Ants, stochastic optimisation and reinforcement learning

Andre Costa

Abstract

Ant colonies are successful and resilient biological entities, which exhibit a number of desirable collective problem-solving behaviours. The study of ant colonies has recently inspired the development of artificial algorithms for stochastic optimisation and adaptive control, which attempt to mimic some of the properties of the biological counterpart. In this paper, we give a brief overview of this class of algorithms, and compare it with other popular stochastic optimisation methods. In addition, we discuss the connections between ant-based algorithms and the well-established field of reinforcement learning.

1 Introduction

Have you ever opened your pantry cupboard, only to find that the honey jar is teeming with ants? Such events are the result of many years of biological evolution, which has endowed ants with some remarkable collective resource-discovery and problem-solving behaviours. An interesting aspect of such phenomena is that each individual ant possesses only a very limited intelligence. Yet when each ant follows a simple set of rules, the emergent behaviour of the ant colony as a whole can appear to be "intelligent". The key to understanding such behaviour lies in the identification of the rules followed by individual ants, and how these rules give rise to positive feedback processes which reinforce successful behaviours at the level of the ant colony. In addition, a certain degree of randomness at the individual ant level is an important factor in the ability of the ant colony to adapt and survive in a changing environment.

The success and resilience of ant colonies in the natural world has recently inspired the development of artificial algorithms which mimic biological ant behaviour in order to solve difficult combinatorial optimisation problems, such as the well-known travelling salesman problem, or to perform adaptive real-time control of complex systems, such as a telecommunications network. Algorithms belonging to this class are known as Ant Colony Optimisation (ACO) algorithms, and the common underlying approach is referred to as the ACO method [2]. Like the biological counterpart that inspired it, the ACO method incorporates a distinct stochastic element, which drives the discovery of new and hopefully better solutions. Thus, ACO belongs to the growing family of stochastic optimisation methods, which includes simulated annealing [1], genetic algorithms [6] and the cross-entropy method [9]. In addition, there exist a number of interesting connections between the ACO method and the field of reinforcement learning [10], which has in recent decades played a prominent role in machine learning and adaptive control.

This paper is organized as follows. In Section 2, we describe some properties of biological ants, and give an example of the type of collective problem-solving behaviours of ant colonies that inspired the development of the ACO method. We give a brief introduction to the ACO method in Section 3, and discuss the similarities and differences compared with other stochastic optimisation methods. In Section 4, we highlight the connections between ACO and reinforcement learning. Conclusions and directions for future research are given in Section 5.

2 Biological ants and collective problem-solving

A well-known experiment involving an ant colony, known as the double bridge experiment [4], has been performed in the laboratory, and serves to highlight how simple rules followed by individual ants are able to give rise to a collective decision-making process. Some of these simple rules, deduced via empirical observation of ant behaviour, are that

- ants deposit a chemical pheromone as they travel, and
- ants are able to detect differences in pheromone concentration in their surroundings, and will tend to move in the direction where the concentration is highest.

In addition, as in many natural systems, there exists some degree of random fluctuation. In particular, an individual ant occasionally follows a path with a lower or even zero pheromone concentration level. This type of behaviour is referred to as exploration. Furthermore, pheromone diffuses and evaporates as time passes, which means that pheromone trails must receive continual reinforcement in order to remain in existence. In the double bridge experiment, ants are allowed to establish a pheromone trail between their nest and a food source, as shown in Figure 1a. An obstacle is then placed in the path of the ants, in such a way that the ants have a choice of two paths around the obstruction, one of which is twice as long as the other. Initially, there is no pheromone on either the long or short path, and therefore ants have equal probabilities of choosing either path. This results in approximately half of the ants initially choosing the long path and half choosing the short path (Figure 1b).

Now consider that ants deposit pheromone at an approximately constant rate, travel at approximately the same speed, and travel in both directions between the nest and food source. As a result, after a given period of time, more ants have traversed the entire length of the short path than have traversed the length of the long path (Figure 1c). This means that after the given period of time, the *concentration* of pheromone on the short path is higher than the concentration on the long path. A positive feedback process ensues, whereby after some time, most of the ants take the short path (Figure 1d). Thus, the ants have collectively found the optimal solution to the shortest path problem.

