Analysis of Ant Colony Optimization

Clay McLeod

University of Mississippi

May 1, 2015

1 Introduction

Ant Colony Optimization (ACO) is Monte Carlo algorithm based on the behavior of ants to solve global minimum problems. The algorithm, first proposed by Marco Dorigo in his thesis [3], was developed to find the shortest path in a graph. Since the original idea was proposed, several variants of the algorithm have been proposed to broaden the range of applications. Surprisingly, the literature surrounding ACO has been dominated by it's original author, Dorigo, even as recent as 2010 [5].

The basis of ant colony optimization lies in the natural concept of stigmergy [6]. *Stigmergy* is a distributed communication paradigm observed in nature (and specifically, ant colonies) where workers indirectly communicate by modifying their environment. In this way, the inspects are "stimulated by the results they have achieved" [9]. In the case of the ant colony, stigmergy is achieved when ants deposit *pheromone* on the ground as they travel towards a food source. Ants can detect the presence and density of a pheromone trail, which evaporates over time. In this way, ants can sense whether or not an ant has walked along the same path recently. This is significant because ants are attracted to pheromone through a biological instinct - the more pheromone deposited on the ground, the more likely an ant is to follow that pheromone trail. However, if there is no pheromone deposited on the ground, the ant is not attracted to any given direction. Thus, it will randomly choose a path to travel along, where paths with pheromone deposited on them are more likely to be traveled along. This biological behavior is the basis for the ACO algorithm.

2 Algorithm Overview

The simplest form of the ACO algorithm is modeled very closely to the biological system referenced in the previous section. Assume a undirected graph $G \in (V, E)$, where a single node represents the ant colony's nest (N_{NEST}) and another single node represents a food source(N_{FOOD}). There should not be a single edge connecting these two nodes, as the problem would prove trivial. Rather, let there be some connected intermediate elements that contain at least one simple path from N_{NEST} to N_{FOOD}. The pseudocode for a simple implementation of the algorithm is outlined on the following page.

We begin with each ant starting at N_{NEST}. The ant will randomly choose an edge to travel along next, hopefully moving towards the food supply. If there is no pheromone along any of the connected edges, the ant will choose an edge to travel with uniform probability. However, if there is pheromone present at one or more of the connected edges, the probability that the ant will choose that edge to travel will be proportional the amount of pheromone on that edge. After the ant has reached the food source, he will deposit a constant amount of pheromone onto the edges that he traveled, making it increasingly likely that other ants will follow after him on that path. This process is repeated until the reaches the food source, and the entire process will begin again for another ant.

Algorithm 1: Simple ACO algorithm

```
initialization;
while not sufficiently sure of optimal solution do
    for m ants do
        currentPosition = N_NEST;
        while currentPosition != N_FOOD do
            Randomly travel to a connected node, paths with pheromone are more likely to be
            chosen.
            Update currentPosition to our current node.
    Update global pheromone map
```

The secret behind this algorithm is that pheromone dissipates as time goes on. Therefore, after every ant has constructed a path to the food source, we should deduct a constant amount of pheromone from every branch, simulating pheromone dissipation in the environment over time. Paths that are shorter will accumulate a larger amount of pheromone, because they will be traveled more quickly and only a small amount of pheromone will evaporate from them. Longer paths will accumlate less pheromone, because they will take a longer amount of time to evaporate. Because the shorter paths have accumulated more pheromone, they will have a higher probability of being traveled again when the next ant comes along, because the amount of pheromone on that path is higher. This process repeats itself until the ants converge to one shortest path.

There are a number of notable variations to the ACO problem, most of which are variations of how the ants supply pheromone. The simplest approach described in this paper is call the Ant System [3, 8]. In this approach, the pheromone is updated by all m ants after every ant has built a solution. An alternative to this approach is the Ant Colony System [7, 11], which introduces ants also updating the pheromone levels after each step in their path construction. This approach produces mixed results based on the context of the problem. In the MAX-MIN Ant System, only the best ant can place new pheromone levels in the graph, and the amount of minimum and maximum pheromone is bounded [15]. This approach achieves significantly improved results over the two previous variations [4].

