Optimizing ACO Algorithm for the TSP Through Parameter Modification

Tan Ray Han
School of Computing
Asia Pacific University of Technology & Innovation (APU)
Kuala Lumpur, Malaysia
TP067355@mail.apu.edu.my

Brayden Yee Kai Zer
School of Computing
Asia Pacific University of Technology & Innovation (APU)
Kuala Lumpur, Malaysia
TP066114@mail.apu.edu.my

Ngan Cheng Lun
School of Computing
Asia Pacific University of Technology & Innovation (APU)
Kuala Lumpur, Malaysia
TP061737@mail.apu.edu.my

Gary Foo Ce Yi
School of Computing
Asia Pacific University of Technology & Innovation (APU)
Kuala Lumpur, Malaysia
TP060880@mail.apu.edu.my

Wong Jun Jie
School of Computing
Asia Pacific University of Technology & Innovation (APU)
Kuala Lumpur, Malaysia
TP067383@mail.apu.edu.my

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

***Abstract*— Ant Colony Optimization (ACO) is an algorithm that is used for optimizes the path or solve optimization problems in particular problems including route problems or Travelling Salesman Problems (TSP). The implementation of the ACO algorithm was applied by using the biological behavior of ants in real life which uses pheromone trails to communicate indirectly, in parameters of the ACO algorithm, some variables will be analyzed by using the behavior of ants in real life such as the number of ants, epoch of ants, distances, and evaluation. In this journal paper, ACO will focus on two parameters which are the evaporation rate and epoch of ants to obtain the nearest and best path for TSP by modifying the parameters in each experiment.**

Keywords—ant colony optimization, travelling salesman problem, algorithm, evaporation rate, epoch of ants

I. INTRODUCTION

The Traveling Salesman Problem (TSP) is a well-known combinatorial optimization problem that has challenged researchers for many years. The problem involves finding the shortest possible route that a salesman can take to visit a set of cities exactly once and return to the starting city. Despite its simple definition, the TSP is a complex problem that is known to be NP-hard, which means that no known polynomial-time algorithm can solve it optimally for all instances.

Ant Colony Optimization (ACO) is a promising metaheuristic algorithm inspired by real ants' behavior in finding the shortest path between their colony and a food source. ACO is a population-based approach that works by simulating the behavior of ants in building pheromone trails that guide their movements toward a food source.

Recently, ACO has gained increasing attention as a powerful tool for solving TSP, particularly for large-scale instances of the problem. The basic idea behind ACO is to create a pheromone matrix that represents the probability of an ant choosing a particular path. As the ants traverse the graph, they deposit pheromones on the edges they visit, and the pheromone concentration on each edge is updated accordingly. The pheromone trail is then used to guide the ants in their subsequent iterations, and the process is repeated until a satisfactory solution is found.

In this research, our group aims to investigate the effect of changing two important parameters in ACO: the evaporation rate and the number of iterations or epochs. These parameters play a critical role in determining the quality of the solution obtained by the algorithm. By varying these parameters and observing their impact on the performance of the algorithm, we hope to gain a better understanding of how ACO works and how it can be optimized for different types of TSP instances.

II. LITERATURE REVIEW

A. Similar Project

A few studies have been conducted on the subject we are discussing in this paper. The conclusions in the relevant publications can provide a better understanding of how to change the Ant Colony Optimization algorithm's settings to obtain a better solution path for the Traveling Salesman Problem.

The original ant colony algorithm lacks appropriate parameter selection, therefore Wei (2014) suggests a four-step procedure to fix the problem. The study outlines an efficient approach for choosing the ideal set of parameters and includes experiments on establishing the parameters m, α, β, ρ, and Q. The performance of the enhanced ant colony algorithms, including the best-worst ant system, max-min ant system, ant-based sorting systems, and optimal retention policy ant system, is compared with that of the same TSP problems. The results of the studies demonstrate how the suggested strategy of parameter combinations significantly accelerates convergence.

