ANT COLONY OPTIMISATION

Office S2.14, Strand Building, Strand C

Email: hak-keung.lam@kcl.

https://nms.kcl.ac.uk

Nature-Inspired Learning Algorithms (7CCSMBIM)

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 limit
- To a prob

Swarm Intelligence

- “Swarm intelligence is the property of a system whereby the collective behaviours of (unsophisticated) agents interacting locally with their envi
- “Sw

expl

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.
- **Rob**
- **Dec**
- **Self**

  predefined.

KING'S College LONDON
University of London

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:**
  -
  - **Negative feedback:** to introduce a time scal through pheromone evaporation, to prevent pr (stagnation), for counter-balance and stabilis
  - **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).

Introduction

Ants

- Ants appeared on earth some 100 million years ago.
- Estimated total population: 10^{16} individuals.
- Soci
- Coll

  coo

  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.

Ant Optimisation Algorithm

- To search for an optimal path in a graph.
- Find the shortest path between their nest and food source, without any visib

Notation

- $x^k(t)$: 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: numb
- n_e: num

- (i,j): an e
- $\tau_{ij}(t)$: ph
- $\Delta\tau_{ij}^k(t)$: the change of pheromone concentration associated with edge (i,j)
- $\Delta\tau_{ij}^e(t)$: the change of pheromone concentration associated with edge (i,j)

- $f(x^k(t))$: 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.

- 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 $P\ (t+\ ) = \frac{(c+n_A(t))^{\alpha}}{\quad}$
    end
end

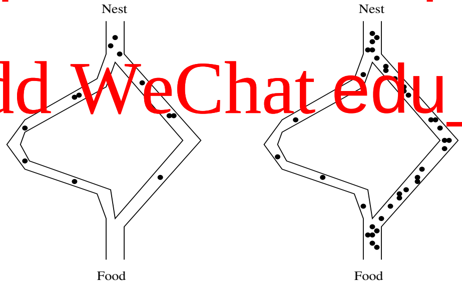
Table 1: Pseudo Code of Artificial Ant Decision P

The Binary Bridge Experiments

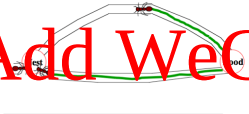
Shortest path selection by forager ants

- Shortest path is selected.
- Ants return to the nest earlier.
- The p

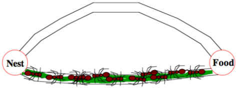
  feed



Shortest path selection by forager ants



(c) (d)



- **Stigmergy** is a class of mechanisms that mediate animal-to-animal interactions.
  - A form of indirect communication mediated by modifications of the environment.
  -

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

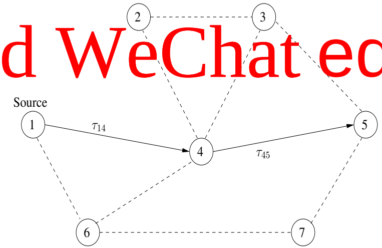
KING'S College LONDON
University of London

- **Artificial stigmergy:** "indirect communication mediated by numeric modifications of environmental states which are only locally accessible by the communicating agent". (Dorigo and Di Caro)
- *Artifi*
  enc
  indir

- (i,j): An edge from node i to node j.
- $\tau_{ij}(t)$: Pheromone concentration associated with edge (i,j) at generation/iteration t.

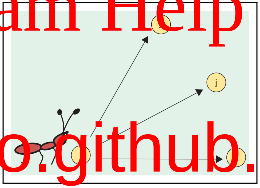
  -

- $L^k(t$
  con



- **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):

$$p_{ij}^k(t) = \frac{\tau_{ij}^{\alpha}(t)}{\tau_{iu}^{\alpha}(t)} \quad \text{if } j \in \mathcal{N}_i^k(t)$$



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 $\mathcal{N}_i^k(t) = 0$, the predecessor to node i i
  - 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

KING'S College LONDON
University of London

Example

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

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	____________	____________
4	____________	____________

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 $p^k_{1,j}(t)$	Accumul
2	$\frac{0.5}{0.5+0.3+0.2} = 0.5$	
3	$\frac{0.3}{0.5+0.3+0.2} = 0.3$	
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.

Evaporation of Pheromone Intensity (negative feedback)

- Pheromone intensity will evaporate.
  - To force ants to explore more.
  - To prevent premature convergence.
- For e

$$_{ij} \leftarrow \quad - \quad _{ij}$$

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

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 edg

where

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

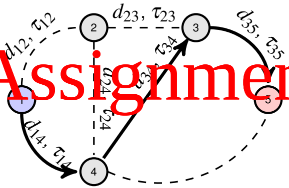
$x^k(t)$ is the solution of ant k,

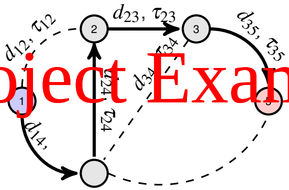
$f(x^k(t))$ is the quality of the solution,

$Q > 0$ is a constant,

n_k is the number of ants.

