Ant Colony Optimisation: Algorithms and Applications

Author
Paul Sharkey

March 6, 2014
1 Introduction

Ant Colony Optimisation (ACO) is an example of how inspiration can be drawn from seemingly random, low-level behaviour to counter problems of great complexity. A specific focus lies with the collective behaviour of ants after being confronted with a choice of path when searching for a food source (see Figure 1). Ants deposit pheromones on the ground having selected a path, with the result that fellow ants tend to follow the path with a higher pheromone concentration. This form of communication allows ants to transport food back to their nest in a highly effective manner. The experiments of Deneubourg et al. [4] highlighted this behaviour. In this process, ants are given a choice of two equally-attributed bridges to traverse in order to reach a food source. Following random fluctuations, one of the bridges presents a higher pheromone concentration. Eventually, the entire colony converges toward the use of the same bridge.

Figure 1: This shows the potential paths a colony can take from nest (N) to food (F), with the route eventually converging as a result of random fluctuations in pheromone deposits.

This report will review some of the most-recognised ACO algorithms and their usage in the solution of combinatorial optimisation problems. In Section 2, the algorithms will be introduced in the context of the famous travelling salesman problem. In Section 3, an application of this class of algorithms to routing problems in telecommunication networks will be discussed. Section 4 will conclude the review.

2 ACO algorithms

In the travelling salesman problem, a number of cities and the distances between each are provided. The objective is to find the shortest path the salesman can take, so that each city is visited exactly once. See [2] for a thorough formulation of the travelling salesman problem. ACO considers solutions to this problem built by artificial ants, which are simulated to reflect the behaviour of actual ants. A connected graph is constructed to represent the problem at hand. Each node represents a city and each edge represents a connection between two cities. A variable called pheromone is associated with each edge that is recognised and modifiable by the ants. These ants construct solutions by walking from node to node. At each step, the ant must choose a path based on a stochastic mechanism that is biased by the pheromone. If a city \( j \) has not been previously visited, it is selected with a probability proportional to the pheromone laid on the edge \((i, j)\) between
cities. At the end of each iteration, the pheromone values of a promising set of solutions, i.e. short tours, are increased, while for bad solutions, the pheromones decrease through evaporation. It is by the treatment of this pheromone update that ACO algorithms tend to differ. See [6] for a more formalised description of ACO as a meta-heuristic for combinatorial optimisation problems.

2.1 Ant System

The Ant System (AS) algorithm, developed by Dorigo [8], is the original proposed ACO algorithm. After each iteration, the updating of pheromone values incorporates the contribution of all $m$ ants that have built a solution in the iteration itself. Let $\tau_{ij}$ be the value of the pheromone on the edge connecting cities $i$ and $j$. This value is updated by:

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

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

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

where $Q$ is a constant and $L_k$ is the tour length of the $k$th ant. It is evident from this expression that more pheromone is deposited along shorter tours. The transition probability of ant $k$ going from city $i$ to city $j$ is defined as:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \nu_{ij}^\beta}{\sum_{c_{ij} \in N} \tau_{ij}^\alpha \nu_{ij}^\beta} & \text{if } c_{ij} \in N; \\ 0 & \text{otherwise,} \end{cases}$$

where $N$ is the set of feasible components and $\nu_{ij}$ is defined as the visibility, which represents the heuristic contribution to the transition probability. Dorigo defined $\nu_{ij}$ as the reciprocal of the distance between two cities. This means that close towns should be selected with high probability. The parameters $\alpha$ and $\beta$ control the relative importance of pheromone versus the heuristics. In other words, the transition probability is a trade-off between city proximity and pheromone intensity. Bullnheimer et al. [3], when applying this algorithm to the vehicle routing problem, found that defining the visibility in terms of a parametrical saving function produced better results.

