BITS Pilani
Pilani Campus

Artificial & Computational Intelligence

AIML CLZG557

M2 : Problem Solving Agent using Search

Dr. Sudheer Reddy

Course Plan

M1 Introduction to AI

M2 Problem Solving Agent using Search

M3 Game Playing

M4 Knowledge Representation using Logics

M5 Probabilistic Representation and Reasoning

M6 Reasoning over time

M7 Ethics in AI

Module 2 : Problem Solving Agent using Search

A. Uninformed Search

B. Informed Search

C. Heuristic Functions

D. Local Search Algorithms & Optimization Problems

Learning Objective

At the end of this class, students should be able to:

1. Convert a given problem into local search problem
2. Understand the significance of hyperparameters of the evolutionary algorithms
3. Apply appropriate local search and show the working of algorithm at least for first 2 iterations

Local Search & Optimization

Optimization Problem

Goal : Navigate through a state space for a given problem such that an optimal solution can be found

Objective : Minimize or Maximize the objective evaluation function value

Scope : Local

Objective Function : Fitness Value evaluates the goodness of current solution

Local Search : Search in the state-space in the neighbourhood of current position until an optimal solution is found

Single Instance Based	Multiple Instance Based
Hill Climbing	Genetic Algorithm
Simulated Annealing	Particle Swarm Optimization
Local Beam Search	Ant Colony Optimization
Tabu Search	

Particle Swarm Optimization

Swarm Intelligence

- Swarm Intelligence (SI) is artificial intelligence based on the collective behaviour of decentralized, self-organized systems.
- Characteristics of Swarms:
    - Composed of many individuals
    - Individuals are homogeneous
    - Local interaction based on simple rules (collision avoidance, velocity matching..)
    - Self-organization

Particle Swarm Optimization (PSO) (Contd..)

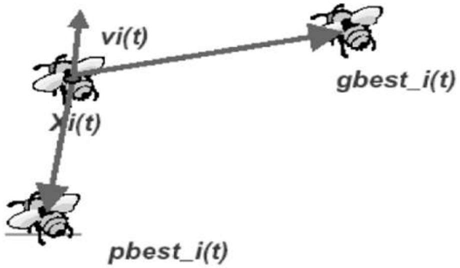
- Collection of flying particles (swarm) – Changing solutions
- Search area – Possible solutions
- Movement towards a promising area to get the global optimum
- Each particle adjusts its travelling speed dynamically corresponding to the flying experiences of itself and its colleagues
- Each particle keeps track:
  - Its best solution, personal best - pbest
  - The best value of any particle, global best – gbest.
- Each particle modifies its position according to:
  - Its current position
  - Its current velocity
  - The distance between its current position and pbest.
  - The distance between its current position and gbest

Particle Swarm Optimization (Contd..)

Each individual is called a particle

Particle = Position vector $x_i(t)$+ Velocity vector $v_i(t)$

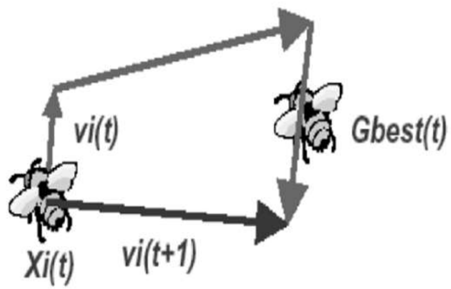
Heuristics Considered = function (Individual's Best So-far , Group's Best So-far)



Updating velocity vector:

$$X_i(t+1) = X_i(t) + V_i(t+1)$$

$$v_i(t+1) = \alpha\ v_i + c_1 \times rand \times (pbest(t)\text{-}x_i(t)) + c_2 \times rand \times (gbest(t)\text{-}x_i(t))$$

