

Node Insertion Algorithm and Ant Colony Optimization: Performance Review on Travelling Salesman Problem Paper for the Optimality of Network Topology

Tutun Juhana¹, Selvi Lukman², Nana Rachmana³, Faizal Husni⁴, Gelar Pambudi Adhiluhung⁵, and Hafizh Mulya Harjono⁶

¹Telecommunication Research Group, School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, Indonesia ²Computer Science, School of Computer Science, Bina Nusantara University, Bandung Campus, Jakarta, Indonesia 11480

^{3,4,5,6}Telematics Laboratory, School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, Indonesia

tutun@itb.ac.id, selvi.lukman@binus.ac.id, nanasyambas@itb.ac.id

Corresponding author: selvi.lukman@binus.ac.id

Abstract: The integrated industrial control systems require a reliable communication among each part of control systems. The limitation of network topology comprises the low latency of data transmission. The network algorithm topologies are sometimes failed to ensure the fault-tolerant network design even if they offer it. Therefore, this works investigates a Node Insertion Algorithm (NIA) and Ant Colony Optimization (ACO) in order to solve the Travelling Salesman Problem (TSP) for network topologies in a communication network. The NP Non deterministic (polynomial time)-hardness based of TSP leads to the impossibility of polynomial time algorithm to solve an exponentially increasing amount of communication addresses. TSP itself is a well-known solution method known in the graph theory which is based on approximation algorithm. Therefore, this work also presents a comparison between the NIA and ACO for 200 nodes of network topologies. The results show that NIA is superior than ACO in terms of resulting ring's cost and computation duration. Three topologies are tested and it is observable that NIA requires only 14.88 seconds to finish the computation in which the resulting ring's length reaches 164019 units. Therefore, the achievement of the NIA can be applied for logical and physical nodes for a reliable connection in an industrial topology network.

Keywords: ACO; NIA; Network Topology; Optimization; TSP

1. Introduction

Network topologies are important in arranging the telecommunication network and relative location in traffic flows and industrial control process communication. The best placement of nodes and the most optimal path for traffic flow are required to design a well-defined network topology to locate faults and errors, to fix issues and to improve data transmission. Adjusting a suitable network topology is a challenging issue as it provides a direct effect on the network functionality. When it is properly chosen, the right topology can increase the data transfer rates and energy efficiency [1]. Network topologies are classified into logical and physical network topology. The physical network topology comprises physical layout of connections and nodes including lines that connect many communication nodes such as fiber optics, microwaves, Ethernet or digital subscriber line wires [2]. However, logical network topologies contain the setting up of connection for each node and patterns of data transmission [3]. Before constructing a network, a diagram of network topology should be firstly designed in order to prepare which components should constitute the network and their interaction. Listing all the devices such as servers, firewalls and routers must be firstly conducted before choosing the appropriate network topology. Accordingly, the chosen devices should be located in areas that are possible for low latency data flows and finally connection lines can be drawn from each network device. To mitigate having too many line crossings, it is obliged to provide a clear and readable diagram.

