Ant Colony Optimisation: From Biological Inspiration to an Algorithmic Framework

Technical Report: TR013

Daniel Angus

dangus@ict.swin.edu.au

Centre for Intelligent Systems & Complex Processes
Faculty of Information & Communication Technologies
Swinburne University of Technology
Melbourne, Australia

April 21, 2006

1 Introduction

The Ant Colony Optimisation algorithm framework here-on referred to as ACO is a new algorithmic framework which is inspired by the foraging patterns of biological ants. Any ACO algorithm (of which there are many) serves to optimise some problem instance by generating a series of solutions to the problem and using the utility (goodness) of these solutions to influence future solution construction.

This report outlines the biological inspiration behind the development of the first ant-inspired algorithms. The report then identifies two of these ant-inspired algorithms, their relation to the biological models and offers a contrast and comparison between them. Finally the report describes and analyses the ACO meta-heuristic framework to which a subset of ant-inspired algorithms belong.

The report is organised as follows. Section 2 describes the recruitment and foraging behaviour of four species of ants. Section 3 identifies two ant-inspired algorithms: Ant Systems (AS) and ant colony optimisation for continuous design spaces. Finally Sec. 4 defines the ACO meta-heuristic framework and comments on the properties of the framework and it's relation to ant-inspired algorithms.

2 Biological Beginnings

This section outlines the biological inspirations underpinning the recent developments in ant inspired search algorithms for optimisation. This section highlights specific characteristics of biological models which have been used to develop optimisation algorithms as well as highlighting other properties of the biological systems which may provide useful inspiration for the development of new ant inspired algorithms.

2.1 The Cataglyphis and Ocymyrmex Species

It was mentioned briefly in the introduction that ACO is based on the foraging behaviour of (biological) ants, i.e. how the ants locate and collect food. However so many species of ant exist, each of which exhibits a distinctly different foraging behaviour, that it is more accurate to describe ACO as being inspired by the **recruitment strategy** of ants which use chemical markers (pheromone trails) to mark the location of a rich food source such as the *Iridomyrmex humilus* (Argentine ant) species. Before elaborating on this point though let's have a look at two different species of ant, the *cataglyphis* and *ocymyrmex* ants of the Sahara and Namib deserts respectively.

The Sahara and Namib deserts are two of the most unforgiving environments on our planet, yet two species of ants: the *cataglyphis* and *ocymyrmex* have evolved to fill this unique ecological niche (For a further discussion of these species refer to [14]). Both of these species rely on the harsh environment to provide them with food, since they search their local area for insects which have succumbed to the extreme heat and stress of this harsh environment. What is interesting about these ants is that they don't use a chemical marker to recruit other ants to a food source rather they use an internal memory to influence their own choice of a direction to travel from the nest. The rules for movement of these species have been described as:

- Continue to forage in the direction of the preceding foraging trip whenever this trip has been successful in finding food.
- If foraging trip is unsuccessful then abandon this direction and randomly select a new direction, decreasing the probability of doing so as the number of previously successful runs increases.

This individual behaviour leads to an efficient (for this specific environment) foraging pattern which in the absence of food will result in the ants searching the environment at random, or in the case of food being found will subtract foraging resources away from the global pool of resources to exploit this discovered food source until it is eventually consumed, at which point the colony will revert to its initial behaviour with a slight bias towards searching previously promising areas. It is important to note that this bias will only exist for the life of the ant which found food in this sector, and that if food is found again in this biased area then this ant will die out reverting the colony back to its initial completely random state.

Interestingly enough if one were to hypothesise about placing two different abundant food resources within close and far foraging range of the colony that the emergent effect would be that the members of the colony would split randomly between the resources and stick to one even though one resource is better (i.e. The closer the food to the nest the quicker it is to forage from hence this resource may be considered better). This is because with the simple model described above there is no intra-colony communication mechanism, which would allow the colony to converge on the better resource. This is not of concern however since these species have evolved to exhibit a decentralised control which augers well with their inherently unstable and dynamic environment.

2.2 The Tetramorium Caespitum Species

Tetramorium caespitum ants are quite distinct from the desert dwelling ants described above (This species and its recruitment behaviour is fully described in [12]). These ants have evolved to suit a different ecological niche where food sources are plentiful and the desired effect is to optimise the distribution of resources to maximise the food collection activity. These ants also rely on randomness to influence their decision making behaviour, however with the absence of a long term memory they rely more on intra-colony communication mechanisms to influence their foraging decisions.