A more advanced version of the Ant Colony Optimization (ACO) technique was developed by Yao et al. (2021) to resolve the venerable Traveling Salesman Problem Utilising Particle Swarm Optimization (PSO) (TSP). The algorithm consists of three steps: creating a mathematical model; utilising PSO to solve the optimal path; and increasing the path's pheromone concentration in the ant colony model. As shown in a typical TSP instance, the proposed algorithm performs better in finding the global optimal solution and converges more quickly than ACO and PSO.

The research work of Yang et al. (2008) suggests extending the ant colony optimization approach from TSP to GTSP, a modification of the TSP. To avoid local minima, the suggested approach considers group effect and uses a mutation process and local searching methodology. Numerical outcomes demonstrate the method's efficacy in resolving GTSP issues.

B. Methodology / Approach

This section provides an overview of earlier investigations and studies that concerned the Ant Colony Optimization method of problem-solving. These methods, which have helped the relevant industry to solve the Traveling Salesman Problem, will be used as a guide to help with this paper's work.

Guoli et al. (2017) discuss the ant colony optimization-based load balancing routing and wavelength assignment (RWA) method (ALRWA) to offer fair load balancing throughout the whole optical satellite network. When developing a multi-objective optimization model, the characteristic of the distribution of traffic around the world is considered. The algorithm's objective is to evenly distribute load among all optical satellite networks.

According to research Yin and Wang (2006), the ant colony optimization (ACO) paradigm can be used to solve the resource allocation problem. This paradigm is detailed in the article (RAP). To achieve its objectives, the RAP seeks the optimal distribution of a limited quantity of resources among several tasks. For instance, cost-cutting, or profit-maximizing strategies are constrained by the resources that are accessible. The article discusses several well-known NP-hard issues and how ACO versions can be developed to solve them, including the travelling salesman problem, quadratic assignment problem, scheduling problem, least weight vertex cover challenge, and curve segmentation problem.

The study of Demirel and Toksarı (2006) explains how the ant colony optimization (ACO) algorithm can be used to solve the quadratic assignment problem (QAP). The QAP is a combinatorial optimization problem that seeks the most optimal way to assign n facilities to n locations given the distances between the facilities and the mobility between them. The essay proposes a hybrid approach that combines ACO with a local search algorithm. The local search phase of the technique employs simulated annealing. The search space exploration makes use of the assessment of pheromones that ants leave behind on the ground. The proposed method is tested against current benchmark issues, and the outcomes show that it is competitive with other state-of-the-art algorithms.

C. Conclusion / Recommendations

The ACO algorithm can be used to solve several combinational problems to provide an optimal solution, such as providing equitable load balancing throughout the complete optical satellite networks, allocating resources to maximize the utilities, solving the quadratic assignment problem, etc. There are some mutations of ACO with can improve the performance.

From the root, it can be known that the factor that affects the performance of the algorithm is the parameters, fine-tuning the parameters according to different situation or problems make it the most suitable.

In this paper, the ACO will be used to solve the TSP problems, tune the parameters, and find the best value for solving the problems.

III. Material and Methods

A. Ant Colony Optimization

Ant Colony Optimization (ACO) is a metaheuristic algorithm that is inspired by the behavior of real ants. The algorithm simulates the foraging behavior of ants and uses this to find optimal solutions to complex optimization problems. The algorithm works by creating a set of artificial ants that traverse a problem space, building up solutions as they go. Each ant makes decisions about which path to take based on a combination of heuristic information (alpha) and pheromone trails left by other ants (beta). The algorithm iteratively updates the pheromone trails based on the quality of the solutions found, and over time, the ants converge on an optimal solution. The parameters for the algorithm include the number of ants, the number of iterations (epochs), alpha and beta values, and the evaporation rate for the pheromone trails. These parameters can be tuned to achieve better performance on specific problem domains. The source code of the algorithm is from LazoCoder on GitHub (LazoCoder, 2017).