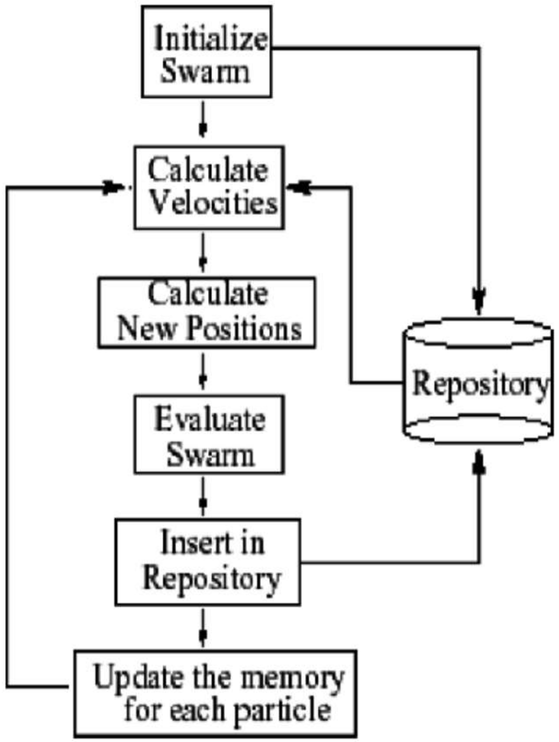


α is inertia weight and controls exploration and exploitation
c_1 and c_2 the cognition and social components respectively
rand is a random number generator

Particle Swarm Optimization (Contd..)

Basic Flow of PSO

1. Initialize the swarm with random initializations
2. Evaluate fitness value for each of these individuals
3. Modify **g_{best}, p_{best},** and velocity
4. Move each particle to new particle
5. Go to step 2, and repeat until convergence



Particles velocities on each dimension are clamped to a max velocity v_{max}

PSO vs GA

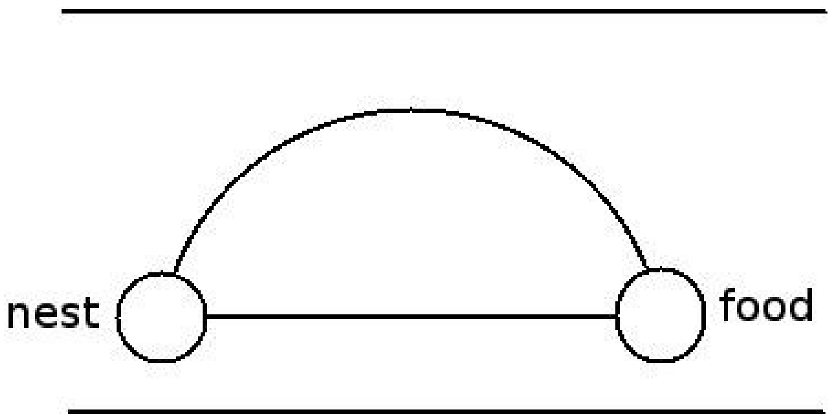
- Common points
  - Both algorithms start with a group of randomly generated population
  - Both have fitness value to evaluate the population
  - Both update the population and search for optimum with random sampling
  - Both do not guarantee success

- Difference
  - PSO does not have genetic operators like crossover and mutation
  - Particles update themselves
  - They also have memory which is critical to the algorithm
  - In GA, the information is shared between individuals, in PSO, the information is shared one-way.

Ant Colony Optimization

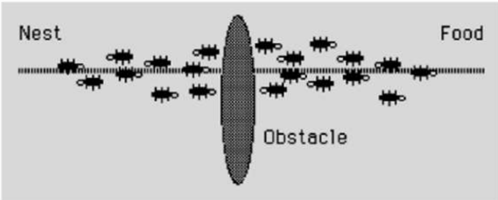
Ant Colony Optimization

- The way ants find their food in shortest path is interesting
- Ants secret pheromones to remember their path
- These pheromones evaporate with time

