Evaluation of Ant Colony Optimization Parameters in Path-Searching Behavior

Gabriel Telles Missailidis
Instituto Tecnológico de Aeronáutica
São José dos Campos, Brazil
gabriel.missailidis@ga.ita.br

***Abstract*—This paper investigates the impact of key parameters in Ant Colony Optimization (ACO), such as the pheromone importance factor, heuristic importance factor, pheromone decay rate, and the number of ants. By simulating different configurations of these parameters on a standard graph, we examine the algorithm's convergence speed and efficiency in finding the shortest path. Our study shows that specific parameter settings significantly influence the algorithm's performance and convergence speed.**

***Index Terms*—Ant Colony Optimization, metaheuristics, parameter tuning, discrete optimization, convergence analysis**

I. Introduction

Many metaheuristics are inspired by nature. Notable examples include genetic algorithms [1], evolution strategies [2] and particle swarm optimization [3]. In this paper, we will explore ant colony optimization (ACO) [4], which is based on the real behavior of ants in an ant colony [5].

A core concept is that of *stigmergy*, which is a mechanism of indirect communication between agents via the environment in which they live [6]. For example, real-life ants deposit chemicals, called *pheromones*, on the ground when foraging for food and materials, which leads to a higher chance that more ants will follow this same path in the future. Biologists have demonstrated that various behaviors observed at the colony level in social insects can be accounted for by relatively straightforward models that rely solely on stigmergic communication [7]. The goal of ACO algorithms is to coordinate societies of artificial ants by utilizing a form of artificial stigmergy. These algorithms target discrete optimization problems, and have been used to solve problems such as Traveling Salesman Problem and Data Network Routing [4].

A. Path-Searching Behavior of Ants

The stage for ACO is a graph with its associated edge costs. There is a source and a destination nodes.

The probability that an ant will go from node i to node j depends on the pheromone amount at that edge τ_{ij} and a heuristic η_{ij} (often the inverse of the cost), in the form of

$$P_{ij} = \frac{\tau_{ij}^{\alpha}\ \eta_{ij}^{\beta}}{\sum_{k \in \mathcal{N}_i} \tau_{ik}^{\alpha}\ \eta_{ik}^{\beta}} \tag{1}$$

where $\mathcal{N}_i$ is the neighborhood of node i, and α and β are the importances of pheromone and heuristics, respectively. Research has shown that real-life ant behavior is accurately described by $\alpha = 2$ [8].

B. Pheromone Update Mechanism

Pheromone update is composed of two parts. First, there is deposit of pheromone on edges traversed by ants. In our model, ants deposit pheromones only when returning from the destination, so that stray ants do not influence the paths of others. The amount of pheromone added is often proportional to the quality of the path (*e.g.*, inversely proportional to its cost).

Second, there is uniform evaporation of pheromone on all edges. Pheromone evaporation serves as an exploration mechanism that prevents rapid convergence of all ants toward a suboptimal path. By reducing pheromone intensity, it encourages the exploration of diverse paths throughout the entire search process, instead of a exploitation of the same solutions found before.

An update rule for pheromone at edge (i, j) including both effects can be written as

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^{M} \Delta\tau_{ij}^{k} \tag{2}$$

where $\rho \in [0, 1)$ is the *pheromone decay rate* and $\Delta\tau_{ij}^{k}$ is the amount of pheromone deposited on edge (i, j) by ant k, in a setup with M total ants.

II. Methods and techniques

A. Algorithm implementation

We implemented ACO in Python. The interface consists on creating an AntColonyOptimization object, passing the desired parameters on the constructor, and then calling the run() method, which simulates the ant colony, and returns a list with all path lengths found by ants. Simple modifications could be made to return the actual paths instead. The parameters necessary for running the simulation are:

- A graph represented as an adjacency list
- A dictionary containing edge costs
- Source and destination nodes within the graph

Optionally, one can provide the number of ants, the number of iterations, the α and β parameters and a pheromone decay rate ρ. However, the algorithm is able to run on default values for these parameters.

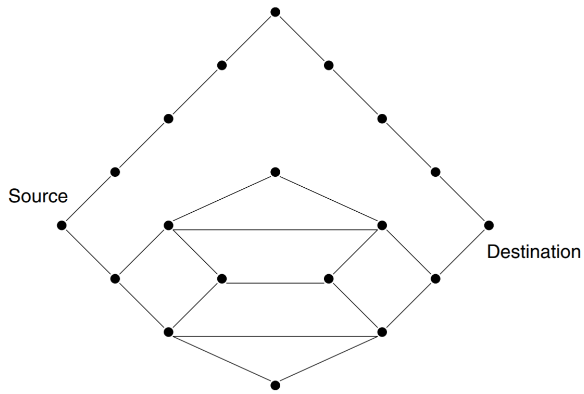

Fig. 1. Graph used in the experiment. All edges have unit cost.

