# Ant Colony Optimization Algorithm

Olive Cheong Yu Xuan
*School of Computing*
*Asia Pacific University of Technology & Innovation (APU)*
Kuala Lumpur, Malaysia
tp070087@mail.apu.edu.my

Liew Kar Yan
*School of Computing*
*Asia Pacific Univeristy of Technology & Innovation (APU)*
Kuala Lumpur, Malaysia
tp068541@mail.apu.edu.my

Loh Xue Li
*School of Computing*
*Asia Pacific University of Technology & Innovation (APU)*
Kuala Lumpur, Malaysia.
tp064478@mail.apu.edu.my

Chu Kwan Yeu Jovi James
*School of Computing*
*Asia Pacific University of Technology & Innovation (APU)*
Kuala Lumpur, Malaysia
tp069489@mail.apu.edu.my

Ko Chin Zheng
*School of Computing*
*Asia Pacific University of Technology & Innovation (APU)*
Kuala Lumpur, Malaysia
tp061927@mail.apu.edu.my

Zailan Arabee bin Abdul Salam
*School of Computing*
*Asia Pacific University of Technology & Innovation (APU)*
Kuala Lumpur, Malaysia
zailan@apu.edu.my

***Abstract*— ACO is a potent algorithm inspired by ants' social behavior. It can solve complex computational problems and offers a promising approach to finding near-optimal solutions in various domains. This essay focuses on ACO's potential to solve the Travelling Salesman Problem (TSP) by mimicking ants' pheromone-based path finding. To examine how different ACO algorithm parameters, such as pheromone power, ant speed, and distance power, affect the algorithm's performance. We aim to uncover optimal parameter ranges that improve the ACO algorithm's efficiency and efficacy in solving TSP situations through thorough experimentation and research. Our results reveal the complex balancing act between exploration and exploitation inside the ACO algorithm, offering insightful information for both scholarly work and real-world applications.**

***Keywords*— *ant colony optimization, travelling salesman problem, algorithm***

## I. INTRODUCTION

Ant Colony Optimization is currently being used in different areas in our everyday life without us even knowing the difference. Ant Colony Optimization is currently being used to solve our day-to-day issues. Moreover, it possesses a lot of beneficial factors that could bring convenience to everyone from different walks of life and as well as being one of the many existing algorithms that currently exist in our modern era. Despite it being an algorithm that was conceived around the 1990s, it is still widely used for a variety of applications. To bring to light more about the algorithm itself, it was first discovered by Marco Darigo through the inspiration from the observation of ants behavioural movements. Ants in general are eusocial insects that are commonly seen surviving together as large community as to continue to prosper. Moreover, their communication style is based on the usage of sound, touch, and pheromone. (GeeksforGeeks, 2020b). Within this documentation, we will be discussing similar projects that had been done using this algorithm, a myriad of approaches deployed by their respective authors, Materials, Software, Hardware, the implementation of said algorithm and so on. Once everything has been gone through, the reader will become more knowledgeable regarding Ant Colony Optimization.

## II. LITERATURE REVIEW

### *A. Similar Project*

There have been a few studies that have been conducted previously on the topic discussed within this paper. The conclusions gathered from a review of these paper are able to provide a clearer understanding of how to alter the settings or the parameters of the Ant Colony Optimization (ACO) algorithm to tackle the Travelling Salesman Problem in a more efficient way.

It was found that traditional has made great breakthroughs in past projects that involved solving the Travelling Salesman Problem but there are some problems that still remained such as long searching times and the way it jumps into local optimal solution easily. To overcome this, Jun-Man and Zhang (2012) came up with 2 improvements to implement in the traditional ACO called the IVRS Algorithm that involves novel optimized implementing approach and also the introduction of individual variation which involves enabling ants to have different route strategies. Through this, the impact of pheromones can be enhanced.

Hlaing and Khine (2011) discussed regarding ACO being a computational bottleneck that accounts for long convergence time and the local optima trap. From this, the study highlighted adopting improvements such as candidate set strategy to improve the convergence speed as well as implemented a dynamic updating rule for the heuristic parameter based on entropy. The system proved to be more effective.

Research work done by Elloumi et al. (2014) involves a hybrid method where a Particle Swarm Optimization was introduced initially but was modified by the ACO algorithm to improve its performance. Conventional PSO has a satisfactory algorithm when it comes to smaller problems with moderate dimensions and search space however it depletes as the complexity increases. However, the implementation of both methods has been proven to be efficient, as validated by the TSP benchmarks.

