# Ant Algorithms: Part 1

# Ant Algorithms

Ant's are social insects, living in colonies of 30 to millions of individuals.

- Current total ant population is estimated at $10^{16}$ individuals
- The complex behaviours that emerge from colonies of ants have intrigued humans, and there have been many studies of ant colonies aimed at a better understanding of these collective behaviours
- Some interesting and well studied behaviours are:
  - foraging behaviour
  - division of labour
  - cemetery organization and brood care
  - construction of nests

# Ant Algorithms

Most (though not all) ants communicate using some form of indirect communication, which was termed as **stigmergy** by the biologist Grassé.

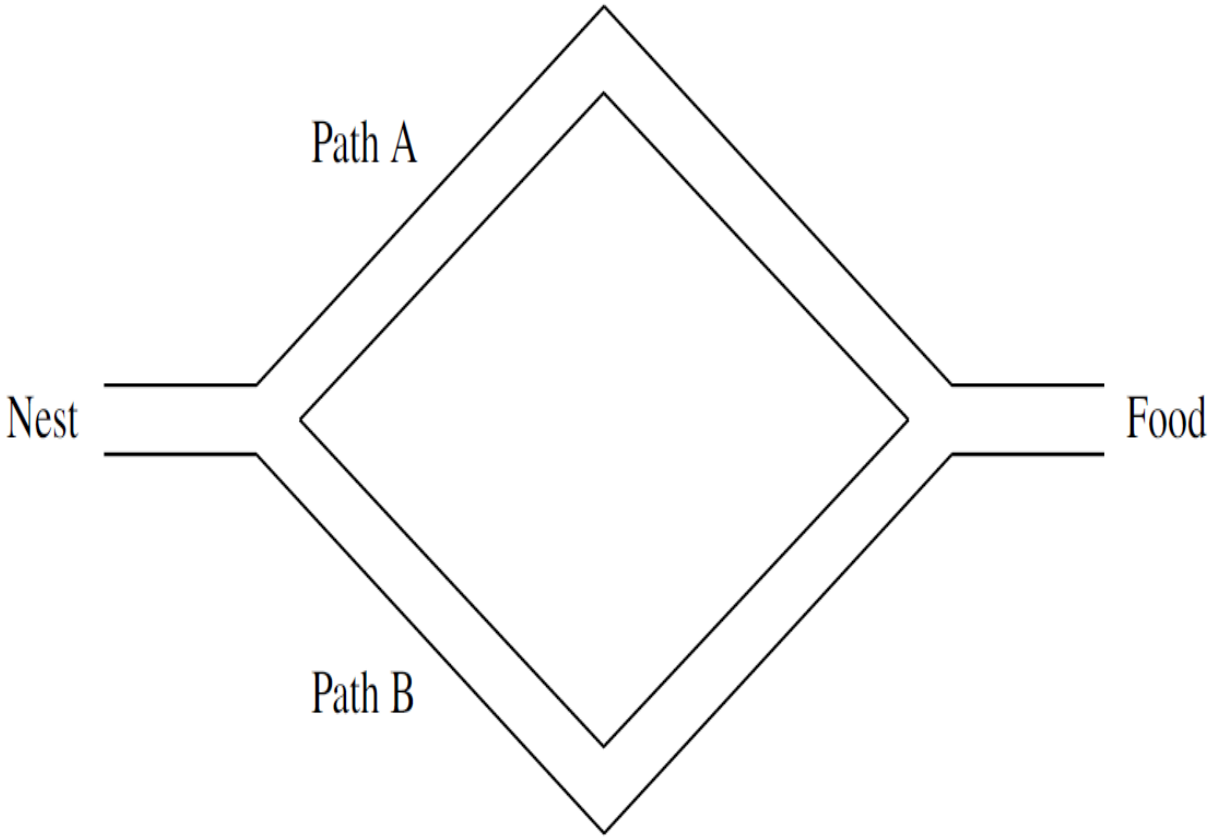
- The indirect communication is often achieved via an interaction with the environment
- The most classic example of **stigmergy** is that of **pheromone** trails.
  - The first algorithmic models of foraging behaviour used pheromones
  - There are two clear advantages of this type of communication
    - ★ One insect need not know the location of the others with whom it is communicating.
    - ★ The communication will potentially outlast the sender. (say the ant died)

# Ant Algorithms

- In the context of collective behaviour, social insects are basically **stimulus-response agents**.
  - Based on information perceived from the local environment, an individual performs a simple, basic action. Often with a random subcomponent.
  - Despite this simplicity in individual behavior, social insects collectively form a highly structured social organism.
  - Emergent behavior can arise form a very simple set of rules.