The major concern of this algorithm is the impact of pheromone trails on the solution search. The long-term goal is to reduce the search space of solutions and concentrate on a small number of arcs. However, if the ants are heavily influenced by the pheromone in the early stages of the search, this prevents a comprehensive exploration of possible solutions. To track this, Gambardella and Dorigo [10] introduced a $\lambda$-branching factor that gives an indication of the dimension of the search space. Letting $\tau_{ij}^{max}$ and $\tau_{ij}^{min}$ be the maximal and minimal pheromone levels on edges exiting from node $i$, then the $\lambda$-branching factor is the number of exiting edges having a pheromone level $\tau > \tau_{ij}^{min} + \lambda(\tau_{ij}^{max} - \tau_{ij}^{min})$. In AS, this value reduces drastically in the early stages of computation as the ants converge to a common path. However, the Ant-Q algorithm [10], for example, performs better by having ants explore subsets of the search space while still reducing the dimension of this space. Local pheromone updating, as will be discussed in Section 2.3, is another measure aimed at encouraging further exploration of the search space.
2.2 Max-Min Ant System

Following [12], the Max-Min Ant System (MMAS) algorithm represents an improvement over the original AS. In contrast to AS, the “best” ant updates the pheromone trails alone. This typically corresponds to the tour of the iteration-best ant or the best found solution during the run of the algorithm. To avoid the production of identical solutions and increase exploration of the search space, limits on the pheromone values are imposed. This helps to prevent edges with extreme pheromone levels having an excessive influence on the solution search and consequently reducing the dimension of the search space. The update is configured as follows:

\[ \tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \Delta \tau_{ij}^{\text{best}}, \]

with the constraint that there exists \( \tau_{\text{min}} \) and \( \tau_{\text{max}} \) such that \( \tau_{\text{min}} \leq \tau_{ij} \leq \tau_{\text{max}} \) and where \( \Delta \tau_{ij}^{\text{best}} \) is given by:

\[ \Delta \tau_{ij}^{\text{best}} = \begin{cases} 1/L_{\text{best}} & \text{if } (i, j) \text{ belongs to the shortest tour;} \\ 0 & \text{otherwise,} \end{cases} \]

The values of \( \tau_{\text{min}} \) and \( \tau_{\text{max}} \) are typically obtained empirically and are influenced by the problem in consideration. The values are carefully chosen in order to avoid stagnation of the search, but also to ensure good solutions are found in an efficient manner.

2.3 Ant Colony System

The Ant Colony System (ACS) algorithm [7] varies from AS in the introduction of a local pheromone update in addition to the update performed at the end of the solution building process. Local updating encourages exploration of the search space by decreasing pheromone levels on traversed edges. Each ant applies the following update to the last edge traversed:

\[ \tau_{ij} \leftarrow (1 - \phi)\tau_{ij} + \phi\tau_0, \]

where \( \phi \) is the local evaporation rate and \( \tau_0 \) is the initial value of the pheromone. Updating also takes place at the end of the iteration and similarly to MMAS, the update is applied by the ant with the shortest tour. This is intended to reward edges belonging to shorter tours. The modified update is thus:

\[ \tau_{ij} \leftarrow \begin{cases} (1 - \rho)\tau_{ij} + \rho\Delta \tau_{ij} & \text{if } (i, j) \text{ belongs to the shortest tour;} \\ \tau_{ij} & \text{otherwise,} \end{cases} \]

ACS also differs from AS in the decision rule applied by the ants when travelling from cities \( i \) to \( j \). The probability of transition depends on a uniformly distributed random variable \( q \) and a parameter \( q_0 \). If \( q \leq q_0 \), then \( j = \arg \max_{c_{ij} \in N} \{\tau_{il}\nu_{il}^{\beta}\} \). If not, then the transition probability in Section 2.1 is used. This formulation suggests that an ant can decide, with probability \( q_0 \), to exploit the experience accumulated by the ant colony based on the higher pheromone levels on the edges belonging to the shortest tours. Alternatively, with probability \( (1 - q_0) \), an ant can apply a biased exploration that incorporates heuristic information as well as the edges belonging to shortest tours.

3 Application to Telecommunication Networks

Unsuccessful call connections in telecommunication networks are frequently blamed on insufficient approaches to routing problems. ACO algorithms have been applied to counter this problem, particularly when key aspects of the network vary over time, as is the case when the call volume
grows unusually high, for example. The concept was first introduced in telephone networks [11], and developed to the point of being state-of-the-art in wired networks, rivalling and outperforming former mainstays in the field. An example of this is the comprehensive and highly adaptive AntNet routing algorithm, presented by Di Caro and Dorigo [5]. More recently, ACO algorithms were developed for use in mobile ad hoc networks and were found to be competitive with the current leaders in the field [9]. This section will give a brief overview of both the usage of ACO algorithms in achieving load balancing in telephone networks and the AntNet algorithm.