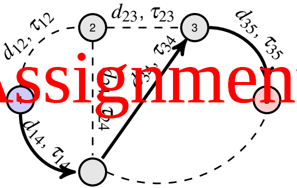
Example: Source node: 1; Target node: 5 ($d_{ij} = d_{ji}$, $\tau_{ij} = \tau_{ji}$ and n_k = 2)

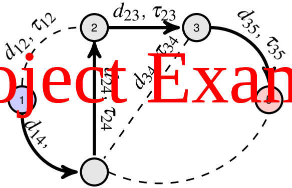




- Ant 1:
- $f(x^1(t)) = d_{14} + d_{34} + d_{35}$
- $f(x^2(t)) = d_{14} + d_{42} + d_{23} + d_{35}$

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





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

Pheromone update:

- Pheromone evaporation: $\tau_{ij}(t) = (1-\rho)\tau_{ij}(t)$ where ρ
- 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**
  - $x^k(t)=0$;
  - **While** *destination has not been reached* **do**
    - Select next node based on translation probability $p^k(t)$;
  - **en**[illegible]
  - Re[illegible]
  - Cal[illegible]
- **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**
    $x^k(t) = \emptyset$;
    While *destination has not been reached* **do**
      Select next node based on translation probability $p^k(t)$;
    en
    Re
    Cal
  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.

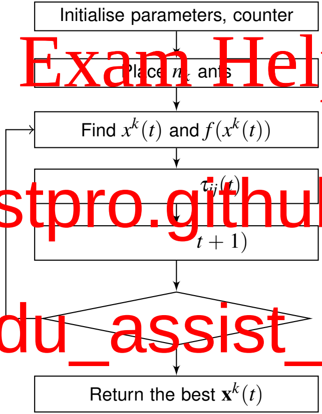


Figure 1: Flowchart of Simple Ant Colony Optimisation Algorithm.

Stopping Criteria:

- a maximum number of iterations has been exceeded.
- an acceptable solution has been found.
- all (or

Assignment Project Exam Help

https://eduassistpro.github.i

Add WeChat edu_assist_pr

Ant System was developed based on SACO.

Improvements:

- Incl k
- Incl
  -
  - May include all nodes not yet visited by ant
- **Different update strategies** for pheromon

- **Elitism** is implemented.

KING'S College LONDON
University of London

Transition Probability (two methods):

- **Method 1:**

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum \tau_{iu}^{\alpha}(t)\eta_{iu}^{\beta}(t)} & \text{if } j \in \mathcal{N}_i^k(t) \end{cases}$$

  - $\eta_{ij}(t)$: a priori effectiveness of the move from 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).

KING'S College LONDON
University of London

Transition Probability (two methods):

- **Method 2:**

$$p_{ij}^{k}(t \quad \frac{\alpha\tau_{ij}(t)+(1-\alpha)\eta_{ij}(t)}{\sum \alpha\tau_{iu}(t)+(1-\alpha)\eta_{iu}(t)} \text{ if } j \in \mathcal{N}_i^k(t)$$

$1, \cdots$

Pheromone evaporation:

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

Update of pheromone intensity/concentration:

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

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-**

  -

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

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

Elitist Strategy:

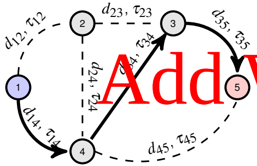
- $\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t) + n_e \Delta\tau_{ij}^e(t)$

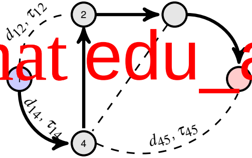
  - $\tilde{x}(t)$: the current best path (solution) at generation/iteration t.
  - $f(\tilde{x}(t)) = \min_{k=1,\ ,n_k} f(x^k(t))$.
  -
  -

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.
  - $f(\tilde{x}(t)) = \min_{k=1,\cdots,n_k} f(x^k(t))$.
  -





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

KING'S College LONDON
University of London

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 an* k *= 1* $\cdots$, n_k **do**
    $x^k(t) = \emptyset$;
    While *destination has not been reached* **do**
      Select next node based on translation probability $p_{ij}^k(t)$;
      A
    end
    Remo
    Calcu
  end
  for *each ed*
    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
      $\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t)$;
    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.