This species of ant exhibits three distinct behaviours: group-recruitment, mass-recruitment and random exploration. Group recruitment occurs when an ant finds a new food source, returns to the nest and upon returning to the nest attempts to coerce other ants to follow it back to the food source laying pheromone along the trail as they move. This group recruitment will eventually lead to a mass-recruitment if the food source is large enough and the pheromone trail is reinforced enough so that ants can begin to follow the pheromone trail unled. Random exploration can occur at any stage where an ant following a pheromone trail decides to leave the trail to search virgin territory in the hope of finding more food or a more efficient path to already discovered food. The probability of such an event occurring is inversely proportional to the amount of pheromone and directly proportional to the distance away from the nest.

2.3 The Iridomyrmex Humilus (Argentine Ant) Species

The importance of randomness in the model above is to encourage exploration and avoid exploitation of one food source (or collection path) neglecting other possibly more rich food sources or shorter paths [5]. This emergent effect was perhaps most profoundly demonstrated in the double bridge experiment performed by Denebourg et. al. [4]. In this experiment a single food source was placed away from a nest of *Iridomyrmex humilus* ants and two bridges of unequal length connected the nest to the food. Initially the ants were observed to use both bridges fairly equally to retrieve the food, however eventually the majority of the colony favoured the shorter branch over the longer branch. The researchers explained this emergent (*autocatalytic*) effect by the fact that a shorter distance means that ants can forage on this path more quickly and over time this branch will be positively reinforced with more pheromone.

3 Ant inspired search algorithms

The biological systems described in Sec. 2 are inherently simple on an individual ant level however they all lead to complex emergent properties such as efficient resource allocation and shortest path finding. More-so, these emergent properties exist without the requirement for centralised control of colony resources. It is not unexpected then that these biological systems have attracted a lot of attention in the field of biologically inspired computation leading to the development of a new sub-field loosely referred to as *ant inspired algorithms*. This section outlines two of the first algorithms inspired by (or loosely equivalent to) the biological systems described in Sec. 2.

3.1 Ant Systems

The double bridge experiment of Sec. 2.3 led to the development of three algorithms by Dorigo et al. [8], *Ant-density*, *Ant-quantity* and *Ant-cycle* for application to the Travelling Salesman Problem (TSP) [13]. In these algorithms each (artificial) ant iteratively constructs a solution to the TSP by probabilistically selecting an edge to include in the growing tour based on a nearest neighbour heuristic and an artificial pheromone which is adapted by the artificial ants as the search progresses (much like the mass-recruitment process described in Sec. 2.2), the specific way that this pheromone is added and adapted is what distinguishes these algorithms. After experimenting with the three simple models *Ant-cycle* was shown to be the most effective at optimising the TSP problems addressed. The *Ant-cycle* algorithm was later refined and reintroduced as *Ant Systems (AS)* [9].

3.2 An Ant-inspired Vector-based Algorithm for Continuous Spaces

The ant colony metaphor for searching continuous design spaces proposed by Bilchev and Parmee [2] references the double bridge experiment and the work of Dorigo et al. [8] as it's inspiration, although it is this author's opinion that the *cataglyphis* and *ocymyrmex* species described in Sec. 2.1, are a closer match.

This algorithm starts by selecting a 'nest' (position) in a continuous n-dimensional space which is found by running a global search process on the search domain. Vectors are projected at random from this position into the local area around and a series of random (successively smaller) jumps are made from these initial vectors until a termination criterion is met. If the resultant position is better than the initial vector than the initial vector is replaced. This process is continually repeated with higher quality (more promising) vectors allocated more computational resource than poorer quality vectors.

This algorithm is essentially a local search algorithm which probabilistically selects a solution from a population of solutions and makes a random change to it. The original solution is replaced if the random change is beneficial, and the probability of selecting this solution as a starting point is adjusted positively or negatively based on the result of the random change.

4 Ant Colony Optimisation

From a historical perspective, after the initial work by a few researchers the field of ant-inspired algorithms grew rapidly with a variety of adaptations and novel applications to mostly combinatorial problem domains. As a result the algorithms which had directly quoted Ant Systems as their inspiration were grouped together under a common framework: Ant Colony Optimisation (ACO) [7, 10, 6, 3]. This section defines the ACO meta-heuristic framework, and offers comments on specific features of the framework as well as a critical review of the utility of defining such a framework.