B. Software Requirement

- Java SE
- NetBeans / Visual Studio Code

Java SE (Standard Edition) is a software requirement for running the Java-based source code. It is a platform for developing and running Java applications on desktops, servers, and embedded devices. Java SE provides a powerful set of APIs (Application Programming Interfaces) for developers to create robust and scalable applications. It includes the Java Runtime Environment (JRE), which is required for running Java programs on a user's computer.

NetBeans and Visual Studio Code are Integrated Development Environments (IDEs) that are commonly used for developing Java applications. They provide a user-friendly interface for writing, testing, and debugging code. NetBeans is a popular open-source IDE that provides a wide range of features, including code completion, debugging, and version control. Visual Studio Code is a lightweight IDE that is gaining popularity among developers because of its extensibility and cross-platform support. Both IDEs have plugins and extensions available to support Java development.

C. Hardware Requirement

There doesn't need any high specification computer to run this code. However, we do recommend having an i5 8th generation (Intel Core I5) processor and 8 GB RAM to run this code. In addition to the processor and RAM requirements, the software may also require a certain amount of disk space for installation and data storage. It is recommended to have at least 500GB of storage available for storing the program files and any associated data. A basic graphics card is also sufficient for running the program, as the software does not require any heavy graphical processing. Overall, the hardware requirements for running this code are relatively modest, and most modern computers should be able to run the software without issue.

IV. ALGORITHM IMPLEMENTATION

The purpose of this research paper is to investigate the Ant Colony Optimization (ACO) algorithm for solving the Traveling Salesman Problem (TSP) and determining the best routing for the algorithm by tuning the parameters. The experimented algorithm was the ACO algorithm from LazoCoder (2017) which is used to solve the TSP instance, bays29.tsp. The bays29 instance is a classic TSP benchmark problem that has been used to test the performance of various TSP algorithms. The instance consists of 29 cities in Bavaria, Germany, and the distances between them are given in a matrix format. The goal of the TSP is to find the shortest possible route that visits all 29 cities exactly once and returns to the starting city (TSPLIB, n.d.). There are a total of five parameters consisting in the ACO algorithm which can be modified to achieve an optimal solution for the algorithm. These parameters are the number of ants, evaporation rate, epoch, alpha (pheromone importance), and beta (heuristic importance).

A. Number of Ants

The number of ants parameter in the ACO (Ant Colony Optimization) algorithm refers to the number of artificial ants used to explore the problem space. Each ant is considered an independent agent that constructs a solution to the problem by probabilistically choosing paths based on the amount of pheromone deposited on each path and the length of the path. The number of ants parameter is an important hyperparameter in the ACO algorithm that can have a significant impact on the performance of the algorithm. Choosing an appropriate value for the number of ants parameter is crucial to ensure that the algorithm explores the problem space effectively and converges to a good solution.

B. Evaporation Rate

The evaporation rate parameter in the ACO algorithm refers to the rate at which the pheromone levels on the paths are reduced over time. In the ACO algorithm, the pheromone levels on the paths are updated after each iteration based on the quality of the solutions constructed by the ants. The evaporation rate parameter determines how quickly the pheromone levels on the paths evaporate over time. A high evaporation rate means that the pheromone levels on the paths reduce more quickly, while a low evaporation rate means that the pheromone levels on the paths reduce more slowly.

C. Epoch

The epoch parameter in the ACO (Ant Colony Optimization) algorithm refers to the number of iterations that the algorithm runs for before stopping. Each iteration involves the movement of the ants through the problem space, updating the pheromone levels on the paths they take, and repeating the process until the stopping criterion is met. The stopping criterion can be based on several factors, such as reaching a certain number of iterations, achieving a certain level of solution quality, or reaching a certain level of convergence. The epoch parameter specifies the maximum number of iterations that the algorithm should run for, regardless of the stopping criterion.

D. Alpha (pheromone importance)

The Alpha parameter in the ACO algorithm refers to the relative importance of the pheromone trails on the decision-making process of the ants. In the ACO algorithm, the ants probabilistically choose paths to construct a solution based on the amount of pheromone deposited on each path and the distance of the path. The Alpha parameter determines the degree to which the ants rely on the pheromone trails to choose a path. A high value of Alpha means that the ants give more weight to the pheromone levels when making their decisions, while a low value of Alpha means that the ants rely more on the distance of the path.