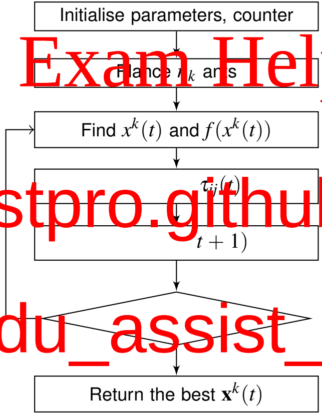


Figure 2: Flowchart of Ant System Algorithm.

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.

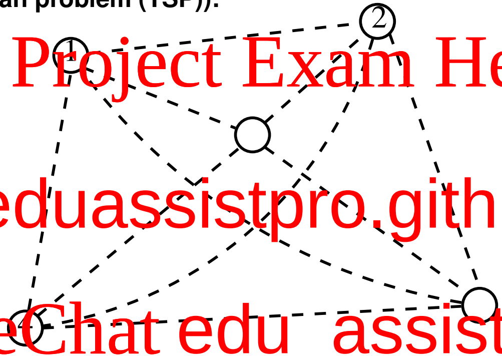
1. Place ants at different nodes.
2. Find the p
3. Update p
4. Go to step 2

(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				
3	3				
4	6				
5	18				



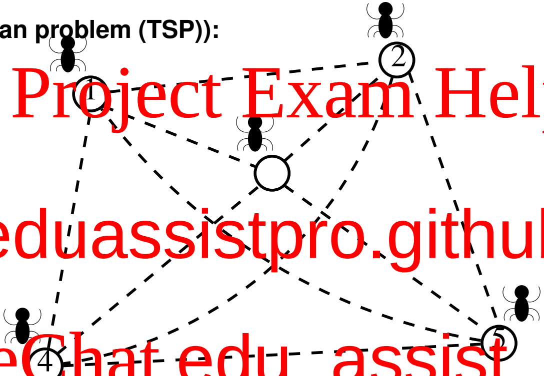
Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10				
3	3				
4	6				
5	18				

$\tau_{ij}(t)$ 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	–



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

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

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

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

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

KING'S College LONDON
University of London

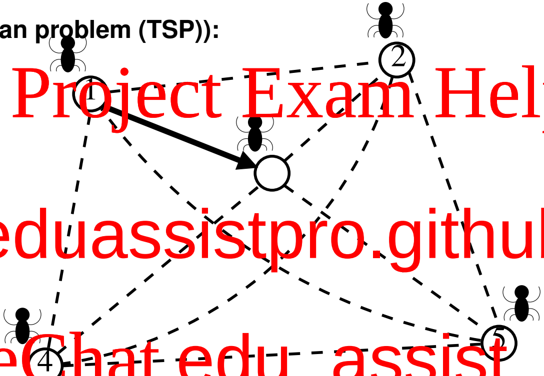
Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10				
3	3				
4	6				
5	18				

$\tau_{ij}(t)$ 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	–



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

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

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

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

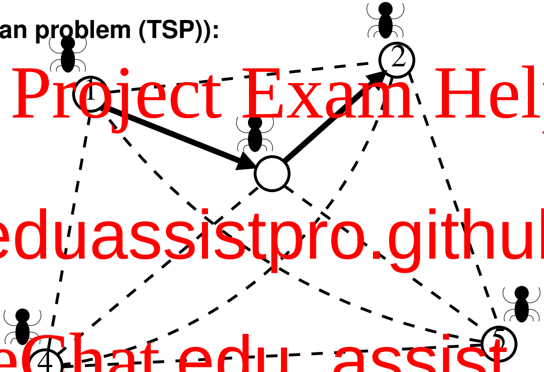
$x^5(t) = \{5,4\}$

King's College London
University of London

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				
3	3				
4	6				
5	18				



$\tau_{ij}(t)$ 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	–

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

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

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

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

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

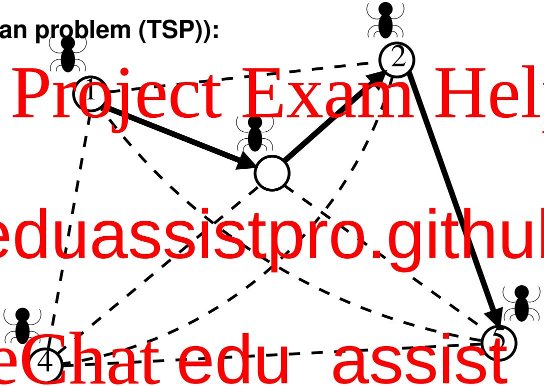
Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10				
3	3				
4	6				
5	18				

