## Ant Colony Optimization (ACO) algorithm

Pierre-Paul Grassé, the French entomologist in the 40s and 50s of the 20th century, shone a light on some interesting findings observed in some species of termites. He observed the reactions of these termites to something he called “significant stimuli” and found that those reactions themselves can also operate as significant stimuli for other insects in the colony, including the insect that produced them. This special type of communication found in these species was termed *stigmergy* and it was described with two main characteristics (Dorigo et al., 2006; Salman et al., 2020):

* It is an indirect, non-symbolic form of communication using the environment as a medium (i.e., communication through modification of the environment).
* The stigmergic information created is local (i.e., can only be accessed when in the vicinity/locus in which it was released).

Since then, stigmergy has been observed in many other species including ant colonies. In ant species, as the members travel in search of, or returning from a food source, they deposit a chemical along the trails they traverse called *pheromone*. Other ants, upon inspecting a trail, can perceive these pheromones and, as a response, tend to follow the trail containing higher pheromone levels. As they traverse their chosen path, they also add their own deposited pheromone trail to the path, further increasing its pheromone concentration and the ‘attractiveness’ of this trail to successive ants on arrival. The remarkable efficacy of this exploratory pattern was demonstrated by the thorough investigation performed by Deneubourg et al. (1990). In their, soon to be well known, “double bridge experiment”, they introduced a diamond-shaped bridge between the ant nest and a chemically unmarked arena for the ants to explore. This provided the ants with a binary left/right choice in such a way that the “dynamics of their cumulative choice [could] be easily quantified”. They noted that the ant’s stigmergic system exploited the positive feedback loop such that it, “after initial fluctuation, rapidly leads to one of the two forks becoming more or less completely preferred to the other” and eventually the whole colony converges on the use of only one of the bridges.

Diagram

Description automatically generatedDiagram

Description automatically generated

Figure 2 - Different length branches

Figure 1 - Original Double Bridge experiment

Goss et al.(1989) expanded on this study by adding a food source to the arena and differing the size of the two bridges in the experiment. Again, the axis of entry is the same 30° on both sides of the bridges (total 60° between bridges) to minimize ‘loop back’ ant journeys and so that the forager has no preference for one or branch or the other based on position. In their experiment, at first, the ants were choosing equally between the short and long but “abruptly, some minutes later, one branch becomes visibly preferred”. The ants, at first choosing stochastically, the shortest bridge were the first to reach the nest, so on return the probability that they take again take the shorter path is higher as there is no pheromone trail attracting them to the longer path yet until those on that path finally arrive. Their choice then reinforces that pheromone trail as they deposit more on the way back, positively affecting bias towards this trail for all successive ants (Blum & Li, 2008). Their proposed model for that observed behaviour became the main source of inspiration for developing the Ant Colony Optimization algorithm we know today (Dorigo et al., 2006).

Dorigo & Blum (2005) defined the framework of the basic ACO as an iterative method through which the ants “probabilistically construct solutions to the combinatorial optimization problem under consideration, exploiting a given pheromone model”. The population of ants are set to traverse a graph, each ant building a solution by walking along the vertices in an iterative process to find the optimal route through the graph. In the algorithm, ants select the next vertex to visit using a stochastic mechanism that, like its natural counterpart, is biased towards the pheromones that have been left on that vertex. Finally, at the end of each iteration, some of the solutions generated are used for performing a pheromone update on the routes traversed.

Over time, ACO has become one of the most popular biologically inspired algorithms in literature (Blum & Li, 2008) and has been used to solve many graph-based or graph adapted combinatorial optimization problems. It has found applications in areas like feature selection using a rough set approach (Chen et al., 2010), heart disease prediction and classification (Khourdifi & Bahaj, 2019), scheduling problems (Deng et al., 2019), and real-time routing problems (Samà et al., 2016). In fact, work has also been done to increase the applications of the algorithm to include problems based on continuous domains (Socha & Dorigo, 2008).

