Understanding the Pheromone System within Ant Colony Optimization

Stephen Gilmour and Mark Dras

Department of Computing,
Macquarie University,
North Ryde, Australia 2109
{gilmour, madras}@ics.mq.edu.au

Abstract. Ant Colony Optimization (ACO) is a collection of metaheuristics inspired by foraging in ant colonies, whose aim is to solve combinatorial optimization problems. We identify some principles behind the metaheuristics' rules; and we show that ensuring their application, as a correction to a published algorithm for the vertex cover problem, leads to a statistically significant improvement in empirical results.

1 Introduction

Ant Colony Optimization (ACO) is a collection of biologically inspired metaheuristics for solving difficult combinatorial optimization problems (COPs); they were applied first to the Traveling Salesman Problem (TSP) [DG97] but since then to many other COPs. A general problem with many heuristic methods is the discovery of good parameter values. One approach is to systematically search the parameter space [DMC96]; another is to analyse the properties of the parameter space. The first strategy is interested in how to find the optimal parameters whereas the second strategy is interested in looking at what range of parameters will allow the system to work effectively. Although research has been done on what are good parameters for ACO algorithms, little research has been done as to how the ACO pheromone system actually works theoretically and what space the parameters can be selected from so that the system works effectively and efficiently. This paper is concerned with starting to explore the Ant Colony System (ACS) pheromone system in this direction. It is difficult to apply ACS to new problems when the variables within the problems do not correspond. Through our work we hope to give an understanding of how the ACS pheromone system should work so that translation between problems is less hit-and-miss.

2 ACS Pheromone System

ACO uses agents modelled on ants to find heuristic solutions to COPs. These agents communicate only indirectly by laying pheromone in the environment (*stigmergy*). The more pheromone a particular part of the environment contains, the more desirable that part of the environment becomes to the ants. This is how the ants find solutions.

The ACS pheromone system consists of two rules; one rule is applied whilst constructing solutions (local pheromone update rule) and the other rule is applied after all ants have finished constructing a solution (global pheromone update rule). These rules have different purposes. The purpose of the local pheromone update rule is to make "the visited edges less and less attractive as they are visited by ants, indirectly favouring the exploration of not yet visited edges. As a consequence, ants tend not to converge to a common path" [BDT99]. The purpose of the global pheromone update rule is to encourage ants "to search for paths in the vicinity of the best tour found so far" [BDT99]. Here, as an example, we look at the local pheromone update rule, which for the TSP is

$$\tau_{ij}(t) \longleftarrow (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_0 \tag{1}$$

where $\tau_{ij}(t)$ is the amount of pheromone on the edge (i,j) at time t; ρ is a parameter governing pheromone decay such that $0 < \rho < 1$; and τ_0 is the initial value of pheromone on all edges. Experimentally, the optimal value for ρ has been found to be 0.1 and a good formulation for τ_0 has been found to be $\tau_0 = \frac{1}{n \cdot L_{nn}}$, where n is the number of nodes in the graph, and L_{nn} is the length of the tour found by a nearest neighbour heuristic.

We now propose some properties for the local pheromone update rule. These are straightforward consequences of the rule definition above. It is possible to do the same for the global pheromone update rule; we give full details of our analysis and general approach in [GD05].

Property 1. *For every graph which contains a solution, since $L_{nn} \geq 1$ and $n \geq 2$, $0 < \tau_0 < 1$.*

Property 2. *For an edge (i,j) at time t, if $0 < \tau_{ij}(t) < 1$ then $0 < ((1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_0) < 1$.*

Property 3. *In the absence of the global pheromone update rule, for every edge (i,j), if $\tau_{ij}(t) > \tau_0$ then $\tau_{ij}(t+1) < \tau_{ij}(t)$.*

Property 4. *In the absence of the global pheromone update rule, for every edge (i,j), if $\tau_{ij}(t) < \tau_0$ then $\tau_{ij}(t+1) > \tau_{ij}(t)$.*

Property 5. *In the absence of the global pheromone update rule, for every edge (i,j), $\lim_{t\to\infty} \tau_{ij}(t) = \tau_0$*

3 Vertex Cover Problem

As one instance for analysis, [LDS04] give an ACS algorithm for the set covering problem. This algorithm fits the properties described here and in more detail in [GD05].

A second instance is the algorithm of Shyu et al [SYL04], an attempt to translate the Ant Colony System metaheuristic to the minimum weight vertex cover problem (MWVC problem). In the MWVC problem we want to find a subset of nodes within a graph such that every edge is covered by a node from this subset of nodes and such that the weight of the subset of nodes is minimised. The difficulty with this translation is that the MWVC problem contains two explicit parameters; the covering of all nodes and the minimisation of weight. This dual focus is not covered within Ant Colony System for the traveling salesman problem as it is only explicitly concerned with minimisation of weight. The forming of a cycle that visits every node occurs implicitly.