$\tau_{ij}(t)$ 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	–



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

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

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

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

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

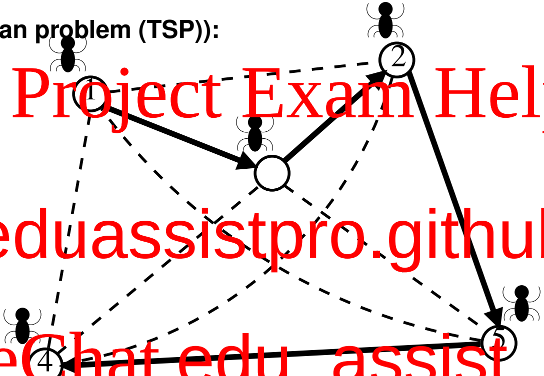
Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i,j)

$i \backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10				
3	3				
4	6				
5	18				

$\tau_{ij}(t)$ 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	–



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

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

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

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

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

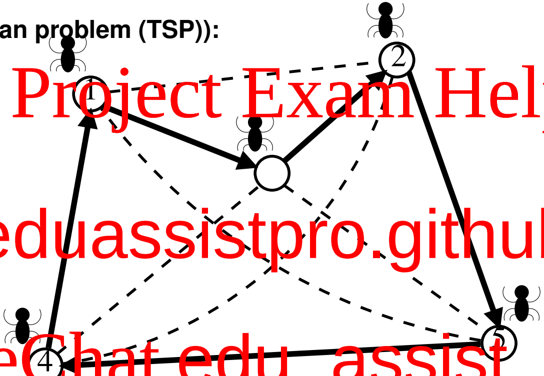
Example 1 (travelling salesman problem (TSP)):

d_{ij} for edge (i,j)

$i\backslash j$	1	2	3	4	5
1	–	10	3	6	18
2	10				
3	3				
4	6				
5	18				

$\tau_{ij}(t)$ 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	–



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

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

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

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

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

KING'S College LONDON
University of London

Example 1 (travelling salesman problem (TSP)):

- Evaporation of Pheromone Intensity (negative feedback)

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

- Updat

th $x^k(t)$

SACO: $_{ij}(t+1) = \ _{ij}(t) + \sum_{k=1} \ _{ij}(t), \qquad _{ij}(t) = \Big\{ \ 0$

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

- 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 was developed based on AS.

Improvements:

- Different transition rule, $p_{ij}^{k}(t)$.
  -
- Diff
  -
  - *Local update rule: pheromone evaporation.*

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

$$j = \begin{cases} \arg\max_{u \in \mathcal{N}_i^k(t)} \{\tau_{iu}(t)\eta_{iu}^{\beta}(t)\} \text{ if } r \leq r_0 \end{cases} \quad (1)$$

- $r \in [0,$
- $r_0 \in [0,1]$: 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 no
- $J \in \mathcal{N}_i^k(t)$ is a node randomly selected according to the probability:

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

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

Example:

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

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: $^{4}(t) = \ 2, 3, 7$
- For ar r_0

  - Assume $r_0 = 0.2$, $r = 0.1$.
  - Assume $\eta^{\beta}_{1,2}(t) = \eta^{\beta}_{1,3}(t) = \eta^{\beta}_{1,7}(t)$; $\tau_{1,2}(t) = 0$.

  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: ${}^{4}(t) = \; 2, 3, 7$
- For ar r_0

  - Assume $r_0 = 0.2$, $r = 0.1$.
  - Assume $\eta^{\beta}_{1,2}(t) = \eta^{\beta}_{1,3}(t) = \eta^{\beta}_{1,7}(t)$; $\tau_{1,2}(t) = 0$.

    which node is chosen?

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

Local and Global Update Rules:

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

King's College London
University of London

Local and Global Update Rules:

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

  - $\rho_L \in (0, 1)$ (0 and 1 are not inclusive): a user-sp
  - $\tau_0 > 0$: 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.

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

Local and Global Update Rules:

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

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

- 0 otherwise
- $\rho_G \in (0,1)$ (0 and 1 are not inclusive, why?): a u parameter.

- $x^+(t)$: the best solution(s) giving the shortest pat
  - **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).

Ant Colony System (ACS)

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**
      Sel
      Ad
    end
    Remove
    Calculat
  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.

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, initialise 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**
      Sel
      Ad
    end
    Remove
    Calculat
  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* $(i,j) \in \hat{x}^+(t)$ **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.

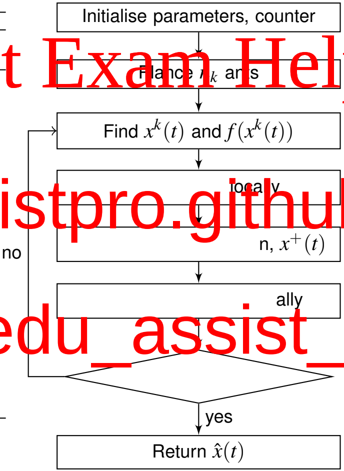


Figure 3: Flowchart of Ant Colony System Algorithm.