## Defining the problem domain

Not much work was needed to adapt the ACO technique to fit the TSP because it was designed to tackle this sort of graph-based problem in the first place. The algorithm needs the problem to be represented as a list of edges for the ants to traverse and deposit pheromone on. For this, two matrixes were implemented, one to store the distances between cities and another for the pheromones. The row and column indexes for both tables represent the ‘to’ and ‘from’ cities that the edge occurs on, and the matrix cells represent the edges containing the information that the matrix stores for that edge. The TSP used in this project was an undirected graph of city vertices allowing edge connections between any two cities. So, any matrix storing information for this study would be symmetric, where either the row or the column may be used to represent the ‘from’ or ‘to’ cities for each edge on the graph (as long as the row and column are not both used to represent the same choice).

Calendar

Description automatically generated

Figure 3 - Example Matrix for a map with 5 cities storing the distances of each edge

As the algorithm runs, each ant’s task is to construct solutions () to the TSP. In this study, the is a sequence of cities stored in the order they should be visited and at the end of each iteration of solution building, the pheromone table will be updated based on these solutions. The score of any solution presented in the algorithm is the inverse of the total sum of distances traversed in that route:

where means the distance between cities and , is a set recording the distances between each edge in the proposed solution sequence (which can each be attained from the distance matrix), including condition2 of the TSP: the loop back to start. The round trip back to the starting city also needed is a method to calculate the scores of individual edges to be used as an assignable heuristic value for the edge .

### Initialization

The pheromone and distance matrix sizes should be set to , where is thenumber of cities on the map. The pheromone matrix should be initialized with trivially small but non-zero values to retain calculational integrity (no divide by zeros) and the values in the distance table should be calculated based on the distances between the vertexes on the map. In the algorithm, the ants developed would travel the graph using the information stored in these matrixes for guidance and updating the values in the pheromone matrix after their solution is completed.

### Ants

For each iteration of the algorithm, the set of *m* ants initially start with empty solutions and are each given a random starting city , from which, to begin building their solutions . For each step in constructing a solution for that iteration, the ant chooses the next valid city to visit , and appends it to its current solution. represents the list of valid cities that can be chosen based on the TSP condition1 (no repeats) and it is a subset of the complete list of cities on the map *C*. If there are no more cities that can be visited , then the ant’s solution is complete and ready to be assessed.

#### Ant System (AS)

There are a few ways in which the ant’s choice in cities can be influenced but the most widely used rule, taking its inspiration from the mathematical model proposed by Goss et al.(1989), is that of the first proposed algorithm model: the *Ant System* (Dorigo et al., 2006; Dorigo & Stützle, 2019):

is the ant’s current city position and the city listed at the end of the current solution set , and is a valid city choice to visit. is the pheromone level on edge () and it can be obtained from the pheromone matrix. The parameters and are weights that control the importance of pheromone vs heuristic information when making a city choice. They should be tuned according to the nature and requirements of the study that the algorithm is applied. Through experiments done, it was found that for and respectively was the optimum ratio for this study.

Because the goal of the algorithm is to optimize towards the shortest path, when an ant completes its solution to the algorithm, the score of its proposed solution becomes the power of the pheromone trail it leaves , and that is added the pheromone levels, stored in the pheromone matrix, for all edges the ant passed through when building its solution . Finally, in keeping true to the natural inspiration, another value , denoting the evaporation rate of pheromone over time, is used at the end of each iteration to simulate pheromone evaporation across the graph .

#### Max-Min Ant System (MMAS)

The MMAS aims to be an improvement over the original AS. In MMAS, for each iteration, only the best ant is allowed to update the pheromone table with its solution trail and all pheromone updates done in the algorithm are bound between a min and max value. The best ant is the one presenting the highest scoring solution compared to the rest of the population. The pheromone update and evaporation calculation for the graph are as follows:

Where and respectively are the upper and lower bounds imposed on the pheromones and the operator is defined as:

And is the for the best performing ant in the population. Stützle & Hoos (2000) proposed some analytical considerations that can be used as a base for choosing appropriate values for and like ….