Received: October 16th, 2023. Accepted: December 30th, 2023

DOI: 10.15676/ijeei.2023.15.4.6

The common representation of network topologies is depicted in Fig.1

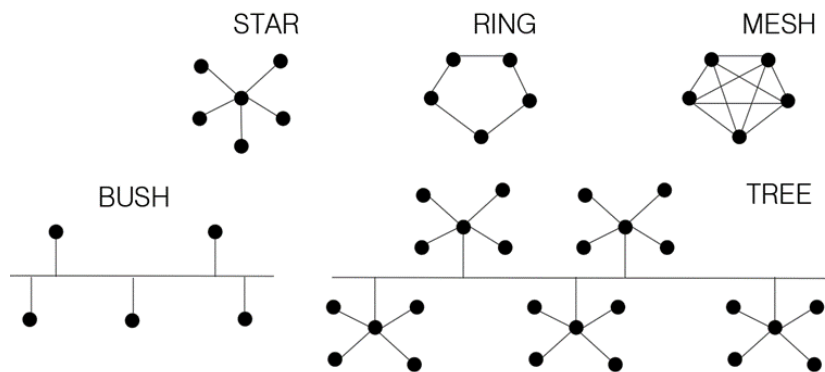


Figure 1. The representation of topology nodes configuration

Creating an optimum network topology has its own challenges in determining the most optimal ring according to the necessities of network conditions. As displayed in Fig.1, many varieties of topology networks lead to the requirement of finding the shortest path between nodes as network addresses tend to become more complicated. The problem remains similar as those nodes must be wired with lots of routers. The nature intelligence has provided some solutions in which one of them was the utilization of ACO in evaluating the performance of different network topologies [4]. Another author offered a better solution for TSP in terms of load efficiencies by using Brute Force. However, it could not guarantee the larger matrix values if additional destinations are put into use [5]. A well-known population-based search metaheuristic algorithm of an improved Genetic Algorithm was experimentally investigated in the prior research. An improvement of network reliability under some given constraints has yielded satisfying solutions for complex issues of computer environment [6]. Series of Greedy Algorithm seemed to be promising in creating a hierarchical network topologies and it outperformed networks the simple star topologies up to 75% [7]. The emerging machine learning approach was performed to identify a network topology for Croatian transmission system. The simulation of Artificial Neural Network (ANN) has shown the worst and best prediction case in the form of Receiver Operating Characteristic (ROC), however it ignored the theory of TSP completely [8]. Substantial contribution of a Deep Neural Network in order to assist a Monte Carlo Tree has yielded a desirable improvement when facing an increasing amount of computing resources compared to the heuristic solutions [9]. However, heuristic solution was still preferable as TSP remains the favorable foundation for finding the shortest path [10]. Even though, several novel techniques have been proposed and verified, this work favors the investigation of Node Insertion Algorithm (NIA) as it has several unique techniques which can improve the performance of network topologies and also reduce the negative behavior typically related with ACO-based methods especially when working on the impact of the control parameters settings in the connection with the tendency to fall prematurely into a local optimum and graph configuration [11]. Consequently, this study is aimed to investigate the comparison of Node Insertion Algorithm (NIA) and Ant Colony Optimization (ACO) in order to establish a reliable network topology based on TSP.

As nodes clustering principle has brought novelty in two ways. Firstly, there is none of single candidate list but sets of candidate list. They comprise a complete number of vertices. Secondly, the vertex selection for transition is conducted in two phases. The initial procedure is to determine a candidate list followed by vertex selection. In both phases, the probability of selecting either a vertex or a list is depended on the heuristic information and the intensity of the pheromone trails. Thus, the principal strength of ACO is still preferable to preserve the full set of vertices as ACO does not require a special strategy if there is not a free vertex inside the single candidate list[1][2]. Finally, the reminder of the paper is structured as follow. After the

introduction, the TSP in integer is discussed in chapter 2, followed by a deep insight about the Ant Colony Optimization in chapter 3. The Node Insertion Algorithm is analyzed in chapter 4. Testing and results are presented in chapter 5 and conclusions are drawn in chapter 6.

2. The Traveling Salesman Problem in Integer

People around the world always try to solve a problem of visiting more places in the shortest way. In a weighted graph, graphs with edges provide an associated cost. The associated cost is in the form of distance between cities. The goal of TSP to discover the shortest possible route in which every city must be visited once and it must return to the starting point. The natural phenomenon create population-based optimization algorithms in which TSP is adapted [3]. As TSP itself is one of non-deterministic polynomial (NP) problems, it leads to some drawbacks when the total number of paths, cities or nodes increase exponentially. However, many efficient solution of TSP are applicable for various real world problems such as network management, scheduling and transportation control [4]. Assuming that $D(C_i, C_j)$ is the distance between each pair of cities C_i and C_j . The solution in TSP is defined as permutation $\langle C_{\pi(1)}, C_{\pi(2)}, \dots, C_{\pi(n)} \rangle$ of the given n cities that minimizes

$$D = \left(\sum_{i=1}^{n-1} d(c_{\pi(i)} + c_{\pi(i+1)}) \right) + d(c_{\pi(n)}, c_{\pi(1)}) \quad (1)$$

The tour starts at the first city $C_{\pi(1)}$, all cities are visited in sequence and shall return directly to $C_{\pi(1)}$ from the last city $C_{\pi(n)}$ [5]. Until now, there is not a complete and efficient algorithm for solving the TSP. Many algorithms have been developed and explored for solving the TSP. The approximation algorithm can obtain the TSP's optimum solutions quickly but still cannot satisfy the degree of a near optimum solution, compared to the exact algorithm. This work presents 200 cities or nodes in the terminology of network topologies. Figure 2 illustrates set of distances, cities and the shortest possible route in this study [6].