It is worth noting that the positive feedback that occurs in the double bridge experiment is essentially a transient effect, arising from the fact that the pheromone concentration increases at a faster rate on the short path. It is also found that once a stable pheromone field is established on the short path, if a shorter path becomes available, then the ants are not always able to switch to it, and continue instead to travel on the same path [4]. This indicates that the positive feedback effect favouring the short path is strongly dependent upon the initial conditions, in this case, the initial absence of pheromone on either path.

This serves to highlight the fact that in designing an artificial ant-based system, it is desirable and indeed often necessary, to augment the artificial ants with properties and behaviours which are not possessed by their biological counterparts. For example, the above stagnation effect might be avoided if ants were somehow able to deposit pheromone in amounts that reflect the length of the path that they are travelling on. This would provide an additional mechanism for the differential reinforcement of the pheromone concentration on paths of different lengths, in a manner that would not be solely dependent upon the initial conditions.

In designing an artificial ant-based algorithm, a differential reward mechanism can easily be introduced, whereby instead of depositing artificial pheromone at a constant rate, artificial

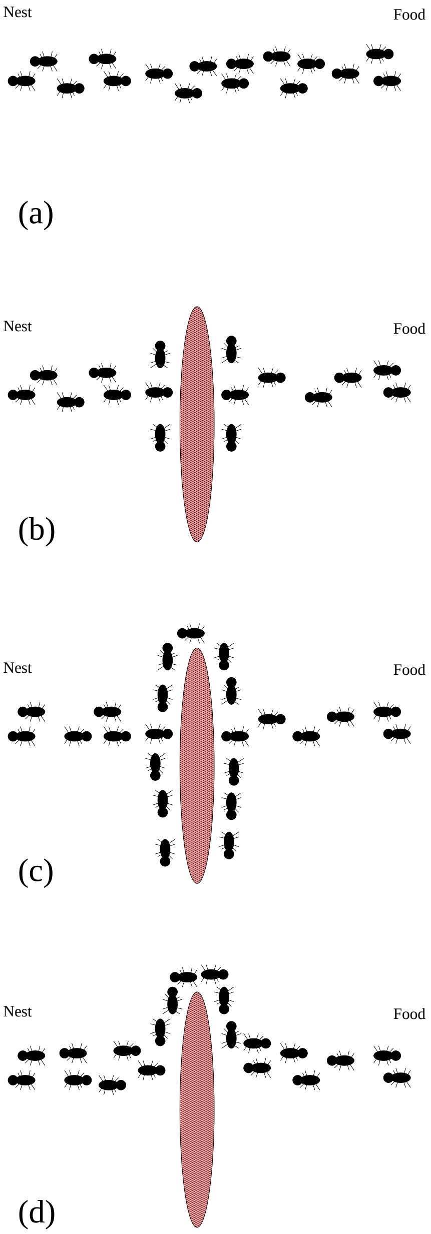

Figure 1. The double bridge experiment.

ants deposit pheromone in quantities that are inversely proportional to the cost associated with the paths they travel on. This can be accomplished by programming the ants to effectively retrace their steps, and deposit pheromone *after* a path has been completed, and the path cost is known.

Indeed, in a different experiment, a simple differential reward mechanism is present in the biological ant colony. In the experiment, ants have the choice between two paths leading from the nest to a high and low quality food source, respectively. The ant colony is able to choose the higher quality food source in the sense that after some time, the majority of ant traffic is directed on the path that leads to the high quality food source. The underlying mechanism which enables this choice, is that ants returning from the high quality food source deposit pheromone at a faster rate than the ants returning from the low quality source.

An interesting theoretical analysis of the above experiments using statistical mechanics appears in [8]. It is shown that the ant colony's collective problem-solving behaviour is present only in certain critical regions of the ant colony's parameter space, comprising the rate at which individual ants deposit pheromone, the rate of pheromone evaporation, and the ant density.

The development of artificial ant-based systems, in which some of the biological ant behaviours described above are reproduced *in silico*, is the subject of the rest of this paper.

3 Ant-based algorithms for discrete stochastic optimisation

The type of problem-solving phenomena described in Section 2 inspired the invention of the ACO method [2] for solving discrete optimisation problems.