# Ant Algorithms

The origins: Pasteels tried to mathematically model the behavior of the ant species:*Iridomyrmex humilis* in the famous binary bridge experiment



# Ant Algorithms

With the knowledge that the ants drop a pheromone trails, Pasteels wanted to model which path the ants would eventually favor. Empirically it was derived that the guiding equation was simply:

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

where

- $n_A(t)$ and $n_B(t)$ ants on paths $A$ and $B$ respectively at time step $t$.
- $c$ quantifies the degree of attraction of an unexplored branch
- $\alpha$ is the bias to using pheromone deposits in the decision process.

# Ant Algorithms

For example: $n_A(t) = 5, n_B(t) = 100$, $c = 20$, and $\alpha = 2$ then

$$P_A(t+1) = 0.04 \tag{2}$$

which is to be expected as path $A$ has very few ants on it.

# Ant Algorithms

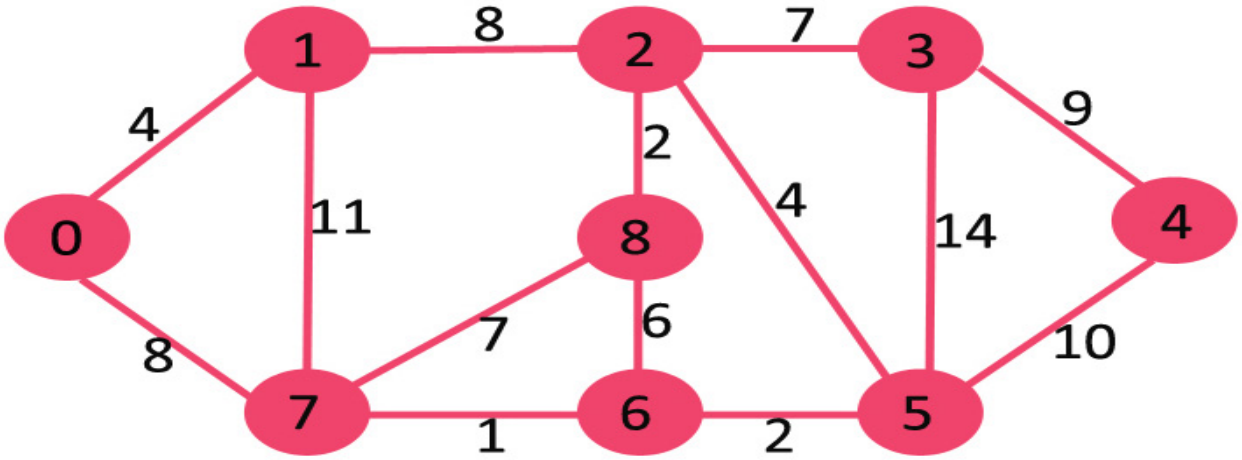
What is more interesting in-terms of the application of AA to optimization is the extended experiment by Goss *et al* where one of the two bridges is longer than the other.

- It was found that given enough time the ants favored the shorted path, with a substantially larger percentage of ants choosing to follow the shortest path.
- Selection is biased towards the shortest path, since ants that follow the shortest path return to the nest earlier than ants on the longer path.
  - The pheromone on the shorter path is therefore reinforced sooner than that on the longer path.
- Goss et al. found that the probability of selecting the shorter path increases with the **length ratio** between the two paths. Refereed to as the differential path length effect

# Ant Algorithms

We will now go over a simplified version of the original ant colony optimization-meta-heuristic (ACO-MH), called simple ACO (SACO). The intention is to get you used to the ACO concepts in a simple context.

