Performance Comparison of SOM and ACO for Travelling Salesman Problem-Case Study on the Indonesia Palapa Ring Network

Ridha M. Negara[1,2]*, Ratna Mayasari[1,2], and Nana R. Syambas[1]

[1] School of Electrical Engineering and Informatics Bandung Institute of Technology, Bandung, Indonesia; Email: nana@stei.itb.ac.id

[2] School of Electrical Engineering Telkom University, Bandung, Indonesia; Email: ridhanegara@telkomuniversity.ac.id; ratnamayasari@telkomuniversity.ac.id

*Correspondence: ridhanegara@telkomuniversity.ac.id.

***Abstract*—Indonesia is an archipelagic country separated by the sea, so it has unique characteristics. Building a telecommunications network that optimally connects Indonesia's islands is essential. Currently, telecommunications networks in Indonesia are connected using the Palapa Ring Indonesia network. Several adjacent islands must specify the closest point to choose a neighbouring route. This paper uses a travelling salesman problem (TSP) that uses the Self-Organizing Map (SOM) and Ant-Colony Optimization (ACO) algorithms to determine the shortest route and optimal computation time. Paper modification has several parameters: number of iterations, neuron size, and space. The measurement results show that SOM is better at measuring the shortest route length and computational time than ACO. By setting the number of iterations below 500 and the M value as much as the number of neurons, the route obtained by SOM will be shorter with a shorter time.**

***Keywords*—traveling salesman problem, Ant-Colony optimization, Self-Organizing Map, Indonesia, palapa ring**

I. Introduction

The Traveling Salesman Problem (TSP) is a challenging optimization problem: it consists of finding the shortest possible route across all cities on each map only once and returning to the original city [1]. Although the explanation is simple, it has its difficulties. If there is a rapid increase in cities, we do not know the general solution to solve the problem.

In the case of TSP, no single general method can solve the problem efficiently. Hence, we need an algorithm that can reduce checking all available combinations so that the search for solutions does not increase exponentially and deterministic algorithms are no longer practical. Therefore, several methods have been proposed to solve these problems: the neural network approach. Neural networks are expected to solve search problems in large spaces on a probabilistic basis. Neural Network represents a system inspired by the nervous system of the human brain (neurons) in classifying data [2–4]. However, the artificial neural modelling of neural networks is much simpler than real neurons. Many NN (Neural Network) models have been proposed based on supervised and unsupervised learning. Neural networks have several advantages, including fault tolerance, automated adaptive learning, parallel information processing, and real-time operations.

The nature of self-organizing or map preservation topology using unsupervised learning is a source of inspiration in solving TSP discrete optimization problems in an ample search space; a group of neurons will carry out the adaptation and learning process independently. Gradually follow the data input pattern properties, thus forming a mapping that represents the shortest route solution with a total cost close to sub-optimal with a relatively fast time. In addition to the self-organizing method, another candidate solution uses a more appropriate probabilistic approach, namely ACO. ACO is a probabilistic algorithm that provides an optimal solution to the TSP problem [5–10]. Studies have employed memetic ACO algorithm is designed to address both symmetric and asymmetric dynamic travelling salesman problems (DTSPs) [5], hybrid Symbiotic Organisms Search (SOS) [6], reduce the adverse effects particularly connected with ACO [7], clustering and simple classification [8], hybrid method of ACO to find best routes and get a better running time [9], Vehicle Routing Problem with Drone (VRPD) [10].

TSP attracts attention and is very easy to describe but difficult to solve. The problem can be expressed as a salesman who travels and visits exactly one city list; the objective function that TSP wants to achieve is to minimize the distance or cost of a set of cities from a salesman's trip. TSP belongs to the NP-hard (Non-deterministic Polynomial-time hardness) category. NP-hard is a problem that has a non-polynomial solution and no polynomial solution. Meanwhile, TSP decision problems are included in the NP-Complete category, namely problems for which polynomial solutions can be found.

Manuscript received August 23, 2022; revised November 2022, accepted January 12, 2023.

Previous works for performance testing of TSP using ACO and SOM [11–14] showed good results on the usual topology. Some studies used the heuristic method to find the minimal value in TSP [11], and another study implemented TSP to create a ring topology with minimum weight or cost [12]. Furthermore, some studies find the minimal route that precisely visits one node from each subcluster [14]. Also, there have been works on bringing graph information into learning models in TSP. Therefore, we propose a study about applying TSP to the Indonesia Palapa Ring network with unique topology characteristics. The authors used the SOM and ACO methods to find the shortest route for the case study. Modifications were made to several parameters, including the number of iterations, neuron size, and space.