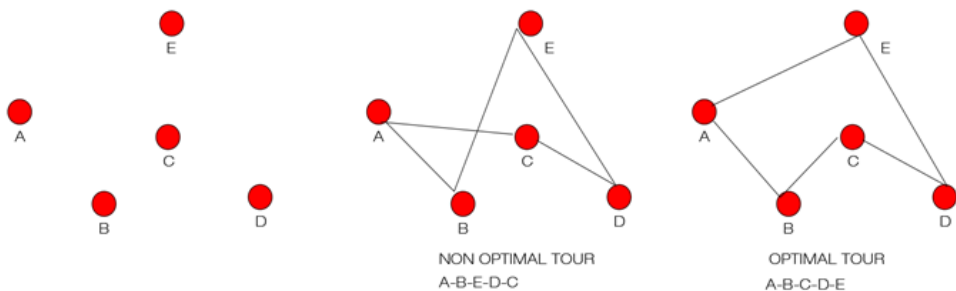


Figure 2. The description of TSP

In Fig.2, the inputs represent collection points or cities in which the key objective is to find a tour of a minimal length. The tour's length is the summing function of inter-point distance along the tour. The inputs are defined as $(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})$. The solution space is calculated for all possible tours in which the cost of tour represents the total length. The key objective is to find the minimal cost or length or what so called the Euclidian TSP in which the cities are distributed independently and randomly in a d-dimensional unit hypercube.

Exact algorithms can provide accurate solutions of TSP, however high computation resources require an efficient time during the computation process. Approximation algorithms can easily

achieve the desirable optimum solution of TSP but they are not capable to fulfil the degree of a near optimum solution, compared to the exact algorithm. Accordingly, this study initiates the utilization of ACO in which the result is compared with the Node Insertion Algorithm. By this procedure, a good approximation for solving the TSP for the most optimal network topologies can be achieved.

3. The Ant Colony Optimization

The Traveling Salesman Problem (TSP) provides a crucial role in the Ant System (AS) algorithms. The Ant Colony Optimization provides some agents that shift from one city to another city. When the tour is finished, they return back to the starting city. In ACO, solution for TSP are conducted by iteratively deposit pheromone on the arc of cities [7]. Figure 3 displays the procedure of ACO in order to achieve the most optimal solution.

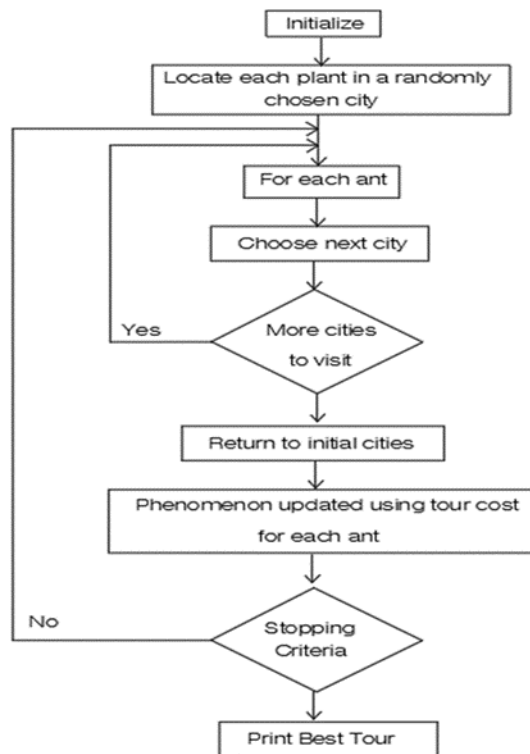


Figure 3. The optimization procedure of the Ant System

In Fig.3, each ant must randomly choose a city at the beginning according to the heuristic information and pheromone values on the edge between the current city and the next one. Their paths represent solution in a form of trail updates. A single path provides the pheromone amount which coincides with best solution for the TSP problem. In practical, ACO is utilized in many TSP cases[8]. The representation of cities is denoted as N and number of ants are denoted as M . Every ant shall make a solution tour in a city location randomly. The next city shall be chosen based by a probability function which depends on pheromone trail which is accumulated on heuristic value and edge. The random proportional rule for choosing the next city is formulated as follows

$$P_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{i \in \square_i^k} [\tau_{ij}]^\alpha [\eta_{ij}]^\beta} \quad \text{if} \quad j = \square_i^k \quad (2)$$