Consider a discrete optimisation problem, where $\mathcal{X}$ represents the set of all feasible solutions, and the objective function $S : \mathcal{X} \to \mathbb{R}$ assigns a cost to every solution $x \in \mathcal{X}$. We seek a solution which minimises the cost function S, that is, an optimal solution x^* satisfying

$$S(x^*) \leq S(x), \quad \text{for all } x \in \mathcal{X}. \tag{1}$$

When the number of feasible solutions is very large, exhaustive search is not an appropriate approach, and often a heuristic search method must be employed. These fall into two main categories: deterministic and stochastic. A well-known deterministic search method is the branch and bound method [11]. In contrast, ACO is a stochastic search method, because candidate solutions are generated by a random process, which we outline shortly.

The first step in applying the ACO method is to encode the optimisation problem in such a way that every feasible solution can be uniquely represented as a walk on a directed graph, $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of vertices and $\mathcal{E}$ is a set of edges. Each solution can thus be represented as an ordered sequence of vertices, or equivalently, edges. Each edge has an associated cost, and the total cost of the solution is given by the sum of the costs of the constituent edges.

As an example, consider the celebrated travelling salesman problem [11]; in this instance, $\mathcal{V}$ would represent a set of cities, and $\mathcal{E}$ would represent transport connections between pairs of cities. The set of feasible solutions, $\mathcal{X}$, also known as tours, consists of all ordered sequences of cities (or connections) for which each city appears exactly once. An additional constraint is often imposed, whereby all tours must start and finish at a given city. A cost is associated with the traversal of each edge, reflecting, for example, geographical distance or transportation costs between the corresponding pair of cities. The goal is to find a tour which minimises the total incurred cost.

In general, the ACO method involves an iterative procedure, where the following two steps are performed at each iteration:

- *Step 1.* A random sample of candidate solutions is generated according to a parameterized probability distribution.
- *Step 2.* The candidate solutions generated in Step 1 are scored using the objective function S, and the parameters of the probability distribution are updated with the

aim of *increasing* the probability that the best solutions found so far will occur in the next iteration.

We now describe each of these steps in greater detail.

In Step 1, the random sample of candidate solutions is generated by a set of artificial ants, where each ant constructs a solution by performing a walk on the graph $\mathcal{G}$. The parameters of the probability distribution governing these walks are the set of artificial pheromone values $\tau(e), e \in \mathcal{E}$, which are non-negative, and reflect the degree to which ants are attracted to each edge when performing a walk. In particular, each ant begins at a given starting vertex of the graph, and selects a sequence of edges. In order to describe this process, consider a given ant $\mathcal{A}$, that has performed a partial walk, having thus constructed a partial solution, and suppose that the ant is currently located at the vertex v. Denote using $\mathcal{E}(v|\mathcal{A})$ the set of outgoing edges at v which would maintain feasibility of the solution encoded by the ant's walk, given the identity of the partial solution it has constructed[1]. Then the probability $p(e)$ that edge e is selected by the ant as its next step, is given by a relation of the form

$$p(e) \propto \begin{cases} \tau(e) & \text{if} \quad e \in \mathcal{E}(v|\mathcal{A}), \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

This process is performed at each subsequent vertex, and typically results in the construction of a feasible solution. In the event that at some vertex v, the set $\mathcal{E}(v|\mathcal{A})$ is empty, such that the ant is unable to construct a feasible solution, then a number of techniques to efficiently re-initialize the ant can be employed [2].

In Step 2, the candidate solutions generated in Step 1 are used to update the pheromone values. In particular, suppose an ant constructs a solution $x = (e(1), e(2), \ldots, e(K))$, where $e(i), i = 1, \ldots, K$, is the sequence of constituent edges. Once the solution is constructed, the corresponding cost $S(x)$ is computed, and each of the pheromone values $\tau(e(k)), k = 1, \ldots, K$, corresponding to the edges that were "walked" by the ant, are increased according to

$$\tau(e(k)) := \tau(e(k)) + \Delta,$$

where

$$\Delta \propto \frac{1}{S(x)}. \tag{3}$$

In addition, the update

$$\tau(e) := \kappa\tau(e) \tag{4}$$

is performed for every edge $e \in \mathcal{E}$, where $\kappa \in (0,1)$. Finally, a renormalization of the pheromone values is usually performed, in order to improve the numerical stability of the method.