### *B. Methodology / Approach*

This section calls for an overview of previous research and studies regarding the problem-solving methods of the Ant Colony Optimization. These methods have proven to have helped industries and organizations to solve the Travelling

Salesman Problem, hence, will be used as a guideline to improve this paper's work.

Research done by Toksari (2016) explains how ACO is used in tackling scheduling problems specifically a real-world instance of the University Course Timetabling Problem (UCTP). The approach used in this research is the Max-Min Ant system where the pheromone works in a way where the trail value is updated at the end of each iteration by best-so-far ant or the iteration-best ant. This differs from the common ACO approach where here, the pheromone is defined individually according to each event. This is to improve the searching process by using the quality of components to determine the gain value of the pheromone for each edge. From this, the system becomes manageable, and memory efficient.

According to (Ray Han et al., 2023), ACO was used as a method to solve issues such as the quadratic assignment problem (QAP) where the algorithm was mainly expected to distribute the load towards all optical satellites evenly. Similarly, the journal also touched on resource allocation problem (RAP) and several common NP-hard issues where an even distribution of resources was needed to be executed by the algorithm that can potentially help reduce cost and maximize profit. In addition to that, the last approach is a hybrid approach that involved ACO and a local search algorithm where an ACO-based load balancing routing and wavelength assignment method that is to be associated with providing fair load balancing for whole optical satellite network. Through these, ACO has been proven to be competitive with other algorithms within its category.

Furthermore, (Qian Chao et al. (2022) discussed further on other implementations of ACO, namely the reliability of a wireless sensor network where ACO helped to provide a more cost-efficient way of deploying WSN by searching the shortest inference route as well as ensuring that the bandwidth and the energy consumption is used in the most efficient way possible. From this, a 20% increase is experienced in terms of quality of solutions in solving these problems as compared to other algorithms. Furthermore, ACO was also used in a satellite broadcasting scheduling (SBS) problem where the pheromones play a role in making the management of alternative signal routes of the satellite easier. Lastly, the approach of ACO also helped solve robot path planning where the algorithm was expected to navigate the robot with the shortest route possible while avoiding obstacles along the way.

### C. *Conclusions / Recommendations*

The ACO algorithm is a famous method for finding the best solution in computational problems. It has several applications such as solving TSP problems, permutations, and combinations in mathematics, managing and allocating resources in business, and studying biological phenomena like protein folding and gene sequencing.

In this paper, the authors aim to address a problem by using the ACO algorithm to find the optimal path for traversing all cities in the United States. To accomplish this objective, the authors will conduct multiple rounds of tests on various variables, such as ant speed, distance power, and pheromone power, to determine the best range of values. The results of these tests will be presented in part V along with the corresponding data and findings.

## III. MATERIAL AND METHODS

### A. *Ant Colony Optimization Algorithm*

The Ant Colony Optimization algorithm has been applied to the Travelling Salesman Problem. The Ant Colony Optimization algorithm used in this research is a Python-based implementation that employs vectorized NumPy arrays for efficient computation. The source code of the algorithms does not create by researchers. The source code used in this paper belong to James Mcguigan from Kaggle. The researchers have adapted the source code to modify parameters showing the effect of the result. The tuneable parameters include *ant_count, ant_speed, distance_power, pheromone_power, decay_power, reward_power, best_path_smell, start_smell*, and *stop_factor*.

### B. *Software Requirement*

- **Python 3.x**: The language of implementation.
- **NumPy Library**: For numerical computations and vectorization.
- **Kaggle Environment**: The original source code was optimized and tested within a Kaggle notebook.

### C. *Hardware Requirement*

The algorithm has been optimized for and tested on Kaggle kernels, require standard personal computer with a multi-core processor for parallel computation.

## IV. ALGORITHM IMPLEMENTATION

This section provides an overview of the application of the Ant Colony Optimization (ACO) method to the Travelling Salesman Problem (TSP). We examine its objectives, key parameters, and performance measures before evaluating it as an ACO method solution for TSP.

### A. *Purpose*

The primary objective is Ant Colony Optimization (ACO) to offer a solution to the Travelling Salesman Problem (TSP). TSP is an NP-hard problem, and ACO is a probabilistic technique for solving computational problems by simulating the behaviour of ants searching for a path from their colony to a food source.