II. Methodology

Modifying SOM and ACO to determine the shortest route begins with a search for implementation of SOM and ACO algorithms in the TSP case. Modify the effect of input parameters on the performance and solutions produced by SOM and ACO algorithms, then proceed with modifying parameters that significantly affect the performance of SOM and ACO algorithms and solutions result. These two algorithms are then in the test again and compared. The results testing is analyzed using the posterior approach to determine the performance of the two algorithms.

The Self-Organizing Map (SOM), in the original paper released by Teuvo Kohonen in 1998, described the abbreviation for the technique [15]. As the name implies, the map is self-organizing, represented as a grid of nodes (usually two-dimensional) inspired by neural networks. SOM performs data analysis automatically. The basic concept of SOM is relatively simple, combining competitive learning principles into a specific topological neuron structure so that adjacent neurons have similar weight vectors. SOM architecture is straightforward, consisting of 2 layers: the input and the competitive layers. The input layer is used to receive input signals in the form of vectors.

Meanwhile, the competitive layer is a processing unit generally containing several neurons arranged in one-dimensional linear or two-dimensional topology as a square or hexagonal [16, 17]. The two layers are fully connected; each unit in the input layer is connected to all units in the competitive layer. Each connection has a weight. Units or neurons in SOM are arranged in a regular topology to have clear neighbours. The connection weights on the competition-winning neurons and several neighbouring neurons will be updated based on a specific formula. Learning or training the SOM network begins with determining the initial learning rate, neighbour size, and initialization of weight vectors.

Based on this similarity, the map nodes are arranged spatially more like each other [8, 15]. Therefore, SOM is a great way for pattern visualization and data organization. SOM uses Artificial Neural Network training based on the winner taking all; the concept is that only winning neurons will be updated their weights. SOM uses Artificial Neural Network training based on the winner taking all; the concept is that only winning neurons will be updated their weights. SOM uses a single-layer NN with the number of neuron units equal to a particular group. The data has a relationship with the weight of the processing neuron. Unlike the perceptron, calculating the output signal from the input data with weights on each inner neuron does not use the inner product. Still, Euclidian Distance (Euclidian square) and the output signal do not need to be activated (linear activation function) because the activation function does not affect the selection of the winning neuron and update weights. The user defines the number of groups (neurons), where each group represents the data processed by the index neuron in its layer.

Determination of the initial learning rate η at the time of initialization, expressed as η_0, is generally chosen, which is relatively large in the interval (0,1). The learning rate η decreases during the iteration process, which is expressed by the Eq. (1):

$$\eta = \eta_0 \exp(-t/\tau_\eta) \tag{1}$$

where τ_η < **T** is the number of iterations where the learning rate can still be reduced to 0, and T is the total iteration, with η close to 0 at the end of the iteration, the connection weights are stable.

Determine the neighbours' size at the initialization time expressed by 0; generally, a relatively large number is chosen. The value will decrease during the iteration process, expressed by the Eq. (2).

$$\sigma(t) = \sigma_0 \exp\,(-t/\tau_\sigma) \tag{2}$$

where t is the iteration, T is the number of iterations where the neighbours can still be reduced to size = 1, and T is the total iteration.

The initialization of the weights of the vectors is done randomly, in the form of minimal values in the interval (0,1). Then the training process can be carried out. An input pattern is represented as an input vector x.

$$\boldsymbol{x} = [x_1, x_2, \dots, x_n] \tag{3}$$

A weight vector representing the connection of input units to a unit I in the competitive layer is denoted as

$$\boldsymbol{w} = [w_{i1}, w_{i2}, \dots, w_{in}], \quad i = 1, 2, \dots, m \tag{4}$$

The next step is calculating the distance for each unit in the competitive layer to the value of the input pattern $d_i(x)$ using Euclidean distance.

$$d_i(x) = \sum_{j=1}^{n} (x_j - w_{ij})^2 \tag{5}$$

All neuron units compete to produce one winner, the one with the smallest distance. Next, the connection weights are updated. Updates are made to the winning unit and its neighbours around the winning unit. The update of the weights on the winning unit aims to reduce the distance of the winning unit from the input pattern that activates it. In contrast, the neighbour aims to estimate a similar position to the input pattern.