We note that the the updates (3) and (4) perform roles that are analogous to the deposition and evaporation of chemical pheromone in a biological ant colony, respectively. We see from (3) that the pheromone values on edges which are "walked" by ants which construct the best solutions at a given iteration receive the largest reinforcement. In addition, the pheromone values on edges which do not occur in any ant walks gradually decay due to (4). A variety of functions to perform the transformations (2) and (3) have been used in ACO algorithms, and it is possible to include additional problem-specific information at this stage. The reader is referred to [2] for details.

Typically, after a number of iterations, the pheromone values parameterize a probability distribution which contains most of its probability mass on a set of optimal solutions. More

[1] For example, in the travelling salesman problem, a walk containing a cycle is not a feasible solution, so $\mathcal{E}(v|\mathcal{A})$ would not contain vertices that the ant has already visited.

importantly, an optimal or near-optimal solution is usually discovered by at least one ant. We make some comments regarding convergence properties of ACO algorithms in Section 5.

It is instructive to compare the ACO method with other popular stochastic optimisation methods. ACO differs significantly from population-based methods, such as genetic algorithms [6] and simulated annealing [1], which operate directly on a set of candidate solutions, and generate new candidates by perturbing or modifying existing ones.

In contrast, in the ACO method, the information about promising solutions is contained in a probability distribution that is parameterized by the pheromone values. Although ants are used to construct candidate solutions, there are no operations performed directly on the population of ants themselves. Instead, at each iteration, the ants update the pheromone values, and a completely new set of ants is created at the start of the next iteration. Thus, the information about promising solutions is propagated from one iteration to the next exclusively via the pheromone values. This is in keeping with the behaviour of biological ants, who communicate with each other *indirectly* via a chemical pheromone fields. In this regard, the ACO method is closely related to the cross-entropy method [9] and estimation of distribution algorithms [7], which also perform an iteration on a probability distribution, rather than on a population of solutions.

4 Ants and reinforcement learning

There exist a number of connections between ACO algorithms and the field of reinforcement learning [10], an area which has attracted a great degree of research interest in recent decades. Reinforcement learning algorithms attempt to learn an optimal solution, or strategy, in situations where the objective function S is not explicitly available, but where cost function values can be obtained by observing the outcomes of trials performed on the system to be optimised.

As an example, consider the situation where a robot has to learn to navigate its way through a maze (whose structure is unknown to the robot), by performing repeated trials in the maze, and updating its strategy based on a set of reward or penalty signals that it receives in response to each trial. There exists a large body of research literature on reinforcement learning algorithms that are able to perform such a task. Other applications range from learning optimal chess strategies, to the optimal real-time control of complex manufacturing processes [10].

Biological ant colonies as well as ACO algorithms have a natural interpretation as reinforcement learning algorithms, since ants effectively perform trials, or experiments, on a system whose global structure and parameter values need not be known in advance. The pheromone values in ACO effectively parameterize a randomized strategy, which is updated and improved by observing the outcomes of the ants' experiments and making appropriate changes to the pheromone values.

For example, in order to apply the ACO method described in Section 3 to solve the travelling salesman problem, it is not necessary to know the matrix of edge costs in advance. Instead, the edge costs and thus the costs of complete tours could be measured directly by autonomous ant-like agents in a real instance of the problem. The situation becomes even more interesting and challenging if the optimal solution is non-stationary. The problem is then one of adaptive optimal control. The most prominent example of an application of the ACO method in a non-stationary environment is the AntNet algorithm for adaptive routing in packet-switched communications network [2]. Here, because of unpredictable changes in traffic demands and events such as network component failures, the set of optimal routing policies is typically non-stationary. AntNet employs a swarm of autonomous ants, which attempt to find the paths of least delay between origin-destination node pairs of the network. In particular, ants perform network delay measurements and update routing tables maintained at the network nodes in such a way that low-delay paths are reinforced. Data traffic is then routed on the paths that are recommended by the ants. A reinforcement learning-type approach, such as the ACO method, is highly appropriate for the task of decentralized adaptive network routing, where routing decisions across the network must be made using local measurements and information, in the absence of a central controller.