4.1 The Ant Colony Optimisation Meta-heuristic Framework

The ACO meta-heuristic framework (here-on referred to as ACO) can be applied to discrete optimisation problems having a finite set of components with connections between these components (with associated costs). More than this there is also a set of constraints on what components and connections compose a feasible solution, and each feasible solution is said to have a quality which is calculated by a function of all of the component costs. For example an ACO algorithm can be used to find solutions to a TSP problem since this problem is represented by vertices (components) with edges (connections) connecting the vertices each with a unique cost, as well as a constraint which states that all feasible solutions must include every vertex once (and only once).

ACO algorithms use a population of artificial ants to construct feasible solutions to a discrete optimisation problem. The solutions are evaluated according to a fitness function and according to a pre-defined rule implant their solution information in a global memory known as a pheromone mapping where each component of the pheromone mapping corresponds to an individual connection of the problem being optimised. As well as associating a pheromone value (or long-term memory) on each connection a problem specific heuristic can be associated to each individual connection based on a-priori information. These two values together form an estimate as to the utility of this specific connection and bias its inclusion in future solution construction.

Each ant has several common properties (These properties are known as *ants generation and activity*):

- An ant searches for a minimum (or maximum) cost solution to the optimisation problem being addressed.
- Each ant has a memory used to store all connections used to date, and so that the path can be evaluated at the completion of solution construction.
- An ant can be assigned a starting position, for example an initial city in a TSP.
- An ant can move to any feasible vertex until such time that no feasible moves exist or a termination criterion is met (usually correlating to the completion of a candidate solution).
- Ants move according to a combination of a pheromone value and a heuristic value which are associated with every edge in the problem, the choice of where to move is usually a probabilistic one.
- When moving from one vertex to another vertex the pheromone value associated with the edge connecting these vertices can be altered (known as *online step-by-step pheromone update*).
- An ant can retrace a constructed path at the completion of a solution updating the pheromone values of all edges used in the solution (known as *online delayed pheromone update*).
- Once a solution is created, and after completing online delayed pheromone update (if required) an ant dies, freeing all allocated resources.

Beside these properties ACO is responsible for the scheduling of two other processes, *pheromone trail evaporation* and *daemon actions*. The purpose of pheromone trail evaporation is to regulate the amount of pheromone on all connections of the problem being addressed. It is probably best described as a way of forgetting older information to allow newer (better quality) solutions to be reinforced, rather than continually positively reinforcing the older (lower quality) solutions. Daemon actions refers to any centralised process which cannot be performed by a single ant. An example daemon action is a local optimisation procedure applied to an individual ants solution, or the collection of global information to update the pheromone mapping, such as identification of the best solution in order to give this solution's components an extra amount of pheromone (*off-line pheromone update*).

To summarise, ACO is described as being responsible for the scheduling of three processes:

- Ants generation & activity
- Pheromone trail evaporation
- Daemon actions

4.2 ACO and Other Ant-inspired Algorithms

The arguments for the introduction of ACO were to provide a unitary view of on-going research, identify the most important characteristics of the algorithms which belong to the framework, define a common nomenclature and therefore make future development more transferable. Ant-inspired algorithms which do not form part of this framework, such as HAS-QAP [11], ant colony optimisation for continuous design spaces (presented in Sec. 3.2) and the author's own discrete history ant systems (DHAS) [1] are left to be categorised as ant-inspired algorithms. It is said that any

ACO algorithm is an ant-inspired algorithm however the converse is not true. Some of the common properties of ant-inspired search algorithms are:

- Use of a common history repository (pheromone mapping) so that every solution component is assigned a singular value (pheromone value) representing the desirability of using this solution component in future solution construction.
- Stepwise solution construction. Starting with an empty solution (or in some cases a partially built solution) incrementally add solution components until a termination criterion is met.
- Use of a population of multiple individual agents (ants) to construct candidate solutions sequentially or in parallel.

So what does this mean for on-going research in the broader field of ant-inspired algorithms? There still exists some contusion by researchers as to what to call their ant-inspired algorithm, and it is often the case that ant-inspired algorithms are referred to as ACO algorithms even though this may not be strictly true. This confusion is not unexpected though as the framework was defined in 1999 and it is reasonable that it take some time to adopt a correct mainstream usage.