The steps in conducting this research are presented in Table I. The first steps include understanding the tools and how SOM and ACO work. After completing the first steps, the work gathered information on the parameters by running a simulation in SOM and33 ACO with Indonesia palapa ring networks. Briefly, the procedure train SOM is illustrated as follows [15]:

TABLE I. SOM PSEUDOCODE

```
Procedure Train SOM
    Begin
      Randomize weight for all neurons
      for ((i=1 to iteration_numer) do
      begin
        take one random input pattern
        find the winning neuron
        find neighbors of the winner
        modify synaptic weight of neurons
        reduce learning rate and neighborhood radius
      end
End
```

III. SIMULATION RESULTS AND DISCUSSION

The author evaluates the performance of SOM and ACO in Palapa Ring Topology with a simulation environment. The simulation was created using Colab. Colab is a product from Google Research that uses Python 3.7.14 language.

Each parameter in the SOM and ACO algorithms influences the quality of the solution in the case of TSP. In the case of TSP, no single general method can solve the problem efficiently because it depends on the topology used. This study chose a Case Study on the Indonesia Palapa Ring Network, which has unique characteristics. Oceans separate the islands in Indonesia, so solving the TSP case requires a suitable algorithm.

Therefore, the author conducted several tests of modifications to the parameters of the SOM and ACO algorithms to determine the shortest route and the most efficient time. The author conducts a simulation to determine the effect of changes in these parameters on performance and the resulting solution. Then the test results are analyzed using a posteriori approach to determine the performance of the two algorithms.

A. Self-Organizing Map (SOM) for the Travelling Salesman Problem

SOM solves the TSP problem for the topology of the Indonesian palapa ring network shown in Fig. 1., which after running, is shown in Fig. 2. SOM will find the shortest route and the most efficient time to connect all of them.

In this study, the authors made several changes to the parameters of the SOM algorithm. Changes are made primarily in the parameter M. There are M unit groups arranged in the architecture of the input signals (input) several n.

The weight vector for a unit group is provided from the input patterns associated with the group. During the self-organizing process, the unit group with the weight vector that best fits the input pattern (marked by the minimum Euclidean distance) is selected as the winner. The winning unit and neighbouring units are weighted. Each neuron is connected to other neurons, which are connected by weights. Therefore, the authors perform simulations to see the impact of increasing the parameter M and the number of iterations on the computation time and the shortest distance.

There are two SOM algorithms that the authors tested in this study to overcome the TSP problem, namely the Gaussian Kernel Neighborhood (GKN) and the Elastic Band Neighborhood (EBN) Function. These two algorithms have compared the length results based on the TSP rules on the palapa ring network.

Fig. 3. shows that using the EBN function is better for finding the shortest length based on TSP requirements. EBN provides a path-finding solution that connects each city in Indonesia that is the shortest with a value of M = 15. Therefore, we will use the EBN function compared with the ACO algorithm to solve TSP in the Indonesia Palapa Ring Network in other simulation scenarios.

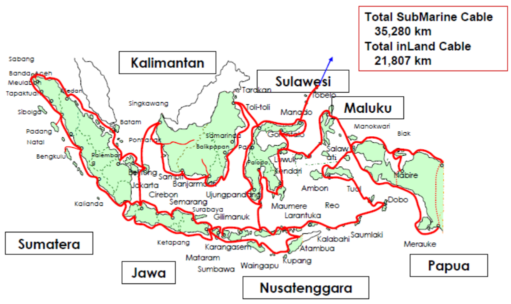

Figure 1. Palapa ring topology. [18]