The influence of heuristic information and pheromone trail is defined as α and β in which α regulates the influence of pheromone train τ_{ij} . Heuristic value η_{ij} is regulated by β and accordingly, the process of choosing the next city is shown as below

- $\alpha < \beta$ the closest is prioritized to be chosen
- $\alpha > \beta$ arcs with more pheromone intensity are more likely to be utilized
- $\alpha = 0$, heuristic value is utilized for choosing without pheromone
- $\beta = 0$, the utilized pheromone with heuristic value will lead to poor result

As $\eta_{ij} = 1/d_{ij}$ denotes the available heuristic value, therefore, d_{ij} is the distance between cities i and j [9]. The pheromone trail matrix τ_{ij} and $\square_i^k$ defines the neighborhood set in which ant k has not been yet visited. The probability of ant k chooses arc (i, j) in order to increase the value of pheromone τ_{ij} and heuristic value η_{ij} . In this study, the initial value of pheromone matrix is set to a small value which is greater than zero as follow

$$\forall (i, j) \Leftrightarrow \tau_{ij} = \tau_0 \text{ where } \tau_0 > 0 \quad (3)$$

Generally, ants choose the city with larger amounts of pheromone by connecting edges between cities. Every ant will add a new city to its tour by updating the new value of pheromone to the old τ_{ij} as follow

$$\tau_{ij}(new) = \tau_{ij}(old) + \Delta\tau_{ij} \quad (4)$$

The value of $\Delta\tau_{ij}$ changes according to tour length L_k as formulated below

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (5)$$

The updating process of pheromone trail continues until all ants complete their tour by visiting all cities and return to the start city. After all ants terminate the tour, pheromone evaporation and global pheromone update the shortest tour. The shortest tour will have the greatest amount of pheromone. Pheromone evaporation will reduce the amount of pheromone in arcs which leads to a gradually disappearance of pheromone trail in the unused arc [10]. This process makes the ants follow one path under the formulation

$$\tau_{ij}^{new} = (1 - \psi) \cdot \tau_{ij}^{old} \quad (6)$$

whereas $(0 < \psi < 1)$ is a constant utilized quantity for reducing the pheromone. When the value of the information heuristic factor α becomes larger, it is likely the probability of ants will choose a local path based on the concentration of pheromone. However, when the value of expected heuristic factor β , ants will tend to choose a local shorter path. Therefore, the randomness of search will decrease and the utilized algorithm will tend to be greedy. The volatility factor ρ is ranging from 0 to 1. It indicates that if ρ is too small then the pheromone on the path will not volatilize in time. The excessive pheromone on the path will affect the efficiency of algorithm's

convergence. On the other hand, if ρ is too large, the pheromone on the path is not retained and the ant colony will not experience the information from previous iterations. Furthermore, when the pheromone coefficient Q is enhanced more pheromones will be released by the ant in a single update. The pheromone accumulation on the path will become faster and also the proportion of state transition probability will increasingly be affected, thus it can be easily fall into the local optimum. ACO provides a better solution over the Genetic Algorithm (GA) and Simulated Annealing (SA) over similar cases especially when the graphs change dynamically. Accordingly, ACO is easily adapted in real time and makes it possible in most routing cases.

4. The Node Insertion Algorithm

Generally, the Node Insertion Algorithm is regulated by inserting an arbitrary node to an existing sub-ring. This procedure will lead to the least of sub-ring in each step [22]. The Node Insertion Algorithm is a kind of a heuristic approach as many optimal solutions in the form of resulting ring are not easy to determine. Theoretically, the comparison of the existing links for each new node must be provided because this algorithm provides a computational complexity. However, in practical the computation step is quite simple and fast.