E. Beta (heuristic importance)

The Beta parameter in the ACO algorithm refers to the relative importance of the heuristic information, such as the distance or cost of a path, on the decision-making process of the ants. In the ACO algorithm, the ants probabilistically choose paths to construct a solution based on the amount of pheromone deposited on each path and the heuristic information of the path, such as its distance or cost. The Beta parameter determines the degree to which the ants rely on the heuristic information to choose a path. A high value of Beta means that the ants give more weight to the heuristic information when making their decisions, while a low value of Beta means that the ants rely more on the pheromone levels.

In the experiment of this paper, we selected three of the above parameters and carried out parameter tuning to determine the best parameter values to achieve the most effective approach for solving the TSP. The experimented parameters are the number of ants, evaporation rate and epoch. The evaluation values obtained from the experiment will be recorded and tested multiple times to increase the accuracy of the data collected.

V. RESULT, TABLES AND DISCUSSION

A. Implementation

- Modification of “Number of ants” and “Evaporation rate”

The parameters, “Number of ants” and “Evaporation rate”, are modified, and the result of each combination will be illustrated in graphical form to prove the relationship among the 2 parameters mentioned and the evaluation rate. For each modification, only the two parameters to be tested will be changed, and the rest of the parameters will be in default. In default, number of generations = 100, alpha = 1, and beta =5. The two parameters will be modified in a sequential manner to study how both of the “number of ants” and “evaporation rate” affects the evaluation value. For each value of number of ants, which is 1, 50, 100, and 150, a line graph will be plotted, where x-axis is the “evaporation rate” of 0.1, 0.3, 0.5, 0.7, 0.9, and the y-axis is the path evaluation rate. In other words, the value of both “number of ants” as well as “evaporation rate” will be changed in such a way all possible combinations are formed, and the evaluation value produced by each combination will be recorded in a table as well as visualised in a graph. Before the graph is created, the program is run 5 times for each different parameter combination, and the average of the 5 results will be seen as the final evaluation value.

- Modification of “number of epochs” and “number of ants"

The “number of epochs” and “number of ants", are modified, and the result of each combination will be illustrated

in graphical form to prove the relationship between the 2 parameters mentioned and the evaluation rate. While the other parameters stay constant, the graph will be plotted with each combination of "number of epochs" = 50, 100, 150, 200, 250 and "number of ants" = 1, 50, 100, 150 against the evaluation rate. The same steps will be repeated as stated above.

B. Results

- The parameter modification of the evaporation rate together with the number of ants

TABLE I. RESULT OF THE EVAPORATION RATE AGAINST THE EVALUATION VALUE WITH DIFFERENT NUMBERS OF ANTS

Number of ants \ Evaporation rate		0.1		0.3
	Test1: 11197		Test1: 12313	
	Test2: 10311		Test2: 11312	
	Test3: 10748		Test3: 11969	
	Test4: 10493		Test4: 11236	
1	Test5: 10910	10731.8	Test5: 11471	11660.2
	Test1: 9543		Test1: 9665	
	Test2: 9282		Test2: 9607	
	Test3: 9657		Test3: 9711	
	Test4: 9819		Test4: 9566	
50	Test5: 9505	9561.2	Test5: 9548	9619.4
	Test1: 9384		Test1: 9550	
	Test2: 9620		Test2: 9572	
	Test3: 9603		Test3: 9375	
	Test4: 9292		Test4: 9417	
100	Test5: 9421	9464	Test5: 9525	9487.8
	Test1: 9585		Test1: 9363	
	Test2: 9356		Test2: 9510	
	Test3: 9537		Test3: 9597	
	Test4: 9551		Test4: 9392	
150	Test5: 9512	9508.2	Test5: 9479	9468.2