### B. *Parameters*

The Ant Colony Optimization Algorithm (ACO) used to address the Travelling Salesman Problem (TSP) is accompanied by a wide range of adjustable parameters, each of which uniquely influence on the performance and results of the algorithm.

The parameter *ant_count* specifies the overall quantity of ants in the NumPy array, influencing the algorithm's efficiency and the rate at which the pheromone trails are established. The *ant_speed* parameter affects the ant's step count each epoch, resulting in a performance optimisation that reduces the required number of epochs.

The algorithm uses two key powers, *distance_power* and *pheromone_power,* to influence an ant's decision-making. While *distance_power* affects the ant's preference for closer nodes, *pheromone_power* controls how sensitive ants are to small differences in pheromone levels. Nevertheless, *decay_power* is a parameter affecting the rate at which pheromones on the trail decay over time. On the other hand, *reward_power* impacts the pheromone reward given by each ant upon completing its round trip, which is relative to the path cost of the route it took.

When an ant finds a new best path, *best_path_smell* is a multiplier used by the queen to double the pheromones along this path, thus increasing the probability of other ants exploring this path. The *stop_factor* is a control parameter that sets how many times the ants will redouble their efforts after finding a new best path before stopping. The *start_smell* parameter affects the initial number of pheromones on the map, influencing the balance between optimization and exploration during the algorithm's early stages.

While the algorithm is not guaranteed to find the optimal path, it is designed to quickly converge on a near-optimal solution. Parameters here are not explicitly defined, but the algorithm performs within certain statistical bounds, offering a level of reliability. Given the numerous hyperparameters like *ant_count, ant_speed, reward_power,* and so on, their optimal values can be determined through a process of hyperparameter optimization. This involves running multiple iterations of the algorithm with different parameter combinations to find the set that yields the best performance.

## V. Results and Discussion

### A. Discussion on Implementation

#### a) Pheromone Power

The power of pheromones lies in how noticeable the differences are. The data for this project will be tested within the range of -2 to 10. To ensure that other variables don't affect the data, they have been set to their default values. For instance, the distance power value will be set to 2. Not only that, to obtain more accurate data, each pheromone power value will be looped three times, and an average will be taken.

#### b) Ant Speed

The ant speed is influencing the number of steps taken per epoch. The data range will undergo testing within the interval of 10, with a default value of +1 being employed to mitigate the potential influence of other variables on the data. For example, the starting value of the distance power is 1, and then by giving the previous result to a power of 2. Furthermore, in order to obtain more precise data, the process will be repeated three times, and the resulting average value will be rounded to the nearest whole number. This final value will be displayed below the table.

#### c) Distance Power

Distance power is the ability of ants to choose a next node rather than just blindly follow the pheromone trail. The tested data set the distance power from -1.0, 0.0, 0.5, 1.0, 2.0, 5.0, 10.0. It will be run through 3 iterations to get a more accurate and get the average number and all result will be converted to whole number to have a better understanding which shown below the table.

### B. Results

Based on the results in Fig. 1 above, it can be observed that increasing the pheromones power initially leads to a decrease in the average path. However, beyond a certain point, i.e., 2, this trend is reversed, and the average path starts to increase, indicating deoptimization. Although the relationship between the pheromone power and the average path distance does not show a positive correlation, the conclusion drawn from this study is that the optimal range for the pheromone power is between 0.5 and 2. Within this range, the path distance will be optimized.

TABLE I. RESULT OF THE EXPERIMENTS ON PHEROMONE POWER

| Distance Power | Pheromone Power | Test Result | Average of Path |
|---|---|---|---|
| 2 | -2 | 3190 | 3262 |
| | | 3208 | |
| | | 3389 | |
| | -1 | 3013 | 3080 |
| | | 3212 | |
| | | 3016 | |
| | 0 | 2843 | 2812 |
| | | 2876 | |
| | | 2718 | |
| | 0.5 | 2200 | 2211 |
| | | 2191 | |
| | | 2241 | |
| | 1 | 2232 | 2220 |
| | | 2225 | |
| | | 2202 | |
| | 1.5 | 2240 | 2245 |
| | | 2257 | |
| | | 2239 | |
| | 2 | 2304 | 2275 |
| | | 2296 | |
| | | 2226 | |
| | 3 | 2319 | 2386 |
| | | 2476 | |
| | | 2363 | |
| | 5 | 2680 | 2548 |
| | | 2348 | |
| | | 2617 | |
| | 10 | 2477 | 2658 |
| | | 2927 | |
| | | 2569 | |