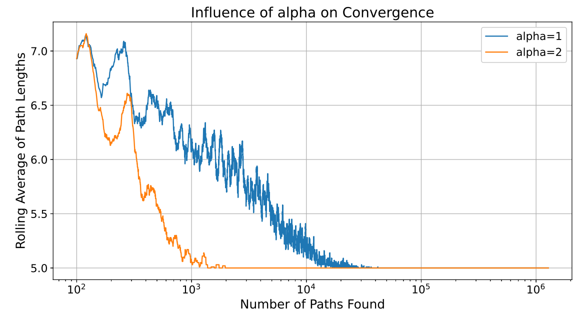

Fig. 2. Influence of α in rolling average of path length.

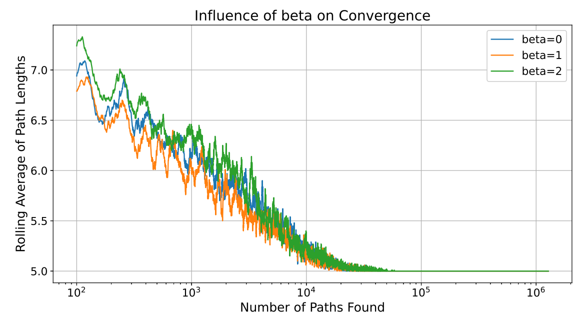

Fig. 3. Influence of β in rolling average of path length.

Each ant in the colony has a memory capable of storing its path, the total path length, and whether it is returning to the source or not.

The main loop of run() executes for the specified number of iterations. At each step, each ant moves once. The movement of an ant can be either forward or backward. In forward movement, the ant chooses a random node from its neighborhood according to Equation 1. By default, it excludes the node it just came from, unless it is the only possible node to visit, such as when the ant reaches a dead end in the graph. Loops are allowed during this phase. While moving forward, ants do not deposit pheromone.

When an ant reaches its destination, it performs two key actions. First, it reviews its path and removes any loops. Then, it calculates the total length of its path as the sum of the edge costs. Each ant then moves back one node along its cycle-free path, depositing pheromone on the traversed edge. The amount of pheromone deposited is the inverse of the ant's total path length. The rationale is that the ant has a single unit of pheromone to deposit on its return path from the destination, so it is distributed uniformly along the path.

Upon reaching the source node, the ant reports its path length and is reset, starting another walk in the next iteration. At the end of each iteration, pheromone decay is applied to all edges.

B. Algorithm evaluation

All simulations were conducted on the graph shown in Figure 1, extracted from [4]. For simplicity, all edges have unit cost, although the simulation can handle non-uniform costs. Note that the shortest path from source to destination has a length of 5, while the top path has length 8. However, the bottom approach has the possibility of more intricate paths, which could include loops. Therefore, even though at the start ants have a 50% probability of choosig the top or the bottom path, we expect the initial average path length to be closer to 8 than to 5, since not all ants choosing the bottom path will follow the shortest path.

A total of 12 simulations were conducted, each with varying parameter choices. Each simulation consisted of 100,000 iterations, running on an Intel i7-9750H with 16GB of RAM and a GTX 1660Ti graphics card. The total runtime was about one hour.

First, the impact of the α value was studied to determine if $\alpha = 1$ (simple probability proportional to pheromone amount) or $\alpha = 2$ (based on real ant behavior [8]) was the better choice. The other parameters were kept constant at $\beta = 1$, $\rho = 1\%$, and $M = 128$.

Second, the importance of the heuristic was evaluated via its exponent β. We tested $\beta = 0$ (no heuristic applied), $\beta = 1$, and $\beta = 2$ for the same reasons as α. For these tests, the standard parameters were $\alpha = 1$, $\rho = 1\%$, and $M = 128$.

Third, the pheromone decay rate was varied as $\rho = 0$ (no pheromone decay), $\rho = 1\%$ (low evaporation), and $\rho = 10\%$ (high evaporation). For this suite, the parameters were $\alpha = 1$, $\beta = 1$, and $M = 128$.

Finally, the number of ants M was varied, doubling between 64 and 512, totaling four simulations. The default parameters were $\alpha = 1$, $\beta = 1$, and $\rho = 1\%$.

III. Results and Discussion

According to Figure 2, for both values of α, the rolling average of path lengths found by the ants started at about 7, which confirmed our hypothesis that the initial path length would be closer to 8 than to 5. We also note both averages converge to 5, which is indeed the shortes path length possible on the test graph. However, it is clear that $\alpha = 2$ converged way quickly. Note the logarithm scale on the horizontal axis: it converged with about 10 times fewer iterations than $\alpha = 1$.