Number of ants \ Evaporation rate		0.5		0.7		0.9
	Test1: 121		Test1: 114		Test1: 112	
	Test2: 109		Test2: 120		Test2: 104	
	Test3: 109		Test3: 118		Test3: 115	
	Test4: 116		Test4: 103		Test4: 116	
1	Test5: 108	11309.2	Test5: 121	11551.4	Test5: 112	11232.6
	Test1: 932		Test1: 971		Test1: 983	
	Test2: 916		Test2: 933		Test2: 933	
	Test3: 957		Test3: 929		Test3: 947	
	Test4: 972		Test4: 957		Test4: 957	
50	Test5: 960	9477.8	Test5: 958	9503.4	Test5: 923	9489.4
	Test1: 961		Test1: 950		Test1: 934	
	Test2: 951		Test2: 954		Test2: 934	
	Test3: 965		Test3: 944		Test3: 948	
	Test4: 965		Test4: 964		Test4: 936	
100	Test5: 939	9564.6	Test5: 958	9545.8	Test5: 945	9398.6
	Test1: 937		Test1: 952		Test1: 950	
	Test2: 956		Test2: 938		Test2: 962	
	Test3: 950		Test3: 945		Test3: 946	
	Test4: 964		Test4: 952		Test4: 943	
150	Test5: 949	9517.6	Test5: 918	9414.6	Test5: 935	9477.2

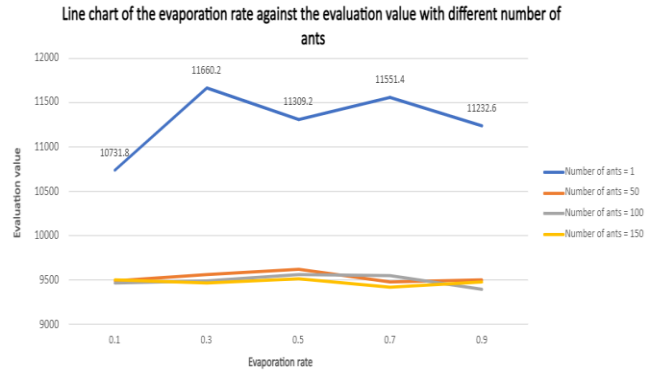

Fig. 1. The evaporation rate against the evaluation value with different numbers of ants

Based on the figure above, there is a significant result in the line "Number of ants = 1". Compared to others, all the evaporation rates in the line "Number of ants = 1" is the highest, and the result shows a clear and obvious gap between the line "Number of ants = 1" and the other lines. In terms of evaporation rate, the line "Number of ants = 1" has the most optimum evaluation rate or best path when the evaporation rate is 0.1, on the other hand, the evaporation value then increases and decreases alternately, showing an inconsistency.

The reason why the lowest evaporation value in the line "Number of ants =1" creates the most optimal evaluation value might be that the only ant can leave the wrong path in a very short time to search for the next destination instead of being stuck in the wrong path and drag time (Abdul Salam et al., 2022). When it comes to the comparison of lines "Number of ants =50", "Number of ants =100", and "Number of ants =150", the difference between each line and the difference of the plots in each line, which is the evaporation rate, is so subtle that it can be summarised that If the number of ants is greater than 50, no matter what is the value of evaporation rate, the evaluation value will be incredibly similar, falling between the range of 9619.4 to 9398.6.

This is because as the number of ants increases, even if ants are stuck in the wrong path due to the high evaporation rate, there are still some ants searching for a better path. The same goes for when the evaporation rate is low, even if the optimal path can be forgotten easily due to the high evaporation rate of pheromones, there is a great chance of the optimal path can be easily found again as the size of the ant colony is large. However, the best result of the three lines can be seen in the line "Number of ants = 100" with the evaporation rate of 0.9.