The Node Insertion Algorithm begins with a TSP sub-tour. Inserting a new node in every step is performed until a valid Hamiltonian path is achieved. Insertion a new node (x,v) means one old edge between two old nodes (u,v) is deleted and create two new edges by connecting (u,x) and (v,x) [11]. The common ways to do the insertion are described as follows

- Nearest Insertion: In every step, the nearest node to any node in the existing sub-tour is selected
- Cheapest Insertion: Choose (x) in which distance $(x,v) + \text{distance } (x,u) - \text{distance } (u,v)$ is minimized at each procedure.
- Farthest Insertion: The opposite of the Nearest Insertion is performed by choosing node (x) to maximize distance (x,v) for any v in the existing sub-tour.
- Random Insertion: Every step must choose a random node that has not been visited and insert it at the best possible position according the tour length

The following flowchart in Fig.4 displays the procedure of obtaining the best solution for TSP.

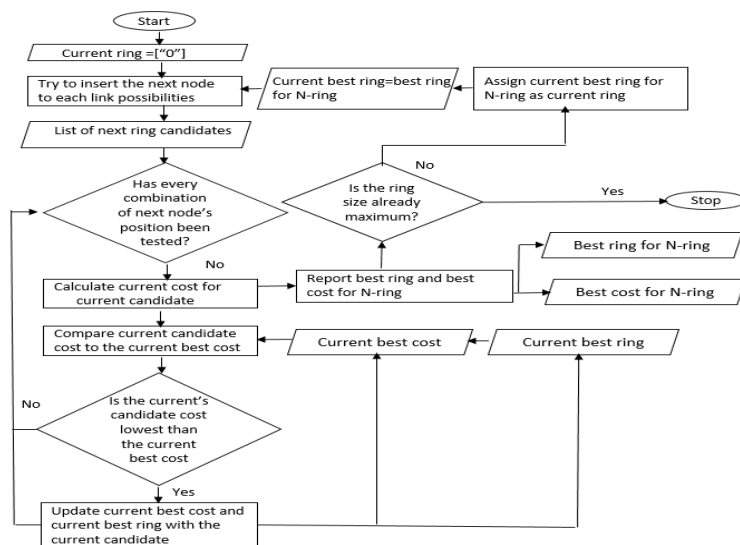


Figure 4 The flowchart of Node Insertion Algorithm

For network topologies optimization, double for loop is utilized. Accordingly, this algorithm provides theoretical time complexity of n^2 to a faster computation time. The maximum number of nodes per ring is depended on the switches required to aggregate the system. The maximum ring sizes are also calculated from the total number of nodes in the system. An intuitive approach to the TSP using node insertion algorithm provide several possibilities for the implementation of the insertion schemes, since even the algorithm does not guarantee an optimal solution but desirable approximations in a reasonable amount of time can lead to effective and efficient solution.

5. Testing and Results

In this study, Node Insertion Algorithm and Ant Colony algorithm are analyzed and compared for 200 nodes in three different topologies. The ring’s cost is described by the total Euclidian distance between the neighboring nodes. The computation is performed by using Python 3 in which the runtime is conducted on a Google Cloud Platform’s Compute Engine with Ubuntu 20.04 LTS OS, 4 vCPUs and 16 GBs of RAM. The Ant Colony Optimization algorithm utilizes 100 ants and 0.5 of pheromone factor ratio in which 1 for alpha and beta factor. The maximum number of 100 iterations are performed to optimize the parameters. For the Node Insertion Algorithm, the node order is chosen arbitrarily. The results are presented in the following graphs

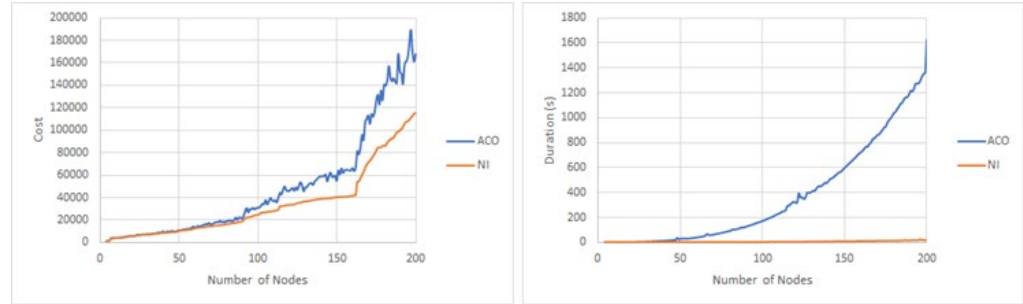


Figure 5 Cost (left) and computation time (right) performance for the first topology

In the first topology, Ant Colony Optimization requires 1627.25 seconds to calculate and the resulting ring’s length is 167389.41 units. The configuration of the first topology can be observed in [12]. On the other hand, the Node Insertion Algorithm provides a faster computation which is only 17.08 seconds and the resulting ring’s length is only 115194.44 units. In the second topology, the Ant Colony Optimization requires 853.38 seconds in the computation process and the resulting ring’s length is 5914805 units. The configuration for the second topology can be seen in [13].