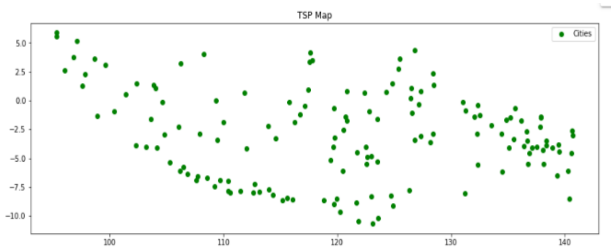

Figure 2. TSP Map – Indonesia palapa ring network

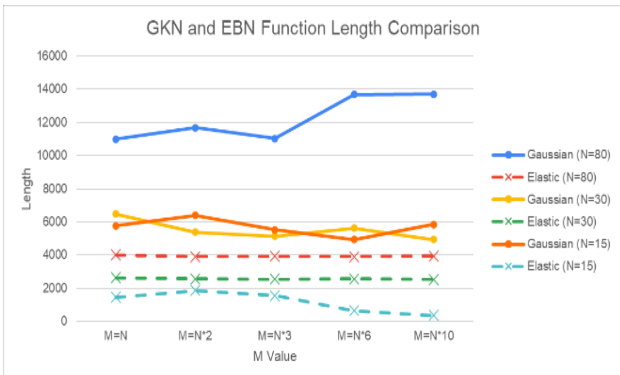

Figure 3. Gaussian Kernel Neighborhood (GKN) dan Elastic Band Neighborhood (EBN) function length comparison

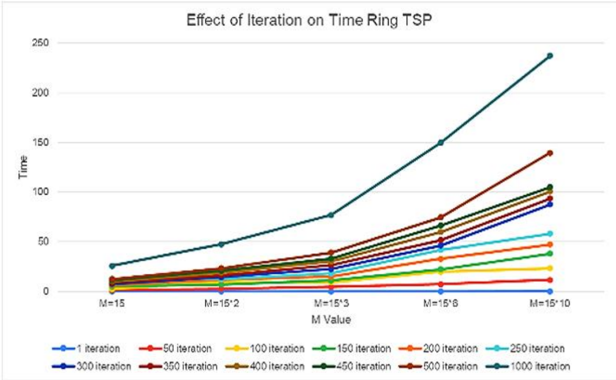

Figure 4. Effect of iteration and M-value in SOM Algorithm on TSP computation time in Indonesia palapa ring network

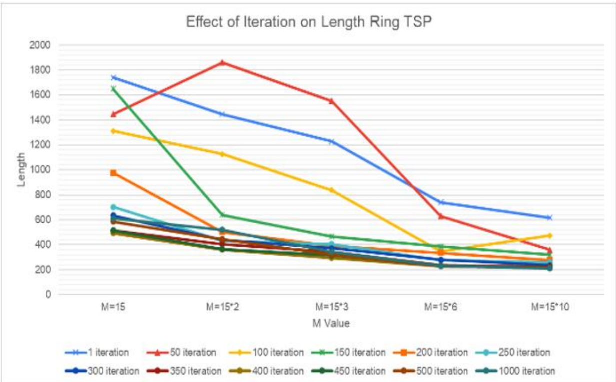

Figure 5. Effect of iteration and M-value in SOM algorithm on TSP length in Indonesia palapa ring network

Fig. 4. shows that changes in the value of M have a significant impact on computing time. Increasing the M value makes a long time to take a TSP solution. However, if we look at the multiples of the M value, it does not match the time needed. When the size of M is increased by 2 to 6 times, the computational time requirement is not significantly increased, but when M is enlarged ten times, the computational time requirement increases sharply. We should avoid this condition because the TSP decision-making time is extended. Meanwhile, let us look at the impact of iteration size on computational time. It is also the same as adding the value of M. Otherwise, the impact of the above scenario on the results of measuring the shortest length using SOM.

Fig. 5. shows that increasing the value of M gives a shorter TSP route. This condition is contrary to the computational time requirements. Therefore, selecting the optimal M value and iteration is needed in every process of solving the TSP problem. In this study, the condition of the Indonesia Palapa Ring network is also a benchmark for selecting optimal parameters in terms of computation time and shortest distance.

B. *ACO for the Travelling Salesman Problem*

Each input parameter in the ACO algorithm affects the measurement results differently. In addition, the topology and the number of nodes used also have a significant impact. The simulation in Fig. 6 uses the ACO algorithm to solve the TSP problem on the Indonesian Palapa Ring network topology.

Each input parameter in an algorithm influences the output of the algorithm. The ACO algorithm has several input parameters: Q, ij, space, M, and n. Each parameter has a different effect. In this simulation, we use the parameters used in previous studies, namely = 1.0, = 1.0, Q = 100, = 0.5, ij = 0.1, space = 15, M and n change according to the scenario. We also observe the impact of iteration on the length and computational time of decision-making.

Fig. 7. shows that the number of iterations does not significantly affect the measurement of the shortest route length. While Fig. 8. shows that the number of iterations above 400 makes the time required to continue to increase exponentially. These results prove that it is necessary to set the number of iterations appropriately to reduce the computation time. However, reducing the number of iterations connected with the number of ants can significantly reduce the resulting solution quality if the problem's size increases.