- The parameter modification of the number of epochs together with the number of ants

TABLE II. EPOCHS AGAINST THE EVALUATION VALUE WITH DIFFERENT NUMBERS OF ANTS

number of ants \ Number of epochs		50		100
	Test1: 11359		Test1:11092	
	Test2: 10926		Test2:10989	
	Test3:10304		Test3:10618	
	Test4:11002		Test4:10828	
1	Test5:11021	10922.4	Test5:10143	10734
	Test1:9622		Test1:9710	
	Test2:10057		Test2:9732	
	Test3:9825		Test3:9739	
	Test4:10119		Test4:9588	
50	Test5:9736	9871.8	Test5:9352	9624.2
	Test1:9741		Test1:9148	
	Test2:9681		Test2:9438	
	Test3:9612		Test3:9462	
	Test4:9777		Test4:9378	
100	Test5:9641	9690.4	Test5:9517	9388.6
	Test1:9511		Test1:9552	
	Test2:9182		Test2:9413	
	Test3:9822		Test3:9415	
	Test4:9634		Test4:9424	
150	Test5:9575	9544.8	Test5:9231	9407

number of ants \ Number of epochs		150		200		250
	Test1:11189		Test1:11162		Test1:10331	
	Test2:11303		Test2:10304		Test2:11292	
	Test3:10954		Test3:10966		Test3:10952	
	Test4:10015		Test4:11035		Test4:10975	
1	Test5:11337	10959.6	Test5:10799	10853.2	Test5:10654	10840.8
	Test1:9234		Test1:9585		Test1:9696	
	Test2:9549		Test2:9449		Test2:9416	
	Test3:9558		Test3:9631		Test3:9497	
	Test4:9446		Test4:9616		Test4:9367	
50	Test5:9520	9461.4	Test5:9546	9565.4	Test5:9639	9523
	Test1:9490		Test1:9319		Test1:9356	
	Test2:9442		Test2:9212		Test2:9269	
	Test3:9592		Test3:9337		Test3:9401	
	Test4:9557		Test4:9169		Test4:9470	
100	Test5:9446	9505.4	Test5:9262	9259.8	Test5:9265	9352.2
	Test1:9212		Test1:9306		Test1:9252	
	Test2:9358		Test2:9419		Test2:9266	
	Test3:9414		Test3:9290		Test3:9232	
	Test4:9449		Test4:9287		Test4:9486	
150	Test5:9398	9366.2	Test5:9295	9319.4	Test5:9426	9332.4

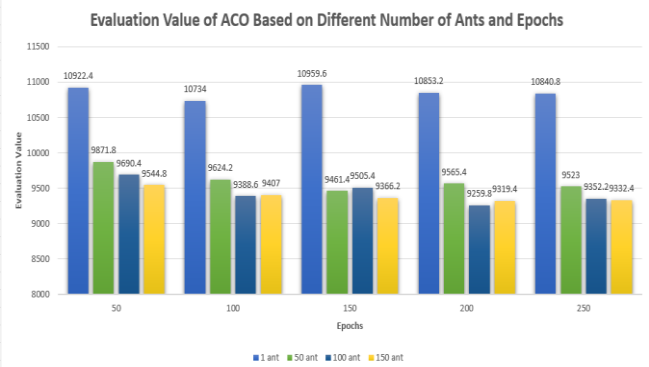

Fig. 2. The evaporation rate against the evaluation value with different numbers of ants

The data above in the spreadsheet represents the evaluation value of Ant Colony Optimization (ACO) based on different numbers of ants and epochs for solving the Travelling Salesman Problem (TSP). From the dataset, we can observe that the number of 1 ant with different epochs is always the highest evaluation in the graph which is not ideal. Furthermore, when it comes to the number of 50 ants, in each different epoch is pretty much the same result of evaluation between 9461.4 to 9871.8, so increasing epochs with the same number of ants doesn't decrease the evaluation. Another observation is that the evaluation value seems to be slightly decreased by increasing the number of ants. For example, when the number of 50 ants and the number of 100 ants in 200 epochs have a significant decrease in evaluation but when it reaches the number of 200 ants, the evaluation rate will still increase.