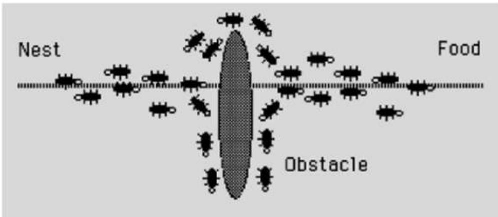


Ant Colony Optimization

As ants have weak vision and weak memory, they communicate using the stimulus

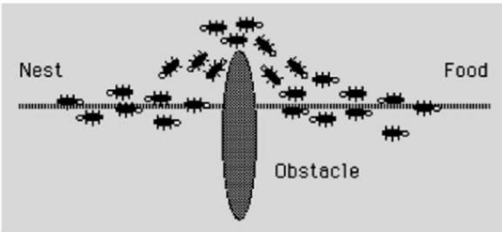


They leave chemical deposits called **Pheromone** to inform other ants about food, colonies, etc.



Using olfactory **stimulus** the insects follow the **trail** left behind by other ants

Other ants follow the pheromone trail and **deposit even more pheromone** leading to a positive feedback effect, **if the selected path is good.**



Ants **prefer** to follow a path with rich in pheromone

Ant Colony Optimization (Contd..)

Ant Colony Optimization

Problem Definition

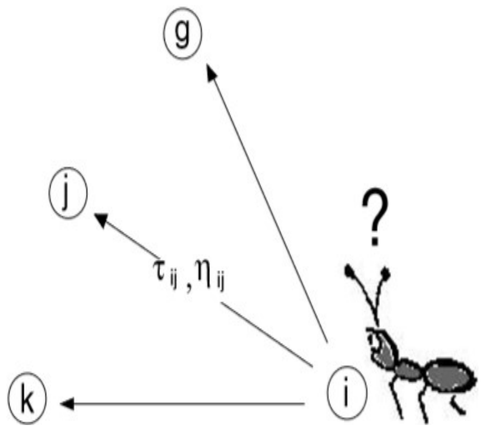
Create Construction Graph

Define & Initialize Pheromone Trails

Define Heuristics

Construct Solution

Update Pheromones



m - No.of.Ants   **n** - No.of.Nodes

Sk - Memory   **α** - Influence of τ

β - Influence of n

ρ – Pheromone evaporation rate

Travelling Salesman Problem
Cities = {i, j, k , g……} $\Delta\tau_{ij}{}^{k}$ is the pheromone laid on edge (i, j) by ant k

Ants move through states of problem by applying a stochastic local decision policy based on two parameters, **trails and attractiveness**

When an ant completes a solution, or during construction phase, the solution is evaluated and trail values on components are modified

Ant Colony Optimization

Problem Definition

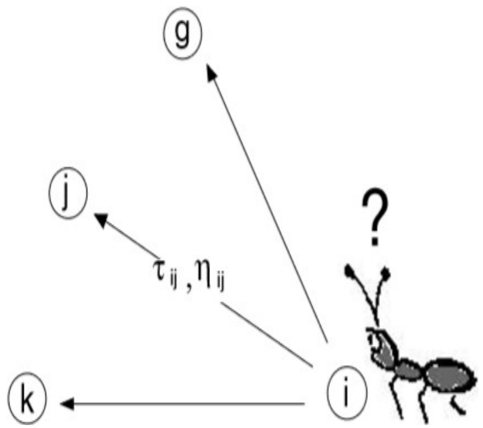
Create Construction Graph

Define & Initialize Pheromone Trails

Define Heuristics

Construct Solution

Update Pheromones



m - No.of.Ants   **n** - No.of.Nodes

S$_k$ - Memory   **α** - Influence of τ

β - Influence of η

ρ – Pheromone evaporation rate

Travelling Salesman Problem
While real ants deposit chemical substance on the visited state, artificial ants change numerical information of problem state, stored locally when that state is visited