4.3 Distribution in ACO

The argument can be made that if a daemon process is to be included in ACO this daemon action may break the distributed nature of ACO. For a majority of daemon actions to be applied a knowledge of the global pheromone structure is required and this does not lend itself to a distributed history representation (as is the case with the biological inspiration). More than this, unless the problem can be broken down into multiple sub-problems for distribution amongst multiple ACO threads, the algorithm will not be able to be distributed, since any ant requires the entire pheromone mapping at any time in order to construct a complete feasible solution drawing on the most up-to-date pheromone information. This is not to say that ACO cannot be modified to be distributed, but that the ACO framework as described in the literature is not inherently distributed.

5 Conclusions

This report has illustrated some of the simple biological ant colony behavioural models and their powerful resultant emergent properties. The report then went on to show how these models were adapted for search and optimisation applications in the field of biologically inspired computation. Finally a framework for a subset of these ant-inspired search algorithms was highlighted and comments about the framework offered.

There is no doubt that there will be more algorithms developed that are based on some (or many) aspects of ant colonies. In fact before the very first ant algorithm was published two prominent biologists (Hölldobler and Wilson, 1990) commented:

> The neglect of ants in science and natural history is a short-coming that should be remedied, for they represent the culmination of insect evolution, in the same sense that human beings represent the summit of vertebrate evolution.

Even though the quote was directed at the field of ecology it could be said that it is only in the past 10 years that the computational intelligence field has begun to tap into the power that ant-inspired search has to offer. With further research the artificial ants should no doubt find their niche in computational intelligence as they have in the real world.

References

[1] D. Angus and J. Brownlee. Discrete history ant systems. Submitted to ANTS2006, Fifth International Workshop on Ant Colony Optimization and Swarm Intelligence, September 2006.

[2] G. Bilchev and I. C. Parmee. The ant colony metaphor for searching continuous design spaces. In T. C. Fogarty, editor, *Proceedings of the AISB Workshop on Evolutionary Computation*, Evolutionary Computing, pages 25–39. Springer-Verlag, April 3-4 1995.

[3] O. Cordon, F. Herrera, and T. Stützle. A review of the ant colony optimization metaheuristic: Basis, models and new trends. *Mathware & Soft Computing*, 9(2,3), 2002.

[4] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3(2):159–168, March 1990.

[5] C. Detrain, J.-L. Deneubourg, and J. Pasteels. Decision-making in foraging by social insects. In C. Detrain, J.-L. Deneubourg, and J. Pasteels, editors, *Information Processing in Social Insects*, chapter 4, pages 331–354. Birkhäuser Verlag, 1999.

[6] M. Dorigo, E. Bonabeau, and G. Theraulaz. Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16:851–871, 2000.

[7] M. Dorigo and G. Di Caro. Ant colony optimization: a new meta-heuristic. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC99)*, pages 1470–1477. IEEE, 1999.

[8] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: An autocatalytic optimizing process. Technical report, Dipartmento di Electronica e Informazione, 1991.

[9] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 26(1):29–41, Feb 1996.

[10] M. Dorigo and T. Stützle. The ant colony optimization metaheuristic: Algorithms, applications and advances. Technical report, IRIDIA, 2000.

[11] L. M. Gambardella, E. D. Taillard, and M. Dorigo. Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society*, 50(2):167–176, February 1999.

[12] J. M. Pasteels, J.-L. Deneubourg, and S. Goss. Self-organization mechanisms in ant societies (1) : Trail recruitment to newly discovered food sources. In J. M. Pasteels and J.-L. Deneubourg, editors, *From Individual to Collective Behaviour in Social Insects*. Birkhäuser Verlag, 1987.

[13] G. Reinelt. Tsplib95, 1995. Available at: `http://www.iwr.uni-heidelberg.de/groups/comopt/software/tsplib95`.

[14] R. Wehner. Spatial organization of foraging behaviour in individually searching desert ants, cataglyphis (sahara desert) and ocymyrmex (namib desert). In J. M. Pasteels and J.-L. Deneubourg, editors, *From Individual to Collective Behaviour in Social Insects*. Birkhäuser Verlag, 1987.