IV. Evaluations

A. *Performance Comparison of ACO and SOM for TSP*

This study analyzed the two main parameters in the TSP: the length and time required to connect all city points in Indonesia's Palapa ring network once and end up in the same city. This study proves that the ACO algorithm is perfect for providing TSP solutions in a short time, but for the smallest length size, SOM is still better. This tradeoff is the basis for this research experiment which was carried out heuristically.

Fig. 9. shows that iteration impacts the time it takes to make the TSP ring. The increasing M value in SOM makes the TSP ring time longer, while the ACO, which does not have an M value, has no effect. Therefore, SOM must be set with iterations and a low M value to compete with the time obtained by ACO.

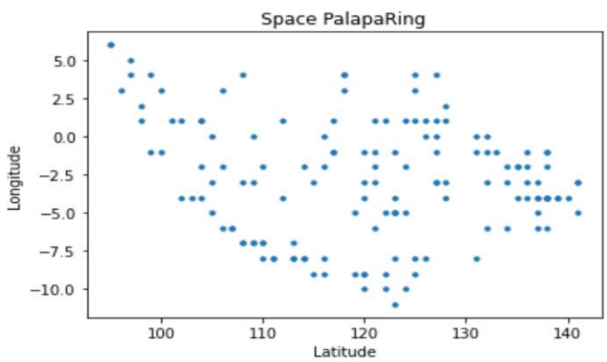

Figure 6. TSP Map – Indonesia palapa ring network

Fig. 10. shows that iteration still affects the distance that connects the TSP ring in the palapa ring network. Interestingly, most of the length obtained from one to 1000 times iteration on SOM gives a shorter length than ACO. This result is evident from the previously discussed tradeoff; SOM provides a better length solution than ACO. We can use the graph below to determine the best number of iterations in the SOM algorithm in solving the TSP ring.

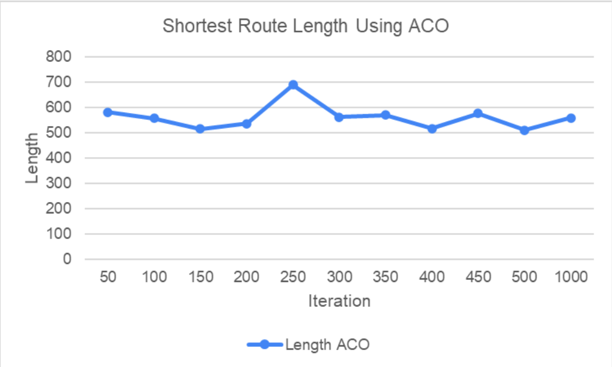

Figure 7. Effect of Iteration using ACO Algorithm on TSP shortest route length in Indonesia palapa ring network

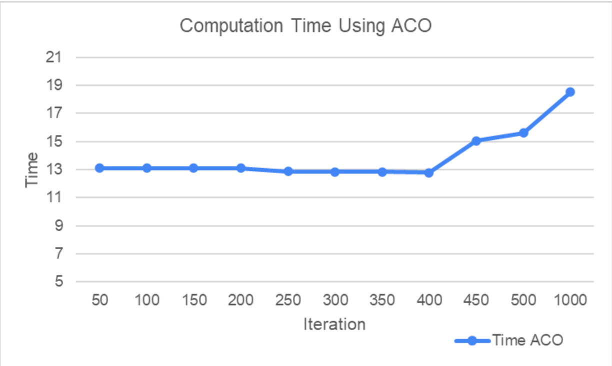

Figure 8. Effect of iteration using ACO algorithm on TSP computation time in Indonesia palapa ring network

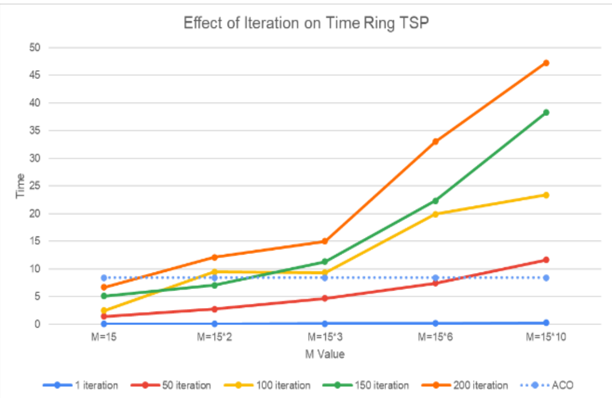

Figure 9. Effect of iteration ACO and SOM on time ring TSP.