Because of this, Shyu et al have had to alter the traditional ACS rules so that they work for the MWVC problem; this has included slight alteration of the ACS pheromone rules. For example, in their new rule for the local pheromone update rule τ_0 is given as $\tau_0 = \frac{n(n-a)}{C}$, where n is the number of nodes within the graph; a is the number of nodes found in an initial approximation; and C is the total weight of the initial approximation.

However, their choices violate our proposed properties. The effect of these violations is that the local pheromone update rule will not encourage the shuffling of solutions to find new and better solutions and will encourage getting stuck in local minimums. Further, the amount of pheromone the global pheromone update rule is depositing will be significantly less than what the local pheromone update rule is depositing; this will cause the current best solution not to be reinforced but the pheromone will become lost among the noise from the local pheromone update rule. To correct this, as one example, we reformulate τ_0 as $\tau_0 = \frac{1}{n \cdot \sum_{j \in V'_{nn}} w(j)}$, where n is the number of nodes in the graph; $w(j)$ is the weight of node j; and V'_{nn} is the solution generated using a simple greedy algorithm.

4 Evaluation

We compared an implementation of the Shyu et al (SYL) algorithm with our own variant (GD). We ran both algorithms five times each on five different graphs, each with 400 nodes and 600 edges. We then repeated this with graphs containing 400 nodes and 800 edges. The weights on every edge were one. Each algorithm ran for 15 minutes. Results are in table 1.

As can be seen from these tables, our algorithm improved upon or equalled Shyu et al for every graph and in every run except one, where the difference was one node. We performed a paired t-test on the 25 pairs generated for the graphs with 600 edges and for the graphs with 800 edges; differences are statistically significant for both graph sizes ($p < 0.0001$).

5 Conclusion

The Ant Colony System metaheuristic is a useful approach for getting optimal or near-optimal solutions to difficult optimization problems. In this paper we have

Graphs	1		2		3		4		5	
Algorithms	SYLH	GD	SYLH	GD	SYLH	GD	SYLH	GD	SYLH	GD
Runs										
1	215	212	212	208	215	214	211	210	211	210
2	209	208	213	210	213	212	214	215	212	210
3	211	211	220	213	216	215	214	208	212	211
4	209	205	208	201	210	207	218	215	203	200
5	211	203	211	209	213	212	210	209	213	212
Average	211	207.8	212.8	208.2	213.4	212	213.4	211.4	210.2	208.6
(SYL-GD)	3.2		4.6		1.4		2		1.6	

Graphs	1		2		3		4		5	
Algorithms	SYLH	GD	SYLH	GD	SYLH	GD	SYLH	GD	SYLH	GD
Runs										
1	225	222	229	223	226	220	230	226	228	225
2	221	218	228	227	227	224	229	223	229	227
3	227	225	227	217	223	219	220	220	228	227
4	226	225	223	222	232	224	221	220	227	224
5	228	223	229	223	220	219	232	226	227	223
Average	225.4	222.6	227.2	222.4	225.6	221.2	226.4	223	227.8	225.2
(SYL-GD)	2.8		4.8		4.4		3.4		2.6	

Table 1. Comparison on graphs with 400 nodes and 600 edges (top) and 400 nodes and 800 edges (bottom).

attempted to show that investigating the properties of the parameters in the pheromone system for ACS, in addition to searching the parameter space, is important. Our empirical results show that a statistically significant improvement on a previous algorithm for the vertex cover problem is thereby possible.

References

[BDT99] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence From Natural to Artificial Systems*. A volume in the Santa Fe Institute studies in the science of complexity. Oxford University Press, 1999.

[DG97] Marco Dorigo and Luca Maria Gambardella. Ant colonies for the traveling salesman problem. *BioSystems*, 43:73–81, 1997.

[DMC96] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(1):29–41, 1996.

[GD05] Stephen Gilmour and Mark Dras. Understanding the Pheromone System within Ant Colony Optimmization. MS. Macquarie University, September 2005.

[LDS04] Lucas Lessing, Irina Dumitrescu, and Thomas Stützle. A Comparison between ACO Algorithms for the Set Covering Problem. *ANTS 2004, LNCS 3172*, pages 1–12, 2004.

[SYL04] Shyong Jian Shyu, Peng-Yeng Yin, and Bertrand M. T. Lin. An Ant Colony Optimization Algorithm for the Minimum Weight Vertex Cover Problem. *Annals of Operational Research*, 131:283–304, 2004.