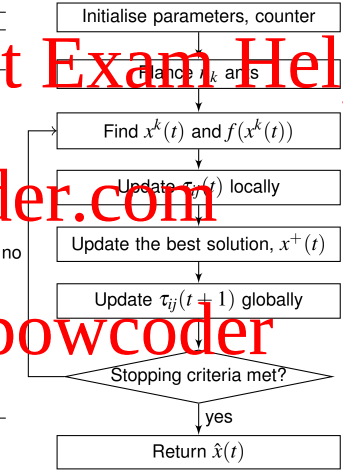


Figure 3: Flowchart of Ant Colony System Algorithm.