Ant Colony Optimisation[a]

Dr Dimitris C. Dracopoulos

email: `d.dracopoulos@westminster.ac.uk`

Course web page:

`http://users.wmin.ac.uk/~dracopd/DOCUM/courses/2ait608/ait608.html`

[a]These slides are based on the following technical report: `http://iridia.ulb.ac.be/IridiaTrSeries/IridiaTr2006-023r001.pdf`

Ant Colony Optimisation (ACO) and Swarm Intelligence

Inspired from *stigmergy* which describes a type of communication in which the "workers are stimulated by the performance they have achieved".

- *Indirect form of communication*: insects exchange information by modifying their environment.
- *Information is local*: it can only be accessed by those insects that visit the locus or the neighbourhood in which its was released.

Example: In many ant species, ants walking from to and from a food source deposit on the ground *pheromone*. Other ants perceive the pheromone and tend to follow the paths where the pheromone concentration is higher.

Choosing a path using ACO

Assuming that there are two equal paths from nest to food:

- Due to random fluctuations, after some time one of the bridges contains more pheromone than the other, thus attracting more ants (depositing further pheromone).
- Eventually all the ants will following the path to the same bridge due to the amount of pheromone in it.

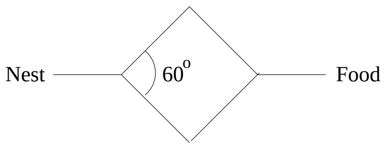

Figure 1: Double bridge with equal path lengths for ant colony optimisation.

Choosing a path using ACO (cont'd)

For paths with different lengths:

- Ants choosing by chance the shortest path reach the next first.
- The short bridge receives pheromone earlier than the long one.
- Therefore, the probability of further ants selecting the short path is increased.

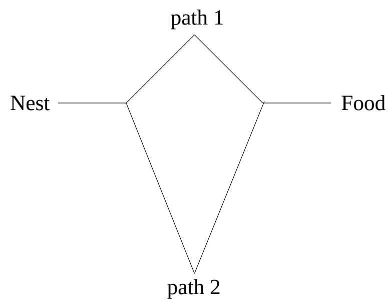

Figure 2: Double bridge with different path lengths.

Probability selecting a path

Assume that m_1 ants have used the first path, and m_2 ants have used the second one.

The probability of choosing the first bridge is:

$$p_1 = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h} \tag{1}$$

where k and h are chosen using Monte Carlo simulations on the specific data (typically $k \approx 20$, $h \approx 2$).

Probability of choosing the second bridge:

$$p_2 = 1 - p_1 \tag{2}$$

ACO for the Travelling Salesman Problem (TSP)

Using ant colony optimisation, TSP can be tackled by simulating a number of ants moving in the problem graph.

- Pheromone is associated with each edge and can be modified by ants.

At each iteration of ACO, a number of ants are considered:

1. Each ant builds a full solution by walking from vertex to vertex, with the constraint of not visiting any vertex already visited.
2. At each step of the full solution construction, an ant selects the following vertex to visit stochastically, based on pheromone:
   - if j has not been visited before, it can be selected with probability proportional to the pheromone associated with edge (i, j).
3. At the end of an iteration, the pheromone values on each edge are modified, based on the quality of solutions constructed by ants.

The ACO Metaheuristic

Ant colony optimisation has been formalised as a metaheuristic for combinatorial optimisation problem.

A model $P = (S, \Omega, f)$ of a combinatorial optimisation problem consists of:

- a search space S defined over a finite set of discrete decision variables $X_i, i = 1, \ldots, n$.
- a set Ω of constraints among the variables.
- an object function $f : S \rightarrow R$ to be minimised.

Variable X_i takes values in a set D.

- A feasible solution $s \in S$ is a complete assignment of values to variables that satisfies all constraints in Ω.
- A solution s^* is a global optimum if and only if: $f(s^*) \leq f(s), \quad \forall s \in S$

Revisiting the TSP

A solution can be represented with a set of n variables, where n is the number of cities.

- The X_i value indicates the city to be visited after city i.
- A solution consists of components, which are pairs of cities (edges) to be visited one after each other. $c_{ij} = (i, j)$ describes that j should be visited immediately after i.
- Ants deposit pheromone on the edges of the graph. The amount $\Delta\tau$ of the pheromone deposited may depend on the quality of the solution found.

Pseudocode for Ant Colony Optimisation

```
Set parameters;
Initialise pheromone trails;
while termination condition not met do
        Construct Ant Solutions;
        Apply Local Search (optional);
        Update parameters;
end while
```

Figure 3: Pseudocode for the ant colony optimisation algorithm.