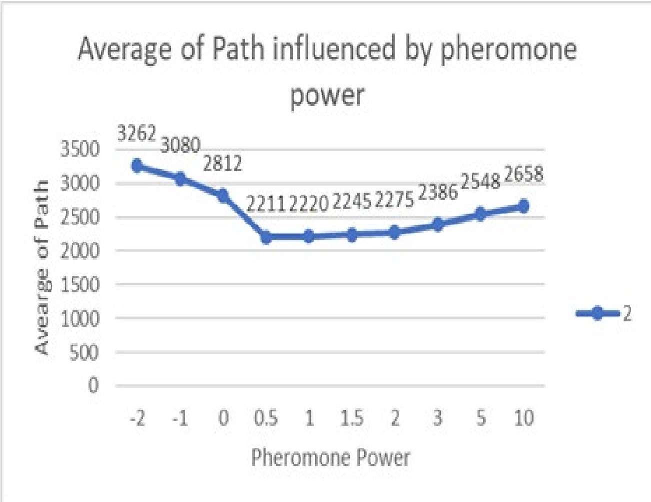


Fig. 1. Pheromone power against path

*a) Ant Speed*

TABLE II. RESULT OF THE EXPERIMENTS ON ANT SPEED

| Ant Speed | Test Result | Average of Paths |
|---|---|---|
| 1 | 2254 | 2290 |
| | 2282 | |
| | 2333 | |
| 2 | 2277 | 2249 |
| | 2238 | |
| | 2232 | |
| 4 | 2287 | 2340 |
| | 2411 | |
| | 2321 | |
| 8 | 2288 | 2259 |
| | 2240 | |
| | 2250 | |
| 16 | 2248 | 2226 |
| | 2219 | |
| | 2210 | |
| 32 | 2188 | 2185 |
| | 2173 | |
| | 2195 | |
| 0 | 2225 | 2347 |
| | 2473 | |
| | 2343 | |

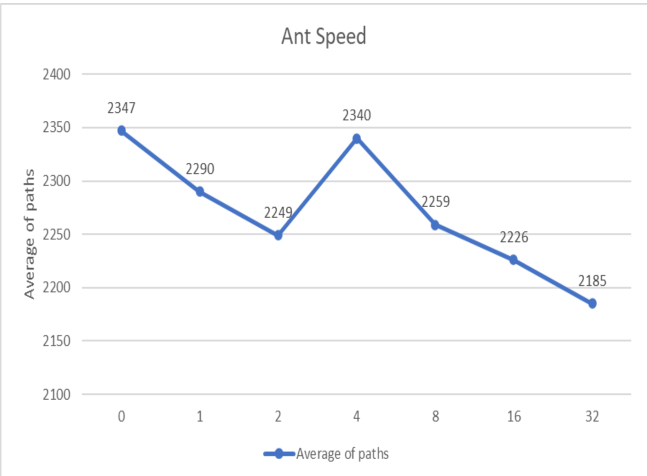


Fig. 2. Ant speed against path

Fig. 2 shows when increasing in ant speed, will increase the number of ants and exploration rate and finding the average shorter paths but there is a fluctuate in the ant speed of 4 due to the randomness in search. More ants speed up, allow the pheromone trails to build up faster on good paths but only to a point. The average path get shorter as the algorithm runs, demonstrating it is finding better solutions.

However, beyond 16 ants the performance stops improving much. This indicates there are diminishing returns as you add more and more ants.

*b) Distance Power*

TABLE III. RESULT OF THE EXPERIMENTS ON DISTANCE POWER

| Distance Power | Test Result | Average |
|---|---|---|
| -1.0 | 7949 | 5390 |
| | 7507 | |
| | 7163 | |
| 0.0 | 3373 | 4477 |
| | 5072 | |
| | 4986 | |
| 0.5 | 2445 | 2486 |
| | 2608 | |
| | 2406 | |
| 1.0 | 2393 | 2303 |
| | 2246 | |
| | 2271 | |
| 2.0 | 2316 | 2269 |
| | 2232 | |
| | 2260 | |
| 5.0 | 2237 | 2220 |
| | 2206 | |
| | 2216 | |
| 10.0 | 2240 | 2245 |
| | 2255 | |
| | 2240 | |