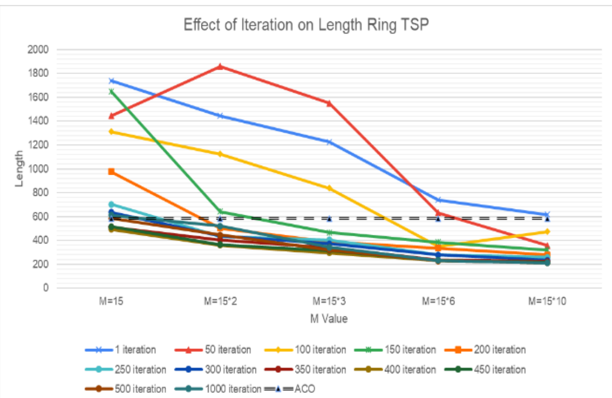

Figure 10. Effect of iteration ACO and SOM on length ring TSP.

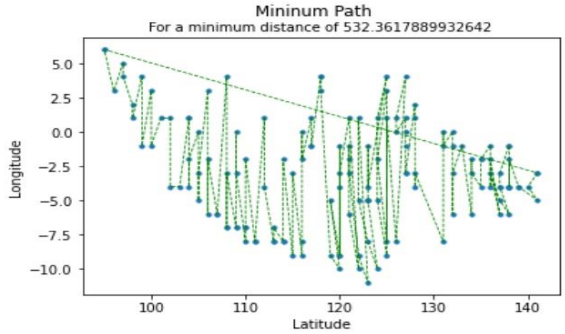

Figure 11. Mapping the palapa ring network using ACO.

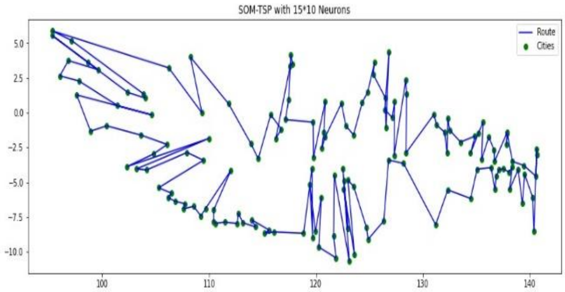

Figure 12. Mapping the palapa ring network using SOM.

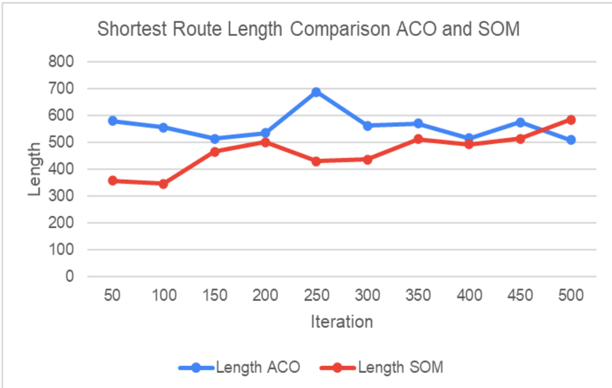

Figure 13. Length performance analysis comparison ACO and SOM

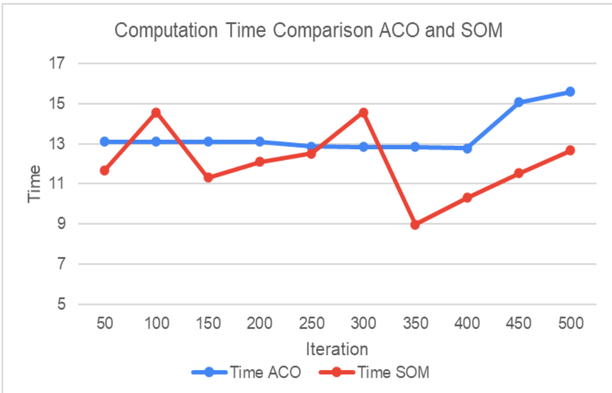

Figure 14. Time performance analysis comparison ACO and SOM

The larger the M value in the SOM, the better the length because a neural network has supported the SOM. This M value can be seen from the mapping process shown by the SOM algorithm, which improves by increasing the M value. This result can be seen in Fig. 11, which maps the palapa ring network using ACO, and Fig. 12, which describes mapping the palapa ring network using SOM.