- SACO and ACO-MH are predominantly used for solving types of combinatorial problems.
- Let us consider the problem of finding the shorted path between two nodes in graph.



# Ant Algorithms

The following are the main components of SACO.

- Begin by assigning a small amount of pheromone to all links, $\tau_{i,j}$.
- $n_k$ ants are placed at the source node.
- Each ant must then attempt to construct a path from the source to the destination but using the following equation to guide it's transition probability

$$P_{i,j} = \begin{cases} \frac{\tau_{i,j}^{\alpha}(t)}{\sum_{e \in \mathcal{N}_i^k} \tau_{i,e}^{\alpha}} & \text{if } j \in \mathcal{N}_i^k \\ 0 & \text{if } j \notin \mathcal{N}_i^k \end{cases} \tag{3}$$

where $\mathcal{N}_i^k$ is set of feasible nodes connected to node $i$, with respect to ant $k$. $\alpha$ is a positive constant used to amplify the influence of pheromone concentrations.

# Ant Algorithms

- If for any node $i$ and ant $k$, if $\mathcal{N}_i^k = \emptyset$, then the predecessor to node $i$ is included in $\mathcal{N}_i^k$
  - Note that this may cause loops to occur within constructed paths.
  - These loops are removed once the destination node has been reached.
- Once all ants have constructed a complete path from the origin node to the destination node, and all loops have been removed, each ant retraces its path to the source node deterministically, and deposits a pheromone amount,
  - The amount dropped should be inversely proposal to the lengths of the found path.

$$\Delta\tau_{i,j}^k(t) \propto \frac{1}{L^k(t)} \tag{4}$$

  - Shorter paths $\implies$ more pheromone dropped.

# Ant Algorithms

- This implies that after all ants have dropped their pheromone trails we have

$$\tau_{i,j}(t+1) = \tau_{i,j}(t) + \Delta\tau_{i,j}(t) \tag{5}$$

where

$$\Delta\tau_{i,j}(t) = \sum_{k=1}^{n_k} \Delta\tau_{i,j}^{k}(t) \tag{6}$$

- If the pheromone trail lasted forever it is likely that paths will become over saturated, and force the ants to over exploit found solutions.
  - To avoid this an evaporation rate is added $\rho$.
  - **Before** the ants lay new pheromone trails the current pheromone level is updated with

$$\tau_{i,j}(t) = (1-\rho)\tau_{i,j}(t) \tag{7}$$

with $\rho \in [0,1]$

# Ant System

The first ant algorithm was developed by Marco Dorigo, who has over 87000 citations, and is refereed to as the **ant system**.

# Ant System

The SACO algorithm can be greatly improved with just a few changes. The most important change proposed in ant system (AS) are

- The inclusion of heuristic information into the transition probability.
- specifically

$$P_{i,j} = \begin{cases} \frac{\tau_{i,j}^{\alpha}(t)\eta_{i,j}^{\beta}(t)}{\sum_{e \in \mathcal{N}_i^k} \tau_{i,e}^{\alpha}\eta_{i,e}^{\beta}(t)} & \text{if } j \in \mathcal{N}_i^k \\ 0 & \text{if } j \notin \mathcal{N}_i^k \end{cases} \tag{8}$$

  where $\eta_{i,j}^{\beta}(t)$ indicates the desirability of the moves, which is computed based on heuristic information.
- The transition probability used by AS is a balance between pheromone intensity and heuristic information. The balance is controlled by $\alpha$ and $\beta$

# Ant System

The other change is the inclusion of a memory.

- Specifically, the inclusion of a tabu list was proposed by Dorigo.
- A list of all visited nodes are stored in a tabu list. Specifically the sets $\mathcal{N}_i^k$ will exclude and node on the tabu list.