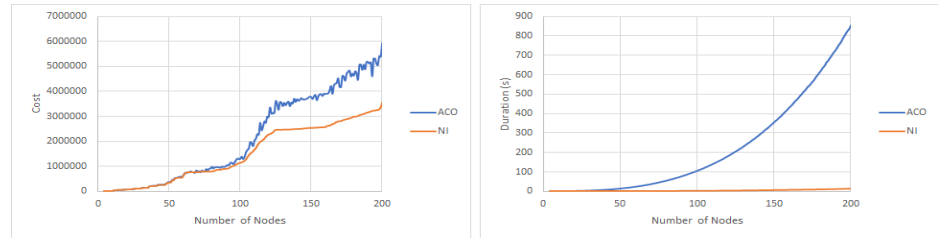


Figure 6 Cost (left) and computation time (right) performance for the second topology

On the other hand, the Node Insertion algorithm yields even faster computation in only 15.69 seconds in which the resulting ring’s length achieves 3525487 units as depicted in Fig.6.

In the third topology as depicted in Fig.7, the Ant Colony Optimization performs the computation in 1364.43 seconds and the resulting ring's length is 285805.91 units. On the other hand, the Node Insertion algorithm requires only 14.88 seconds to calculate and the resulting ring's length is 164019 units. The third topology can be reviewed in [14]. All topologies are presented in the form of XML from which is built on top of multi-layer network representation.

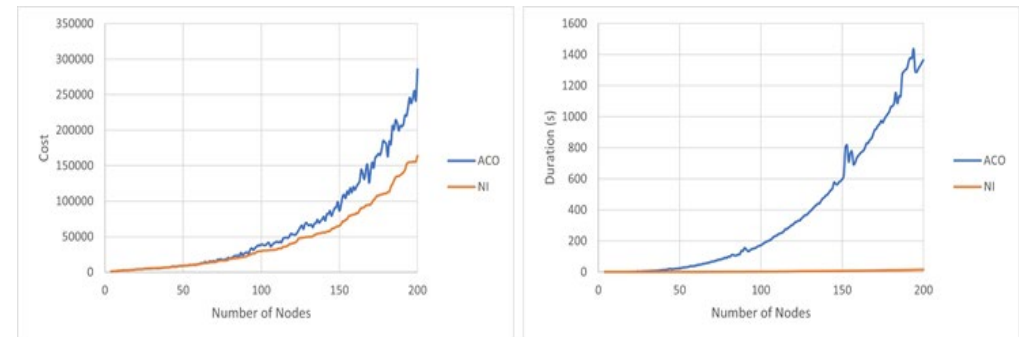


Figure 7 Cost (left) and computation time (right) performance for the third topology

For a better comparison visualization, Table 1 displays the recapitulation of the algorithm results for three topologies.

Table 1. The Recapitulation of Algorithm Results

Topology	Applied Algorithm			
	Node Insertion Algorithm		Ant Colony Optimization	
	Computation time in seconds	Resulting Ring in units	Computation time in seconds	Resulting Ring in units
First	17.08	115194.44	1627.25	167389.41
Second	15.69	3525487	853.38	5914805
Third	14.88	164019	1364.43	285809.91

It is observable that Node Insertion Algorithm is more superior than Ant Colony Optimization as it almost eliminates the computational time uncertainty for deriving the minimal number of serially multi-polling sequences. Therefore, the Node Insertion Algorithm increases the performance of the algorithm. More complex problem will lead to the higher performance improvement for NIA in which the general ACO ignores deterioration. Hence, it is important to integrate the general ACO algorithm with the local search optimization to further reduce deterioration

6. Conclusions and Future Works

This paper attempts the analysis of TSP for network topology optimization. The performance of Node Insertion Algorithm and Ant Colony Optimization was presented and it is concluded that Node Insertion Algorithm is far more superior than Ant Colony Optimization both in the resulting ring's length and computation time. As the Node Insertion Algorithm has achieved a good approximation for solving the Traveling Salesman Problem for 200 nodes within a reasonably short duration. As three topologies are tested for NIA and ACO in which NIA requires only 14.88 seconds to finish the computation in which the resulting ring's length reaches 164019 units. Therefore, it is observable that NIA is more superior both in terms of resulting ring's cost and computation duration. However, it is advisable to experiment with other optimization techniques so the most optimal network topology for a reliable transmission connection can be realized. For further future works, simplification or decomposing problems might be helpful to

determine heuristics or metaheuristics methods in order to find the best solution in a reasonable time.