#### Ant Colony System (ACS)

What makes the ACS unique is that it aims to perform the pheromone update after each construction step where each ant applies its pheromone to the last edge traversed. This allows all ants to dynamically affect one another’s decisions as they all attempt to construct solutions for that iteration of the algorithm. However, because the metric that was chosen for determining the pheromone level that each ant deposits on its travelled path are the scores of their proposed complete solutions, and it is not possible to calculate that value until the solution construction process is complete, it was decided not to use this variant in this study. However, the details for this technique are described in further detail in the paper by Dorigo et al. (2006).

(Dorigo & Gambardella, 1997; Gambardella & Dorigo, 1996; Stützle & Hoos, 2000) also justified the settings Alhpa = 1, beta = 2, p = 0.9

## Bibliography

Blum, C., & Li, X. (2008). Swarm Intelligence in Optimization. In C. Blum & D. Merkle (Eds.), *Swarm Intelligence: Introduction and Applications* (pp. 43–85). Springer. https://doi.org/10.1007/978-3-540-74089-6\_2

Chen, Y., Miao, D., & Wang, R. (2010). A rough set approach to feature selection based on ant colony optimization. *Pattern Recognition Letters*, *31*(3), 226–233. https://doi.org/10.1016/j.patrec.2009.10.013

Deneubourg, J.-L., Aron, S., Goss, S., & Pasteels, J. M. (1990). The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, *3*(2), 159–168. https://doi.org/10.1007/BF01417909

Deng, W., Xu, J., & Zhao, H. (2019). An Improved Ant Colony Optimization Algorithm Based on Hybrid Strategies for Scheduling Problem. *IEEE Access*, *7*, 20281–20292. https://doi.org/10.1109/ACCESS.2019.2897580

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, *1*(4), 28–39. https://doi.org/10.1109/MCI.2006.329691

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, *344*(2–3), 243–278. https://doi.org/10.1016/j.tcs.2005.05.020

Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. *Biosystems*, *43*(2), 73–81. https://doi.org/10.1016/S0303-2647(97)01708-5

Dorigo, M., & Stützle, T. (2019). Ant Colony Optimization: Overview and Recent Advances. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics* (pp. 311–351). Springer International Publishing. https://doi.org/10.1007/978-3-319-91086-4\_10

Gambardella, L. M., & Dorigo, M. (1996). Solving symmetric and asymmetric TSPs by ant colonies. *Proceedings of IEEE International Conference on Evolutionary Computation*, 622–627. https://doi.org/10.1109/ICEC.1996.542672

Goss, S., Aron, S., Deneubourg, J. L., & Pasteels, J. M. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, *76*(12), 579–581. https://doi.org/10.1007/BF00462870

Khourdifi, Y., & Bahaj, M. (2019). Heart Disease Prediction and Classification Using Machine Learning Algorithms Optimized by Particle Swarm Optimization and Ant Colony Optimization. *International Journal of Intelligent Engineering and Systems*, *12*. https://doi.org/10.22266/ijies2019.0228.24

Salman, M., Garzón Ramos, D., Hasselmann, K., & Birattari, M. (2020). Phormica: Photochromic Pheromone Release and Detection System for Stigmergic Coordination in Robot Swarms. *Frontiers in Robotics and AI*, *7*. https://www.frontiersin.org/article/10.3389/frobt.2020.591402

Samà, M., Pellegrini, P., D’Ariano, A., Rodriguez, J., & Pacciarelli, D. (2016). Ant colony optimization for the real-time train routing selection problem. *Transportation Research Part B: Methodological*, *85*, 89–108. https://doi.org/10.1016/j.trb.2016.01.005

Socha, K., & Dorigo, M. (2008). Ant colony optimization for continuous domains. *European Journal of Operational Research*, *185*(3), 1155–1173. https://doi.org/10.1016/j.ejor.2006.06.046

Stützle, T., & Hoos, H. H. (2000). MAX–MIN Ant System. *Future Generation Computer Systems*, *16*(8), 889–914. https://doi.org/10.1016/S0167-739X(00)00043-1