# Ant System

There where originally three ways of calculating $\Delta\tau_{i,j}^k(t)$

- Ant-cycle AS:

$$\Delta\tau_{i,j}^k(t) = \begin{cases} \frac{Q}{f(x^k(t))} & \text{if link } (i,j) \text{ occurs in path } x^k(t) \\ 0 & \textit{otherwise} \end{cases} \tag{9}$$

  $Q$ is a positive constant, and $f(x^k(t))$ is the quality of the complete path constructed by the ant. (if we are minimizing)

- Ant-density AS:

$$\Delta\tau_{i,j}^k(t) = \begin{cases} Q & \text{if link } (i,j) \text{ occurs in path } x^k(t) \\ 0 & \textit{otherwise} \end{cases} \tag{10}$$

- Ant-quantity AS:

$$\Delta\tau_{i,j}^k(t) = \begin{cases} \frac{Q}{d_{i,j}} & \text{if link } (i,j) \text{ occurs in path } x^k(t) \\ 0 & \textit{otherwise} \end{cases} \tag{11}$$

  $d_{i,j}$ is the length of the link.

# Ant System

Other addition to the AS is to add the concept of elitism

- Specifically, the pheromone update equation changes to

$$\tau_{i,j}(t+1) = \tau_{i,j}(t) + \Delta\tau_{i,j}(t) + n_e\Delta\tau'_{i,j}(t) \tag{12}$$

where

$$\Delta\tau'_{i,j}(t) = \begin{cases} \frac{Q}{f(\hat{x}(t))} & \text{if } (i,j) \in \hat{x}(t) \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

an $n_e$ is positive constant controls the strength of the elitist force.

- The elitist strategy has as its objective directing the search of all ants to construct a solution to contain links of the current best route.

# Ant Colony System

The ant colony system (ACS) was developed by Gambardella and Dorigo to improve the performance of AS. ACS differs from AS in four aspects

- a different transition rule is used
- a different pheromone update rule is defined
- local pheromone updates are introduced
- candidate lists are used to favor specific nodes (added in scalability)

# Ant Colony System

The ACS transition rule, also referred to as a pseudo-random-proportional action rule is:

- Ant k, currently located at node i, selects the next node j to move to using the rule,

$$j = \begin{cases} argmax_{u \in \mathcal{N}_i^k(t)} \tau_{i,u}(t) \eta_{i,u}^{\beta}(t) & \text{if } r \leq r_0 \\ J & \text{if } r > r_0 \end{cases} \tag{14}$$

  where $r \sim U(0,1)$, and $r_0 \in [0,1]$ is user-specified parameter
- $J \in \mathcal{N}_i^k(t)$ is a node randomly selected using according to probability

$$p_{i,J}^k = \frac{\tau_{i,J}^{\alpha}(t) \eta_{i,J}^{\beta}(t)}{\sum_{u \in \mathcal{N}_i^k} \tau_{i,u}^{\alpha} \eta_{i,u}^{\beta}(t)} \text{ if } j \in \mathcal{N}_i^k \tag{15}$$

  $\mathcal{N}_i^k$ is a set of valid nodes to visit from $i$.

# Ant Colony System

The pheromones updates rule changes such that

- Only the "best" ant lays down new pheromones.
- let $\hat{x}(t)$ represent the best path, then the global pheromone update rule becomes

$$\tau_{i,j}(t+1) = (1-\rho_1)\tau_{i,j}(t) + \rho_1 \Delta\tau_{i,j}(t) \tag{16}$$

where

$$\Delta\tau_{i,j}(t) = \begin{cases} \frac{Q}{f(\hat{x}(t))} & \text{if link } (i,j) \text{ occurs in path } \hat{x}(t) \\ 0 & \textit{otherwise} \end{cases} \tag{17}$$

- The ACS global update rule causes the search to be more directed, by encouraging ants to search in the vicinity of the best solution found thus far.

# Ant Colony System

Gambardella and Dorigo implemented two methods of selecting the "best" path:

- **iteration-best**, where $\hat{x}(t)$ represents the best path found during the last completed iteration, $t$
- **global-best**, where $\hat{x}(t)$ represents the best path found from the first iteration of the algorithm.

An additional local update rule is used when an ant traverses a link, specifically

$$\tau_{i,j}(t) = (1 - \rho_2)\tau_{i,j}(t) + \rho_2\tau_0. \tag{18}$$

where both $\rho_2$ and $\tau_0$ are positive control parameters. It was found that $\tau_0 = \frac{1}{n_g L}$ was effectve for TSP problems. ($n_g$=number of nodes, $L$ is the shortest tour produced by the nearest neighbor heuristic). Both are however problem dependent

# Ant Colony System

The last change present in ACS is the use of a candidate list.