Application of ACO Pseudocode to TSP

- *Construct Ant Solutions*: m ants construct solutions from components $C = \{c_{ij}\}, \ \ i = 1, \dots, n, \ \ j = 1, \dots, n, \ i \neq j$.
  - A solution construction starts from an empty partial solution $s^p = \emptyset$.
  - At each construction step, the partial solution s^p is extended by adding a feasible solution component from the set $N(s^p) \subseteq C$, which is the set of components that can be added to the current partial solution s^p without violating constraints in Ω.
  - The choice of a solution component from $N(s^p)$ is done stochastically, based on the pheromones associated with each element in $N(s^p)$. The rule for stochastic selection varies according to the specific ACO algorithm but typically is based on equation (1).

Application of ACO Pseudocode to TSP (cont'd)

- *Apply Local Search*: once solutions have been constructed, and before updating the pheromones on the edges, solutions can be improved by applying local search techniques. This step is optional and problem-specific.
- *Update Pheromones*: aim is to increase pheromone values associated with good solutions and decrease those associated with bad solutions. Usually this is achieved by:
  1. decreasing all the pheromone values using *pheromone evaporation*
  2. increasing the pheromone levels associated with a chosen set of good solutions (not necessarily all good solutions)

Main ACO Algorithms

Although several ACO algorithms have been proposed, three of the main ones are:

- Ant System (AS)
- Max-Min Ant System (MMAS)
- Ant Colony System (ACS)

These are described next in the context of TSP.

The Ant System (AS) Algorithm

The main characteristic of AS is that in each iteration, the pheromone values are updated by all the m ants that have built a solution in that iteration.

- The pheromone τ_{ij} associated with the edge joining cities i, j is updated using:

$$\tau_{ij} \longleftarrow (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{3}$$

where ρ is the evaporation rate, m is the number of ants, and $\Delta\tau_{ij}^{k}$ is the quantity placed on edge (i, j) by ant k:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k}, & \text{if ant } k \text{ used edge } (i,j) \text{ in its tour} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where Q is a constant, and L_k is the length of the tour constructed by ant k.

The Ant System (AS) Algorithm (cont'd)

- In the construction of a solution, an ant selects the city to be visited next, using a stochastic approach. When ant k is in city i and so far has constructed the partial solution s^p, the probability of visiting city j next, is given by:

$$p_{ij}^{k} = \begin{cases} \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{c_{il} \in N(s^p)} \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}}, & \text{if } c_{ij} \in N(s^p) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $N(s^p)$ is the set of valid components, i.e. edges (i, l) where l is a city not visited yet by ant k. Parameters α, β control the relative importance of the pheromone versus the heuristic information η_{ij} which is given by:

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{6}$$

where d_{ij} is the distance between cities i and j.

The MAX-MIN Ant System (MMAS) Algorithm

Improvement over the original AS. Only the best ant updates the pheromone trails in every iteration, with pheromones having a lower and an upper bound.

- Pheromone update rule:

$$\tau_{ij} \longleftarrow \left[(1-\rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{\text{best}}\right]_{\tau_{\min}}^{\tau_{\max}} \tag{7}$$

where $\tau_{\min}, \tau_{\max}$ the lower and upper bounds of the pheromone values respectively.

$[x]_b^a$ is defined as:

$$[x]_b^a = \begin{cases} a, & \text{if } x > a \\ b, & \text{if } x < b \\ x, & \text{otherwise} \end{cases} \tag{8}$$

The MAX-MIN Ant System (MMAS) Algorithm (cont'd)

and $\Delta\tau_{ij}^{\text{best}}$ is:

$$\Delta\tau_{ij}^{\text{best}} = \begin{cases} \frac{1}{L_{best}}, & \text{if } (i,j) \text{ belongs to the best tour} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

L_{best} is the length of the tour of the best ant. This can be the best tour in the current iteration or the best solution found so far.

The bounds of the pheromone are determined empirically and tuned by hand for the specific problem.

The Ant Colony System (ACS) Algorithm

Introduces an additional *local pheromone update* which is done after each construction step (i.e. add of an edge to the current partial solution).

- All the ants perform a local pheromone update after each construction step. This is applied only to the last edge traversed (added in the partial solution):

$$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 \tag{10}$$

  where $\varphi \in (0, 1]$ is the pheromone decay coefficient and τ_0 is the initial value of the pheromone.

  The main goal of the local update is to diversify the search performed by the subsequent ants during an iteration. Decreasing the pheromone on the last edge traversed will encourage subsequent ant to choose other edges and produce different solutions.

The Ant Colony System (ACS) Algorithm (cont'd)

- The offline pheromone update (i.e. the one done at the end of each iteration) is similar to MMAS but slightly different:

$$\tau_{ij} \longleftarrow \begin{cases} (1-\rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij}, & \text{if } (i,j) \text{ belongs to best tour} \\ \tau_{ij} & \text{otherwise} \end{cases} \tag{11}$$

  As in MMAS:

$$\Delta\tau_{ij} = \frac{1}{L^{best}} \tag{12}$$

Theoretical Results

- Convergence theory exists.
- However, cannot predict how quickly optimal solutions will be found

Applications

- Routing problems (e.g. distribution of goods)
- Assignment/scheduling problems (assigning objects/tasks to resources)
- Routing in telecommunication networks
- ...