Ant Colony Optimization

Problem Definition

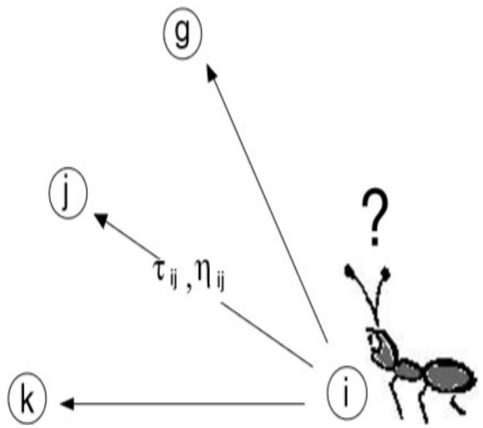
Create Construction Graph

Define & Initialize Pheromone Trails

Define Heuristics

Construct Solution

Update Pheromones



m - No.of.Ants

n - No.of.Nodes

Sk - Memory

α - Influence of τ

β - Influence of n

ρ – Pheromone evaporation rate

Travelling Salesman Problem

$$\tau_0 = \frac{m}{C} \qquad \tau_0 = \frac{1}{\rho C}$$

**ρ - ** In ACO and real ants it is important to slowly forget the history to avoid stacking in local extremes and move toward new regions using the evaporation rate

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

where Q is a constant

L_k is the length of tour constructed by ant k

Ant Colony Optimization

Problem Definition

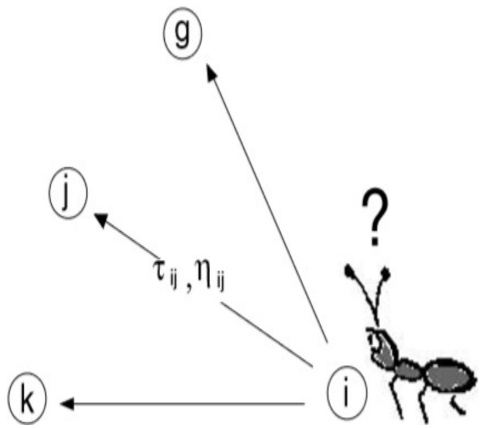
Create Construction Graph

Define & Initialize Pheromone Trails

Define Heuristics

Construct Solution

Update Pheromones



m - No.of.Ants **n** - No.of.Nodes

S_k - Memory **α** - Influence of τ

β - Influence of n

ρ – Pheromone evaporation rate

Travelling Salesman Problem
$n_{i,j}(S_k) = \frac{1}{d_{i,j}}$ d_{ij} is the distance between cities i and j $n_{i,j}$: To optimize for shortest paths, every state transition from i to j, is guided by the attractiveness as the heuristic

Ant Colony Optimization

Problem Definition

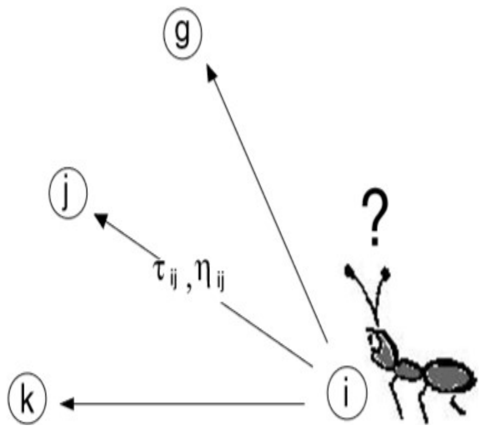
Create Construction Graph

Define & Initialize Pheromone Trails

Define Heuristics

Construct Solution – Action Choice Rule

Update Pheromones



m - No.of.Ants **n** - No.of.Nodes

Sk - Memory **α** - Influence of τ

β - Influence of n

ρ – Pheromone evaporation rate

Travelling Salesman Problem

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\,\eta_{ij}^{\beta}}{\sum_{c_{ij} \in N(s^{p})} \tau_{ij}^{\alpha}\,\eta_{ij}^{\beta}} & if\, c_{ij} \in N(s^{p}) \\ 0 & otherwise \end{cases}$$