- This was a TSP specific addition.
  - when trying to solve big TSP problems it is common practice to use a data structure known as a candidate list.
  - A candidate list is a list of preferred cities to be visited;
    - ⋆ it is a static data structure which contains, for a given city , the closest cities, ordered by increasing distances;
  - IN ACS's introduction a size of 15 was used.

This was a way of inserting heuristic (bias) to speed up the process that was specific to TSP. Full detail can be found at
`http://people.idsia.ch/~luca/acs-ec97.pdf`

# Max-Min Ant System

A variant to AS was proposed by Stützle and Hoos. The primary changes in max-min ant system (MMAS) from AS are

- Pheromone intensities are restricted within given intervals.
- Only the best ant may reinforce pheromones (a mixed approach was used in subsequent versions)
- Initial pheromones are set to the max allowed value
- a pheromone smoothing mechanism is used.

# Max-Min Ant System: Pheromone Interval

- Set a minimal amount of pheromone $\tau_{min}$ allowed on all links.
  - This ensures that all links have a non-zero chance of being selected.
  - Forces some minimum amount of exploration.
- Set a maximum allowable amount of pheromone $\tau_{max}$
  - This prevents a path becoming over saturated, which would cause all ants to follow the same path.
  - Prevents over exploitations

# Max-Min Ant System: Trail smoothing

While MMAS was less susceptible to stagnation than AS, it still occurred. As a result Stützle and Hoos added the following mechanism

- If **stagnation** was detected then all arcs $(i, j)$ undergo "trail smoothing" by adding

$$\Delta\tau_{i,j} \propto \tau_{max} - \tau_{i,j} \tag{19}$$

  namely, all pheromone concentrations are increased proportional to the difference with the maximum bound.
- Using this smoothing strategy, stronger pheromone concentrations are proportionally less reinforced than weaker concentrations.

# Max-Min Ant System: Stagnation

The stagnation detection method was simple: if the mean $\lambda$-branching factor ($\bar{\lambda}$) dropped below a very small predefined constant the search was in a state of stagnation: where

$$\bar{\lambda} = \frac{\sum_{i \in V} \lambda_i}{N_G} \tag{20}$$

where $\lambda_i$ is defined as the number of links leaving node $i$ with $\tau_{ij}$ values greater than $\lambda(\tau_{i,max} - \tau_{i,min}) + \tau_{i,min}$, where

- Let $\tau_{i,max}$ and $\tau_{i,min}$ be the max and min pheromone intensity of links leaving node $i$.

# Fast Ant System

The fast ant system (FANT) was designed to solve the quadratic assignment problem (QAP). A QAP is a placement problem. There are many ways to understand the QAP but the simplest is in graph form.

- There a $n$ facilities that need to be assigned to $n$ locations.
- There exists a link between each location pair, which contains a distance (could be some other cost)
- There exists a link between each facility pair, which contains a flow (how much product needs to be moved between the facilities)
- Placement of facilities must be made to minimize the sum of products between the distance and flow on each arc.
- *In the above description a 0 weight is how the lack of an arc is modeled

# Fast Ant System

The two main difference between FANT and AS are:

- FANT uses only one ant.
- A different pheromone update rule is applied which does not make use of any evaporation.

The pheromone update rule is:

- Let $l(t)$, and $g(t)$ be iteration best and global best paths respectively
- Then

$$\tau_{i,j}(t+1) = \tau_{i,j}(t) + w_1\Delta_{i,j}^{(l)} + w_2\Delta_{i,j}^{(g)} \tag{21}$$

where

$$\Delta_{i,j}^{(l)} = \begin{cases} 1 & \text{if } (i,j) \in l(t) \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

$$\Delta_{i,j}^{(g)} = \begin{cases} 1 & \text{if } (i,j) \in g(t) \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

# Fast Ant System

- $\tau_{i,j}(0)$ is initialized to 1
- If a new global best path is found $\tau_{i,j}$ is reinitialized.
- If a iteration yields no improved global best path, $w_1$ is increased by 1 to promote exploration.