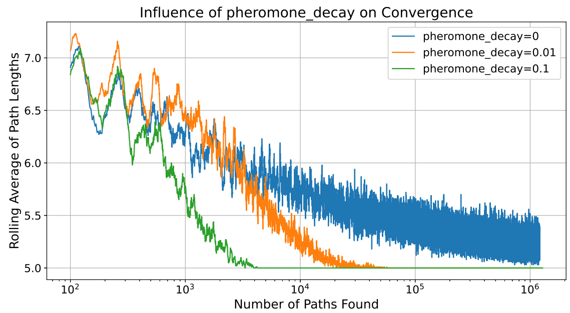

Fig. 4. Influence of pheromone decay rate ρ in rolling average of path length.

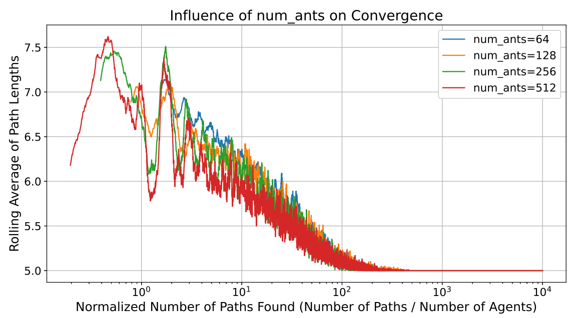

Fig. 5. Influence of number of ants M in rolling average of path length.

Testing the simulation for different values of β, however, did not significantly alter the results, as shown in Figure 3. If the simulation stops before convergence, it appears that $\beta = 1$ gives the smallest average path length, but all three values tested converged at similar points, around 60,000 paths found. Remember, this run had 128 ants, so it averaged about 470 trips per ant for convergence.

Changing the pheromone decay rate ρ had substantial impacts, as seen in Figure 4. Without pheromone evaporation ($\rho = 0$), the simulation did not converge even after 100,000 iterations and more than one million paths traveled. The average path length oscillated between 5 and 5.5. It can also be seen that high evaporation rates, around 10% per iteration, yielded better results in the form of faster convergence, at about 5,000 paths found, while low evaporation rates only reached convergence after 60,000 paths.

Finally, we can see the effect of the number of ants on the simulation convergence, as shown in Figure 5. For this plot, we scaled the x axis based on the number of ants, since naturally more ants would yield more paths. Remarkably, the algorithm's convergence remains consistent across this range of ant population sizes. However, employing more ants for early stopping does lead to a marginal decrease in average path length.

IV. Conclusion

This paper investigated the impact of pheromone importance factor α, heuristic importance factor β and pheromone decay rate ρ in the performance of an ant colony optimization algorithm for path-searching. The value $\alpha = 2$ was selected based on research in real ants' behavior [8], and our simulation showed it is indeed a better choice than $\alpha = 1$, the naïve approach, would suggest. Tests showed that $\beta = 1$ and $\rho = 10\%$ provided the best convergence speed and shortest average path lengths.

As future work, tests on other types of graphs should be conducted, such as graphs with non-uniform edge costs or a larger number of nodes. Another possible future study is evaluating the effect of the number of ants in the convergence speed.

References

[1] Melanie Mitchell, An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press, 1996. ISBN 9780585030944.

[2] Thomas Bäck and Hans-Paul Schwefel, "An Overview of Evolutionary Algorithms for Parameter Optimization," Evolutionary Computation, vol. 1, no. 1, pp. 1–23, 1993. doi: 10.1162/evco.1993.1.1.1. ISSN 1063-6560.

[3] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in Proceedings of IEEE International Conference on Neural Networks, vol. IV, 1995, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[4] Marco Dorigo and Thomas Stützle, Ant Colony Optimization. Cambridge, MA: MIT Press, 2004. ISBN 0-262-04219-3.

[5] M. Birattari, P. Pellegrini, and M. Dorigo, "On the Invariance of Ant Colony Optimization," IEEE Transactions on Evolutionary Computation, vol. 11, no. 6, pp. 732–742, Dec. 2007. doi: 10.1109/TEVC.2007.892762. ISSN 1941-0026. S2CID 1591891.

[6] G. Theraulaz, "A Brief History of Stigmergy," Artificial Life, vol. 5, no. 2, pp. 97–116, Spring 1999. doi: 10.1162/106454699568700. PMID 10633572. S2CID 27679536.

[7] O. Feinerman and A. Korman, "Individual versus collective cognition in social insects," Journal of Experimental Biology, vol. 220, no. 1, pp. 73-82, Jan. 2017.

[8] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels, "The self-organizing exploratory pattern of the Argentine ant," Journal of Insect Behavior, vol. 3, pp. 159–168, 1990.