VI. Conclusion

As a conclusion, "the higher the value of "number of ants", the more optimal the result will be", the study above proves that this statement is not always the case. However, a certain high value of "number of ants" does help with the evaluation rate, but to a certain degree even if the number of ants keeps increasing, the evaluation value will still keep constant; meanwhile, the computational processing time will be doubled. Simultaneously, as the number of ants grows to a specific number, the evaporation rate no longer creates any impactful result in the evaluation value. According to the line graph above, the ants work best in the environment when the number of ants = 100 with the evaporation rate of 0.9 as the most optimal result, 9398.6, is obtained. On the modification of the number of ants and epochs wise, the higher values of both parameters do not lead to better solutions for TSP when using ACO. The same computational situation happens due to hardware constraint when the value of the number of ants and epochs are high. From the experiment, it is observed that 100 ants and 200 epochs will be the optimal solution in this analysis. Ultimately, the overall analysis proves that the best evaluation rate is produced if the number of ants is 100, and the number of epochs = 200 with an evaporation rate of 0.9. However, the experiment results of Yap et al. (2021) also agreed that Fine-tuning the parameters would be crucial for improving the solution's quality over the long term situation requiring a large amount of calculation between cost and path such as transportation sectors.

Acknowledgment

The authors would like to thank to all School of Computing members who involved in this study. This study was conducted for the purpose of optimizing aco algorithm for the tsp through parameter modification Project.

References

Abdul Salam, Z.A., Gan, Q.C., Ong, C.X., & Aruljodey, S. (2022). Solution optimization of ACO in TSP by modification of parameters. *Journal of Applied Technology and Innovation*. Volume6_Issue1_Paper12_2022.pdf (dif7uuh3zqcps.cloudfront.net).

Yap, S.Q., Tian, K. W., See, R., Gan, D., & Abdul Salam, Z.A. (2021). Parameter tuning for artificial Bee Colony algorithm. *Journal of Applied Technology and Innovation*. Volume 5 Issue 1 pp. 11-14

Wei, X. (2014). Parameters analysis for basic ant colony optimization algorithm in TSP. *International Journal of u- and e-Service, Science and Technology*, *7*(4), 159–170. https://doi.org/10.14257/ijunesst.2014.7.4.16

Yao, X.-S., Ou, Y., & Zhou, K.-Q. (2021). TSP solving utilizing improved ant colony algorithm. *Journal of Physics: Conference Series*, *2129*(1), 012026. https://doi.org/10.1088/1742-6596/2129/1/012026

Yang, J., Shi, X., Marchese, M., & Liang, Y. (2008). An ant colony optimization method for generalized TSP problem. *Progress in Natural Science*, *18*(11), 1417–1422. https://doi.org/10.1016/j.pnsc.2008.03.028 4

Guoli, W., Qi, Z., Houtian, W., Qinghua, T., Wei, Z., & Xiangjun, X. (2017). Ant colony optimization based load balancing routing and wavelength assignment for Optical Satellite Networks. *The Journal of China Universities of Posts and Telecommunications*, *24*(5), 77–86.https://doi.org/10.1016/s1005-8885(17)60236-x

Yin, P.-Y., & Wang, J.-Y. (2006). Ant colony optimization for the Nonlinear Resource Allocation Problem. *Applied Mathematics and Computation*, *174*(2),1438–1453. https://doi.org/10.1016/j.amc.2005.05.042.

Demirel, N. Ç., & Toksarı, M. D. (2006). Optimization of the quadratic assignment problem using an ant colony algorithm. *Applied Mathematics and Computation*, *183*(1),427–435. https://doi.org/10.1016/j.amc.2006.05.073 .

LazoCoder. (2017). Ant-Colony-Optimization-for-the- Traveling-Salesman-Problem. Retrieved from Github: https://github.com/LazoCoder/Ant-Colony-Optimization-for-the-Traveling-Salesman-Problem.

TSPLIB. (n.d.). Symmetric Traveling Salesman Problem Instances. Retrieved from TSPLIB: http://comopt.ifi.uni-heidelberg.de/software/TSPL-IB95.