3 Applications

Ant colony optimization has several applications concerning global minimum problems, especially problems that are easily expressed using graph theory, as they closely emulate the biological process that was the inspiration for this algorithm. Some applications include, but are not limited to, the traveling salesman problem, vehicle routing, project scheduling, the set covering problem, cardinality trees, the maximum clique problem, Bayesian networks, and protein folding [4]. Specifically, several recent studies found that ACO was a world class algorithm for vehicle routing [13, 12] with significant improvments over traditional problem solving methods, such as the GRASP algorithm [13]. ACO seems to outperform most algorithms for other applications as well, such as Bayesian Networks [1, 2]. Unfortunately, ACO produces good, but not great, results for some high profile graph theory problems, such as the Traveling Salesman Problem [9, 7, 11]. Studies concerning other applications, such as the max clique problem [10], state that ACO is outperformed by other algorithms at this time. In many interesting fields of research, such as protein folding [14], the use of ACO has not been thoroughly studied, but ”holds promise” for future applications.

4 Conclusion

The ant colony optimization solution is widely used algorithm based on a biological communication system ants use to find food. In this paper, a brief history concerning the origins of the ACO algorithm was presented. A simple ACO algorithm was presented, and some variants of the algorithm were briefly discussed. Several applications of the ACO algorithm were listed while highlighting it's performance on some key applications. In general, the algorithm tends to perform well in problems that closely emulate the biological system it was inspired by, with the effectiveness of the algorithm hinging on the design of graph [7].

References

[1] De Campos, L. M., Fernandez-Luna, J. M., Gámez, J. A., and Puerta, J. M. Ant colony optimization for learning bayesian networks. *International Journal of Approximate Reasoning 31*, 3 (2002), 291–311.

[2] De Campos, L. M., Puerta, J., et al. Learning bayesian networks by ant colony optimisation: searching in two different spaces. *Mathware & soft computing 9*, 3 (2008), 251–268.

[3] Dorigo, M. Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy* (1992).

[4] Dorigo, M. Ant colony optimization and swarm intelligence: 5th international workshop, ants 2006, brussels, belgium, september 4-7, 2006, proceedings.

[5] Dorigo, M., and Birattari, M. Ant colony optimization. In *Encyclopedia of Machine Learning*. Springer US, 2010, pp. 36–39.

[6] DORIGO, M., BONABEAU, E., AND THERAULAZ, G. Ant algorithms and stigmergy. *Future Generation Computer Systems 16*, 8 (2000), 851–871.

[7] DORIGO, M., AND GAMBARDELLA, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on 1*, 1 (1997), 53–66.

[8] DORIGO, M., MANIEZZO, V., AND COLORNI, A. Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 26*, 1 (1996), 29–41.

[9] DORIGO, M., AND STÜTZLE, T. The ant colony optimization metaheuristic: Algorithms, applications, and advances. *Handbook of metaheuristics* (2003), 250–285.

[10] FENET, S., AND SOLNON, C. Searching for maximum cliques with ant colony optimization. In *Applications of Evolutionary Computing*. Springer, 2003, pp. 236–245.

[11] GAMBARDELLA, L. M., AND DORIGO, M. Solving symmetric and asymmetric tsps by ant colonies. In *International conference on evolutionary computation* (1996), pp. 622–627.

[12] GAMBARDELLA, L. M., TAILLARD, É., AND AGAZZI, G. Macs-vrptw: A multiple colony system for vehicle routing problems with time windows. In *New ideas in optimization* (1999), Citeseer.

[13] MONTEMANNI, R., GAMBARDELLA, L. M., RIZZOLI, A. E., AND DONATI, A. V. Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization 10*, 4 (2005), 327–343.

[14] SHMYGELSKA, A., AND HOOS, H. H. An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem. *BMC bioinformatics 6*, 1 (2005), 30.

[15] STÜTZLE, T., AND HOOS, H. H. Max–min ant system. *Future generation computer systems 16*, 8 (2000), 889–914.