Exploration plays an extremely important role in reinforcement learning and ACO algorithms in a non-stationary environment. In the stationary optimisation scenario described in Section 3, exploration is important in order to minimise the probability that an ACO algorithm becomes stuck on a set of sub-optimal solutions. In a dynamic environment, exploration performs the additional role of enabling adaptation when the optimal solution changes. In particular, provided that all possible candidate solutions (or strategies) have a positive probability of being generated by an ant, then the ACO method can, in principle, eventually discover the new optimum solution (or strategy). Not surprisingly, there is often a tension between the imperative to explore the space of candidate solutions, and to exploit the solutions that are currently estimated to be the best ones. This tradeoff is strongly manifested in the context of real-time adaptive control, where exploratory actions incur a real cost, since the reward and penalty signals are derived from the actual system which one wishes to optimise, rather than a model. While in many instances the tradeoff described above is unavoidable, it is sometimes possible to *decouple* the tasks of exploration and optimal decision-making. An example of such a situation is in the application of ACO methods to adaptive communications network routing. In [3], a number of modifications to ant-based network routing algorithms such as AntNet are proposed, which achieve this decoupling and improve their performance.

5 Conclusions

The ACO method is a stochastic optimisation technique for discrete optimisation, which can be applied to stationary as well as non-stationary (adaptive control) problems. A key feature of this method is the fact that it performs an iteration on a probability distribution over the set of feasible solutions, with the aim of increasing at each iteration the probability mass located on the set of optimal solutions.

The basis, or motivation, for the ACO method is heuristic; it was inspired by the problem-solving behaviours observed in biological ant colonies, and thus the artificial ants in ACO are loosely modeled on their biological counterparts. However, it has recently been recognized that ACO has much in common with other stochastic optimisation methods, including the cross-entropy method, which is derived from more rigorous mathematical foundations [9]. There is significant scope for gaining a greater understanding of the ACO method via such comparisons, as shown in [12].

Also, there is scope for extending existing convergence results for ACO algorithms. In particular, sufficient conditions are given in [5], which guarantee that an ACO algorithm will eventually generate the optimal solution with probability one. However, these conditions impose a very conservative limit on the rate at which the pheromone values are permitted to change from one iteration to the next, and thus result in extremely slow convergence of the algorithm. In practice, one would allow the pheromone values to change fairly rapidly from iteration to the next, so that within a reasonable amount of time, most of the associated probability mass is located on *some* small subset of solutions, but the tradeoff is that convergence to the set of *optimal* solutions can no longer be guaranteed. The nature of this tradeoff is an interesting and important area for future research on ACO and related methods.

References

[1] E.H. Aarts and J.K. Lenstra, *Local Search in Combinatorial Optimisation*, (Wiley Chichester U.K. 1997).

[2] M. Dorigo, and T. Stutzle, *Ant Colony Optimization*, (MIT Press Cambridge 2004).

[3] A. Costa, *Analytic Modelling of Agent-based Network Routing Algorithms*, PhD Thesis, School of Applied Mathematics, University of Adelaide 2003.

[4] J. Deneubourg, S. Aron, S. Goss, and J. Pasteels, *The self-organizing exploratory behaviour of the Argentine ant*, Journal of Insect Behaviour **3** (1990), 159–168.

[5] W. Gutjahr, *ACO algorithms with guaranteed convergence to the optimal solution*, Information Processing Letter **82** (2000), 145–153.

[6] J. Holland, *Adaptation in Natural and Artificial Systems*, (University of Michigan Press Ann Arbor MI 1975).

[7] P. Larranaga, and J.A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, (Kluwer Academic Publishers 2001).

[8] M. Millonas, *Swarms, Phase Transitions, and Collective Intelligence*, in: *Artificial Life III, SFI Studies in the Science of Complexity*, (Addison Wesley 1994).

[9] R.Y. Rubinstein and D.P. Kroese, *The Cross-Entropy Method. A unified Approach to Combinatorial Optimisation, Monte-Carlo Simulation and Machine Learning*, (Springer Heidelberg 2004).

[10] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, (MIT Press Cambridge MA 1998).

[11] W.L. Winston, *Operations Research Applications and Algorithms*, (Duxbury Press 1994).

[12] M. Zlochin, M. Birattari, N. Meuleau and M. Dorigo, *Model-based search for combinatorial optimisation: A critical survey*, to appear in Annals of Operations Research (2004).

Received 17 December 2004, accepted 21 December 2004.