Transition Policy: Decision making is based on probabilistic inference. It depends on local state information and existing pheromone trails

edge(i,j) where j is not yet visited $j \in \mathcal{N}_i^k$

Parameters α, β control relative importance of pheromone vs heuristic

Ant Colony Optimization

Problem Definition

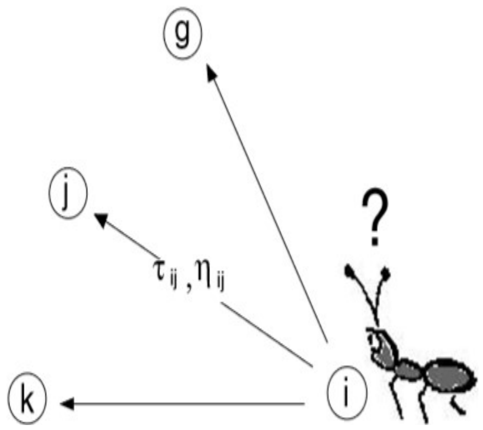
Create Construction Graph

Define & Initialize Pheromone Trails

Define Heuristics

Construct Solution

Update Pheromones: Evaporation



m - No.of.Ants   **n** - No.of.Nodes

S_k - Memory   **α** - Influence of τ

β - Influence of n

ρ – Pheromone evaporation rate

Travelling Salesman Problem

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij}, \quad \forall (i,j) \in L$$

Local Update

$$\tau_{ij} = (1 - \varphi) * \tau_{ij} + \varphi * \tau_0$$

A Local pheromone update is performed by all ants only to the last edge it traversed after each construction step

$\varphi \in (0,1)$ is pheromone decay coefficient

Global Update

$$\tau_{ij} \leftarrow \begin{cases} (1-\rho) * \tau_{ij} + \rho * \Delta\tau_{ij} & \textit{if } (i,j) \textit{ belongs to best tour} \\ \tau_{ij} & \textit{otherwise} \end{cases}$$

$$\Delta\tau_{ij} = 1/L_{best}$$

Where L_{best} - length of tour of best ant either in current iteration or best so far

Applied at the end of iteration by only one ant (either iteration-best/best-so-far)

Ant Colony Optimization

Problem Definition

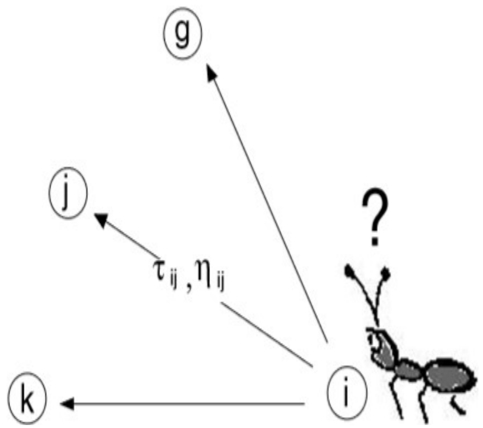
Create Construction Graph

Define & Initialize Pheromone Trails

Define Heuristics

Construct Solution

Update Pheromones: Deposit



m - No.of.Ants **n** - No.of.Nodes

S$_k$ - Memory **α** - Influence of τ

β - Influence of n

ρ – Pheromone evaporation rate

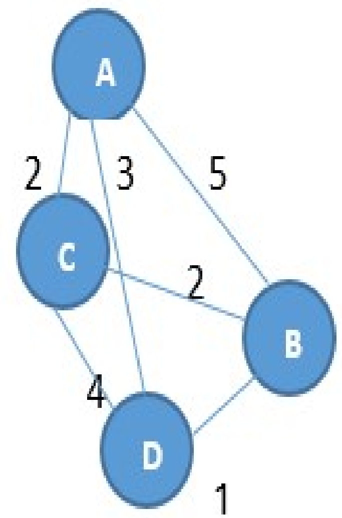
Travelling Salesman Problem
$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^{best},$$ where $\Delta\tau_{ij}^{best} = 1/C^{best}.$

Deposited amount of Pheromone: Depends on quality of discovered food source / cost of solution found so far

Ant Colony Optimization – Sample Problem

A	B	C	D	
0	1	1	1	**A**
	0	1	1	**B**
		0	1	**C**
			0	**D**

A	B	C	D	
0	5	2	3	**A**
	0	2	1	**B**
		0	4	**C**
			0	**D**



Packet routing:

Optimal path finder for communication over network.