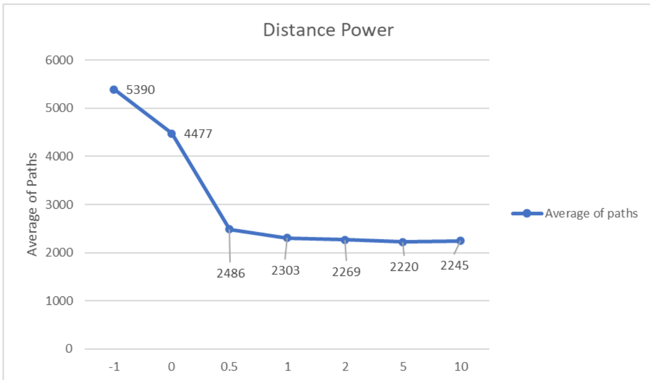


Fig. 3. Distance power against path

Fig.3 shows the result that for negative distance power values (e.g., -1.0), the algorithm tends to favor longer paths which is 5390. While at distance power is between 0 and 1 has a tremendously decreased from 4477 to 2486 because the ants switch from an unbiased random search to a focused optimization of path length. This is why there is a sharp drop in the average path length.

When the distance power =5 is more quickly and able to find the shortest optimal average path length possibly encouraging more exploration. Extremely high distance power values (e.g., 10.0) seem to have little effect on improving the solution further, as ants become too focused on exploiting the shortest paths.

As distance power increases from negative to positive values, the average path distance decreases, indicating that ants are finding better solutions as more emphasis is given to the actual distance in the pheromone update rule.

## VI. CONCLUSION

The Travelling Salesman Problem (TSP) has been addressed using the Ant Colony Optimization (ACO) method, and this research has shown its adaptability and potential in doing so. We have learned a lot about the algorithm's behavior and its capacity to find nearly perfect solutions by investigating the effects of several important parameters.

Regarding the effects of pheromone power, ant speed, and distance power on ACO's performance, our experiments have produced important conclusions. Notably, we found that pheromone power between 0.5 and 2 results in the best path distances, balancing exploitation and exploration. An ideal number of ants should be chosen since ant speed showed a trend of improvement up to a certain point, after which decreasing returns were seen. Additionally, shifting from random search to focused optimization with distance power values between 0 and 1 resulted in lower average path lengths.

These discoveries help to clarify how the ACO algorithm's parameters can be tuned, allowing practitioners to tailor their implementations to issue scenarios. Our research provides an invaluable road map for researchers and practitioners looking to fully realize the promise of ACO, which continues to play a crucial role in addressing computational difficulties in a variety of areas. In the end, the adaptability of ACO, motivated by the teamwork of ants, continues to be a viable approach for solving challenging optimization problems.

## ACKNOWLEDGEMENT

This project would not have been possible without the contributions of many, and we are grateful to all of them. We are also particularly thankful to James for sharing the code and project on Kaggle that facilitated our work. While the purpose of this project is to study how modified parameters in ACO algorithms impact TSP pathfinding.

## REFERENCES

Elloumi, W., Abed, H. E., Abraham, A., & Alimi, A. M. (2014). A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP. Applied Soft Computing, 25, 234–241.

GeeksforGeeks. (2020b). Introduction to Ant Colony Optimization. GeeksforGeeks.

Hlaing, Z. C. S. S., & Khine, M. A. (2011). Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm. International Journal of Information and Education Technology, 404–409.

Jun-Man, K., & Zhang, Y. (2012). Application of an improved ant colony optimization on generalized traveling salesman problem. Energy Procedia, 17, 319–325.

Sallim, J. A Background Study on Ant Colony Optimization Metaheuristic and its Application Principles in Resolving Three Combinatorial Optimization Problems. school of computer sciences, University Sains Malaysia: Marco Dorigo.

Han, T., Foo, G., Yi, C., Yee, B., Zer, K., Wong, J., Jie, Lun, N., Arabee, Z., & Salam, A. (2023). Optimizing ACO Algorithm for the TSP Through Parameter Modification. Journal of Applied Technology and Innovation, 7(3), 52.

Gan, Q., Chao, Aruljodey, S., Ong, C., Xian, Arabee, Z., & Salam, A. (2022). Solution optimization of ACO in TSP by modification of parameters. Journal of Applied Technology and Innovation, 6(1), 58.