3.1 Load Balancing in Telephone Networks

Switch-based telecommunication networks, such as telephone networks, can be represented by an undirected graph. Each node represents the switching station, while the edges between nodes represent communication channels between callers. When congestion occurs in the network, load balancing distributes activity evenly over the nodes with the aim to minimise the number of lost calls [11]. A node tends to be linked only to its geographical neighbours. Each node has the following key attributes:

- **Capacity** - the amount of simultaneous calls the node can handle.
- **Routing Table** - each entry informs which is the next node on the route to the destination node.
- A probability of being the source or destination node of a call.

![AntNet Algorithm Diagram](image)

Figure 2: This shows a possible network configuration and pheromone table. An ant travelling from node 1 to node 3 chooses to travel via node 2 with probability 0.44 and via node 4 with probability 0.56, for example.

Upon implementing an ant-based control mechanism on the network model, the routing tables are replaced by “pheromone tables”. Every node has a pheromone table for every possible destination in the network, and each table has an entry for every neighbour. After initialising an ant at a given node, it moves from node to node according to the probabilities in the pheromone tables for the destination node (see Figure 2). At the next node, the ant updates the probabilities of that node’s pheromone table entries corresponding to their source node. The table is altered to increase the probability of returning to previous nodes. The entry is increased according to the simple formula:

\[ p_{\text{new}} = p_{\text{old}} + \Delta p \]

where \( \Delta p \) is the probability increase. The other entries are reduced so that the sum of the entries is 1. In order to encourage the ant to find routes that are relatively short, \( \Delta p \) is chosen so that is
reduces with the age of the ant. This biases the network in favour of ants that have taken shorter tours. Upon returning to the source node, the ant dies.

Calls from a source node will choose the route to its destinations by means of the entries in the pheromone table. Calls and ants interact with each other through these probabilities. Ants will influence the routes by the update of the pheromone tables, which in turn will influence the route taken by a call. This dynamic can be seen in Figure 3.

Figure 3: This figure shows the relationship between calls, nodes, pheromone tables and ants [11]

This approach has been compared with mainstay methods using mobile software agents (see [1, 11]) and was found to outperform them.

### 3.2 AntNet

Outlined in [5], AntNet is a routing algorithm based on the concept in Section 3.1 and first applied to packet-switched networks, such as the Internet. The algorithm follows the same principles as that discussed with regard to telephone networks. Ants search for a minimum cost path connecting the source and destination nodes. The choice of path is made according to a similar probabilistic rule. However, this probability (referred to as the goodness) incorporates heuristics specific to the problem. Upon reaching the destination node, ants return to the source updating the routing table in the process, where they eventually die. AntNet has been proven to be highly adaptive and robust under application to different networks and scenarios. It is regarded as being among the state-of-the-art routing algorithms.

### 4 Conclusion

This report has given a brief overview of Ant Colony Optimisation, an approach inspired by the behaviour of ant colonies aimed at approximating solutions to complex combinatorial optimisation problems, in particular. The operation of these ant colonies are reflected in algorithms designed to find the shortest path on a connected graph. These artificial ants are driven by pheromone deposits laid on the edges of this graph. This report has described the evolution of these algorithms, with a focus on encouraging efficient exploration of the search space in order to prevent the discovery of trivial solutions. It is through the pheromone update that these algorithms tend to differ.

This report also introduced an application of ACO to routing problems in telecommunication networks, which helps to identify and counter congestion issues that result in lost connections. Calls and ants choose routes based on their associated pheromone levels. A vast array of literature
exists on this topic. AntNet, in particular, is highly regarded. Current research in ACO includes a focus on whether these algorithms are convergent (which has been found to be the case for MMAS and ACS). In recent years, ACO has also been applied to problems in systems biology, including protein folding. Areas of future research include application to optimisation problems that include degrees of stochasticity, dynamic data modifications and multiple objectives. The idea of using ant behaviour as a basis for developing algorithms to tackle such problems of great complexity may have seemed infeasible in the past. However, ACO is now regarded as a novel approach to solving these problems, whose vast array of applications indicates that this field will expand over time.

References