Based on the comparison of the mappings in Fig. 11 and Fig. 12, we can conclude that using the ACO and SOM algorithms has their respective advantages. However, in this study focused on finding a TSP solution for the Indonesia Palapa Ring Network, the mapping will be an essential value. Indonesia is an archipelagic country separated by sea, so if we look at the mapping provided by ACO and SOM, the most suitable one is SOM.

Fig. 13 further strengthens that SOM is better than the results of measuring the shortest route length compared to ACO. By setting the number of iterations below 500 and M = n, the route obtained by SOM is shorter. In addition, Fig. 14. shows that the computational time required to solve the TSP-SOM problem with iterations below 500 is better than others.

V. Conclusion

This study implemented TSP to optimize the selection of data transmission lines in the Indonesian Palapa Ring network. The Palapa Ring consists of 144 cities spread across several islands. We solve the TSP problem with the shortest path and time using the ACO and SOM algorithms. We modified several parameters, such as the number of neurons, size M, and space, to suit the Palapa Ring topology conditions. If it is not arranged in such a way, the selection of neighbouring routes will be very detrimental to the data delivery process. The experimental results show that each parameter impacts the TSP solution. Nevertheless, this study focused on the number of neurons, M size, and iterations.

The ACO algorithm shows that the number of iterations does not significantly affect the measurement of the shortest route length. However, reducing the number of iterations connected with the number of ants can significantly reduce the resulting solution quality if the problem's size increases. In contrast, the SOM algorithm shows that changes in the value of M have a significant impact on computing time. Increasing the M value makes a long time to take a TSP solution.

The conclusion from all measurements shows that we can solve the TSP problem on the Indonesia Palapa Ring network using the SOM algorithm because it provides the minor shortest route and the time it takes is also faster.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Ridha M. Negara and Ratna Mayasari conducted the research and wrote the paper; Ridha M. Negara analyzed the data; Nana R. Syambas supervised research and review the paper; All authors have read and agreed to the published version of the manuscript.

Acknowledgement

This work is supported by Indonesian Ministry of Research, Technology and Higher Education under the PDD 2022 Research Grant, Research Grant Number 083/E5/PG.02.00.PT/2022.

References

[1] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artif. Intell. Rev.*, vol. 13, no. 2, pp. 129–170, 1999, doi: 10.1023/A:1006529012972.

[2] W. Shao, J. Chan, dan F. D. Salim, "Approximating optimization solutions for the travelling officer problem with neural networks," in *Proc. Int. Jt. Conf. Neural Networks*, 2020, doi: 10.1109/IJCNN48605.2020.9207041.

[3] I. Khan and M. K. Maiti, "A swap sequence based Artificial Bee colony algorithm for traveling salesman problem," *Swarm Evol. Comput.*, vol. 44, pp. 428–438, 2019, doi: 10.1016/j.swevo.2018.05.006.

[4] A. Nammouchi, H. Ghazzai, and Y. Massoud, "A generative graph method to solve the travelling salesman problem," in *Proc. Midwest Symp. Circuits Syst.*, 2020, pp. 89–92, doi: 10.1109/MWSCAS48704.2020.9184505.

[5] M. Mavrovouniotis, F. M. Muller, and S. Yang, "Ant colony optimization with local search for dynamic traveling salesman problems," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1743–1756, 2017, doi: 10.1109/TCYB.2016.2556742.

[6] Y. Wang and Z. Han, "Ant colony optimization for traveling salesman problem based on parameters optimization," *Appl. Soft Comput.*, vol. 107, p. 107439, 2021, doi: 10.1016/j.asoc.2021.107439.

[7] P. Stodola, P. Otřísal, and K. Hasilová, "Adaptive ant colony optimization with node clustering applied to the travelling salesman problem," *Swarm Evol. Comput.*, vol. 70, no. 11, 2022, doi: 10.1016/j.swevo.2022.101056.

[8] T. Mersiovsky, A. Thekkottil, T. Hanne, and R. Dornberger, "Optimal learning rate and neighbourhood radius of Kohonen's self-organizing map for solving the travelling salesman problem," in *Proc. ACM Int. Conf. Proceeding Ser.*, 2018, pp. 54–59, doi: 10.1145/3206185.3206194.