No.of.Ants : **m**= 5

Pheromone evaporation rate: **ρ** =0.5

α - Influence of τ = 1

β - Influence of n = 1

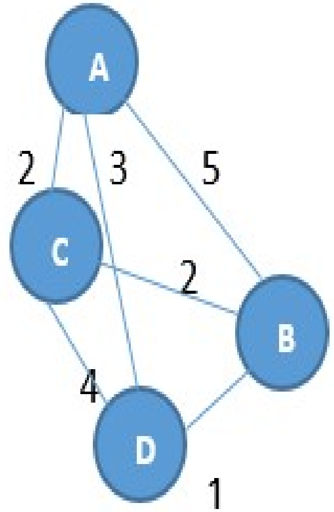
$\tau_{i,j}$ - Pheromone level = 1

1. Represent the solution space by a construction graph
2. Initialize ACO parameters
3. Generate random solutions from each ant's random walk
4. Update pheromone information
5. Go to 3 and repeat until convergence or a stopping criteria is met

Ant Colony Optimization – Sample Problem

A	B	C	D	
0	1	1	1	A
	0	1	1	B
		0	1	C
			0	D

A	B	C	D	
0	5	2	3	A
	0	2	1	B
		0	4	C
			0	D



<u>Iteration 1: 5 packets randomly sent</u>

1: ABDCA → C_k = AB+BD+DC+CA = 12

2: ADBCA → C_k = 8

3: ABDCA → C_k = 12

4: ADBCA → C_k = 8

5: ACBDA → C_k = 8

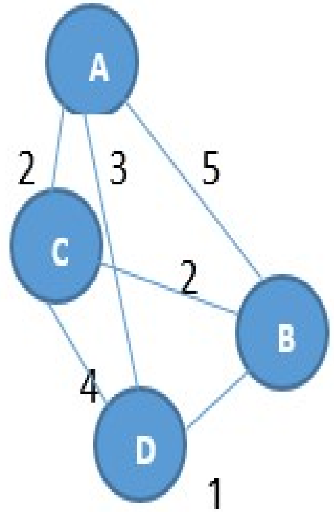
This is a random trail .
Typically the algorithm uses the probability value based Transition Policy for optimal search to construct the paths

1. Represent the solution space by a construction graph
2. Initialize ACO parameters
3. Generate random solutions from each ant's random walk
4. Update pheromone information
5. Go to 3 and repeat until convergence or a stopping criteria is met

Ant Colony Optimization - Sample Problem

A	B	C	D	
0	~~1~~ **0.58**	1	1	**A**
	0	1	1	**B**
		0	1	**C**
			0	**D**

A	B	C	D	
0	5	2	3	**A**
	0	2	1	**B**
		0	4	**C**
			0	**D**



<u>Iteration 1: 5 packets</u>

1: **AB**DCA → C_k =12

2: ADBCA → C_k = 8

3: **AB**DCA → C_k = 12

4: ADBCA → C_k = 8

5: ACBDA → C_k = 8

$$\Delta \tau_{ij} = \frac{1}{Ck}$$

<u>Pheromone Update</u>

AB:

2 ants with C_k=12

$ant\ 1: \Delta \tau_{ij} = \frac{1}{12}$

$ant\ 3: \Delta \tau_{ij} = \frac{1}{12}$

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

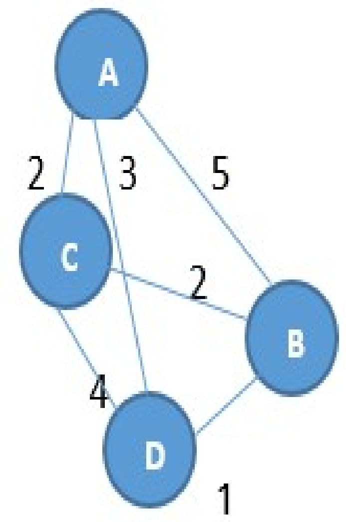
$$\tau_{ij} \leftarrow (1 - 0.5) * 1 + \left(\frac{1}{12} + \frac{1}{12}\right) * 0.5$$

$\boldsymbol{\tau_{ij}} \leftarrow$ **0.58**

Ant Colony Optimization – Sample Problem

A	B	C	D	
0	0.58	1.04	0.875	**A**
	0	0.875	1.04	**B**
		0	0.75	**C**
			0	**D**

A	B	C	D	
0	5	2	3	**A**
	0	2	1	**B**
		0	4	**C**
			0	**D**



Iteration 1: (Above table is a random values filled for rest. Students should try to work out for the right values.)

1: ABDCA → C_k =12

2: ADBCA → C_k = 8

3: ABDCA → C_k = 12

4: ADBCA → C_k = 8

5: ACBDA → C_k = 8

$$\Delta \tau_{ij} = \frac{1}{Ck}$$

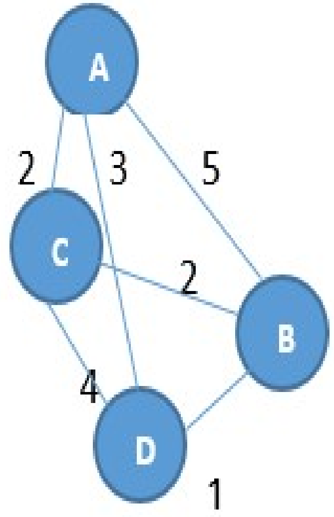
$$\tau_{ij} \leftarrow (1-\rho) * \tau_{ij} + \rho \sum_{k=1}^{m} \Delta \tau_{ij}^{\ k}$$

1. Represent the solution space by a construction graph
2. Initialize ACO parameters
3. Generate random solutions from each ant's random walk
4. Update pheromone information
5. Go to 3 and repeat until convergence or a stopping criteria is met

Ant Colony Optimization – Sample Problem

A	B	C	D	
0	**0.58**	**1.04**	**0.875**	**A**
	0	**0.875**	**1.04**	**B**
		0	**0.75**	**C**
			0	**D**

A	B	C	D	
0	5	2	3	**A**
	0	2	1	**B**
		0	4	**C**
			0	**D**



<u>Iteration 2: 3 packets randomly sent</u>

1: BCADB→ $C_k = 8$

2: BDACB → $C_k = 8$

3: DBACD → $C_k = 12$

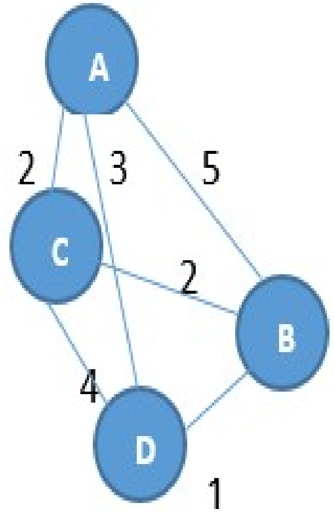
$$\tau_{ij} \leftarrow (1-\rho) * \tau_{ij} + \rho \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k}$$

$$\Delta\tau_{ij} = \frac{1}{Ck}$$

Ant Colony Optimization – Sample Problem

A	B	C	D	
0	0.58	~~1.04~~ 0.687	0.875	A
	0	0.875	1.04	B
		0	0.75	C
			0	D

A	B	C	D	
0	5	2	3	A
	0	2	1	B
		0	4	C
			0	D



Iteration 2: 3 packets

1: BCADB→ C_k =8

2: BDACB → C_k = 8

3: DBACD → C_k = 12

$$\triangle \tau_{ij} = \frac{1}{Ck}$$

Pheromone Update

AC:

3 ants with C_k =12 , 8, 8

$ant\ 1: \triangle \tau_{ij} = \frac{1}{8}$

$ant\ 2: \triangle \tau_{ij} = \frac{1}{8}$

$ant\ 3: \triangle \tau_{ij} = \frac{1}{12}$

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

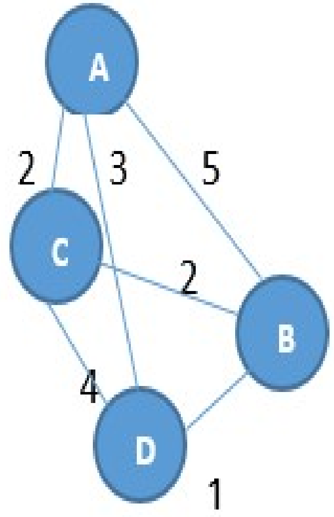
$$\tau_{ij} \leftarrow (1 - 0.5) * 1.04 + 0.5 \left(\frac{1}{8} + \frac{1}{8} + \frac{1}{12}\right)$$

$\boldsymbol{\tau_{ij}} \leftarrow$ **0.687**

Ant Colony Optimization – Sample Problem

A	B	C	D	
0	**0.413**	**0.853**	**0.687**	**A**
	0	**0.687**	**0.853**	**B**
		0	**0.458**	**C**
			0	**D**

A	B	C	D	
0	5	2	3	**A**
	0	2	1	**B**
		0	4	**C**
			0	**D**



<u>Iteration 2: 3 packets</u>

1: BCADB→ C_k =8

2: BDACB → C_k = 8

3: DBACD → C_k = 12

<u>(Above table is a random values filled for rest. Students should try to work out for the right values.)</u>

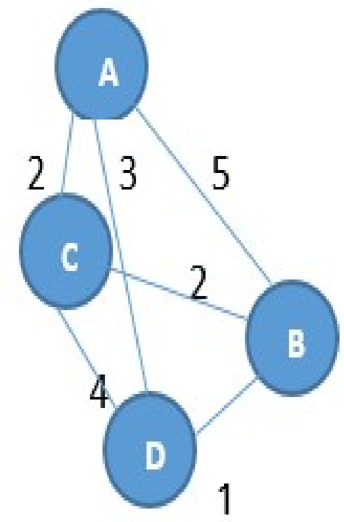
$$\tau_{ij} \leftarrow (1-\rho) * \tau_{ij} + \rho \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k}$$

$$\Delta \tau_{ij} = \frac{1}{Ck}$$

Ant Colony Optimization – Sample Problem

A	B	C	D	
0	**0.413**	**0.853**	**0.687**	**A**
	0	**0.687**	**0.853**	**B**
		0	**0.458**	**C**
			0	**D**

A	B	C	D	
0	5	2	3	**A**
	0	2	1	**B**
		0	4	**C**
			0	**D**



<u>Solution Choice:</u>

A new packet arrives at router C . Which is the next hop ?

$$CA := \frac{\tau_{ij}^{\alpha}\,\eta_{ij}^{\beta}}{\sum_{c_{ij}\in N(s^p)} \tau_{ij}^{\alpha}\,\eta_{ij}^{\beta}} = \frac{0.853 * (\frac{1}{2})}{(0.853*\frac{1}{2}+0.687*\frac{1}{2}+0.458*\frac{1}{4})} \sim 60-70\%$$

Similarly calculate for below edges as well before deciding the next node to send packets to.

CB:=

CD:=

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l\in N_i^k}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}}, \quad \text{if } j \in N_i^k$$

Required Reading: Some Reading materials are uploaded in the canvas pages

Research papers & web resources

Thank You for all your Attention

Note : Some of the slides are adopted from AIMA TB materials