[9] R. W. Dewantoro, P. Sihombing, and Sutarman, "The Combination of Ant Colony Optimization (ACO) and Tabu Search (TS) Algorithm to solve the Traveling Salesman Problem (TSP)," in *Proc. 3rd Int. Conf. Electr. Telecommun. Comput. Eng. ELTICOM 2019 - Proc.*, 2019, pp. 160–164, doi: 10.1109/ELTICOM47379.2019.8943832.

[10] S. H. Huang, Y. H. Huang, C. A. Blazquez, and C. Y. Chen, "Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm," *Adv. Eng. Informatics*, vol. 51, p. 101536, 2022, doi: 10.1016/j.aei.2022.101536.

[11] I. J. Rahman, A. R. Zain, and N. R. Syambas, "A new heuristic method for optical network topology optimization," in *Proc. - ICWT 2016 2nd Int. Conf. Wirel. Telemat*, 2017, pp. 73–77, doi: 10.1109/ICWT.2016.7870855.

[12] B. Supriadi, B. Budi, and N. R. Syambas, "Comparison of proposed algorithm to ant colony optimization algorithm to form a ring topology with minimum path value," in *Proc. 2nd Int. Conf. Wirel. Telemat.*, 2017, pp. 67–72, doi: 10.1109/ICWT.2016.7870854.

[13] N. R. Syambas, S. Salsabila, and G. M. Suranegara, "A heuristic algorithm for travelling salesman problem," in *Proc. 11th International Conference on Telecommunication Systems Services and* Applications, *TSSA 2017*, DOI: 10.1109/TSSA.2017.8272945.

[14] P. Baniasadi, M. Foumani, K. Smith-Miles, and V. Ejov, "A transformation technique for the clustered generalized traveling salesman problem with applications to logistics," *Eur. J. Oper. Res.*, vol. 285, no. 2, hal. 444–457, 2020, doi: 10.1016/j.ejor.2020.01.053.

[15] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1–3, pp. 1–6, 1998, doi: 10.1016/S0925-2312(98)00030-7.

[16] Y. M. Cheung and L. T. Law, "Rival-model penalized self-organizing map," *IEEE Trans. Neural Networks*, vol. 18, no. 1, pp. 289–295, 2007, doi: 10.1109/TNN.2006.885039.

[17] V. Chaudhary, A. K. Ahlawat, and R. S. Bhatia, "An efficient Self-organizing map learning algorithm with winning frequency of neurons for clustering application," in *Proc. 2013 3rd IEEE Int. Adv. Comput. Conf. IACC 2013*, vol. 2, pp. 672–676, 2013, doi: 10.1109/IAdCC.2013.6514307.

[18] Ministry of Communication and Information of the Republic of Indonesia. Palapa Ring Project. Retrieved from https://web.kominfo.go.id/sites/default/files/Proyek%20Palapa%20Ring.jpg

Ridha M. Negara was born in North Sumatera, Indonesia, in 1985. She received a B.S. degree from the Telkom Institute of Technology, Indonesia, in 2009 and an M.S. degree from the Telkom Institute of Technology, Indonesia, in 2012, both in telecommunication engineering. She pursues her Doctoral course at Bandung Institute of Technology, Indonesia. Her research interests include Named Data Networking, Software Defined Networking, VANET and Artificial Intelligent.

Ratna Mayasari was born in North Sumatra Province, Indonesia, in 1985. She received her B.S. degree from the Telkom Institute of Technology, Bandung, in 2009 and her M.S. degree from the Telkom Telkom Institute of Technology, Indonesia, in 2013, both in wireless telecommunication engineering. She is pursuing a PhD at the School of Electrical Engineering and Informatics, ITB, Bandung. Her research interests are Telecommunication Networks, Named Data Networking, and Artificial Intelligence.

Nana R. Syambas He graduated with his bachelor's degree in Electrical Engineering Department, ITB, in 1983. He got his Master of Research degree from the Royal Melbourne Institute of Technology, Australia, in 1990 and his doctoral degree from the School of Electrical Engineering and Informatics, ITB, in 2011. He is currently a Full Professor with School of Electrical Engineering and Informatics Bandung Institute of Technology, Indonesia. His research interest includes Telecommunication Networks, Telematic Services, Content-Centric Network (CCN), Software Define Network (SDN), Protocol engineering and Tele-traffic engineering.