7. References

- [1]. H. Fahmi, M. Zarlis, E. B. Nababan, and P. Sihombing, "Ant Colony Optimization (ACO) Algorithm for Determining the Nearest Route Search in Distribution of Light Food Production," *J. Phys. Conf. Ser.*, vol. 1566, no. 1, 2020, doi: 10.1088/1742-6596/1566/1/012045.
- [2]. A. M. Armond, Y. D. Prasetyo, and W. Ediningrum, "Application of Ant Colony Optimization (ACO) Algorithm to Optimize Trans Banyumas Bus Routes," in *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, 2022, pp. 132–137, doi: 10.1109/CyberneticsCom55287.2022.9865394.
- [3]. S. Sangwan, "Literature Review on Travelling Salesman Problem," *Int. J. Res.*, vol. 5, p. 1152, Jun. 2018.
- [4]. G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *Eur. J. Oper. Res.*, vol. 59, no. 2, pp. 231–247, 1992, doi: [https://doi.org/10.1016/0377-2217\(92\)90138-Y](https://doi.org/10.1016/0377-2217(92)90138-Y).
- [5]. S. Sánchez, G. Cocho, J. Flores, C. Gershenson, G. Iñiguez, and C. Pineda, "Trajectory Stability in the Traveling Salesman Problem," *Complexity*, vol. 2018, 2018, doi: 10.1155/2018/2826082.
- [6]. T. S. Problem, *Travelling salesman problem: a foot-in-the-door?*, vol. 1, no. 4. 1997.
- [7]. M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, 2006, doi: 10.1109/MCI.2006.329691.
- [8]. R. Bhavya and L. Elango, "Ant-Inspired Metaheuristic Algorithms for Combinatorial Optimization Problems in Water Resources Management," *Water*, vol. 15, no. 9. 2023, doi: 10.3390/w15091712.
- [9]. P. Duan and Y. AI, "Research on an Improved Ant Colony Optimization Algorithm and its Application," *Int. J. Hybrid Inf. Technol.*, vol. 9, no. 4, pp. 223–234, 2016, doi: 10.14257/ijhit.2016.9.4.20.
- [10]. M. Mulani and V. L. Desai, "Design and Implementation Issues in Ant Colony Optimization," *Int. J. Appl. Eng. Res.*, vol. 13, no. 16, pp. 12877–12882, 2018, [Online]. Available: <http://www.ripublication.com>.
- [11]. V. Rajaramon and V. Pandiri, "A Multi-Start Iterated Local Search Algorithm for the Bottleneck Traveling Salesman Problem," in *2022 IEEE 19th India Council International Conference (INDICON)*, 2022, pp. 1–7, doi: 10.1109/INDICON56171.2022.10039842.
- [12]. <https://bit.ly/48oieCj>
- [13]. <https://bit.ly/41FY30h>
- [14]. <https://bit.ly/41IgLUS>



Tutun Juhana received the Ph.D. degree from the Institut Teknologi Bandung (ITB), Indonesia in 2011. He is currently an Associate Professor in telecommunication engineering with the School of Electrical Engineering and Informatics (SEEI), ITB. His research interests include wireless ad-hoc networking, vehicular ad-hoc networks, the Internet of Things, and network protocols design and analysis. He also serves as The IEEE Circuits and Systems Society Indonesia section.



Selvi Lukman was born in Bandung, West Java, Indonesia, in 1975. She received her Bachelor degree of Electrical Engineering in Universitas Jendral Ahmad Yani (UNJANI), Indonesia. In 2019, She received her Master degree in Physics Engineering Department from Institut Teknologi Bandung Indonesia. In 2023, She has achieved her Ph.D in the Engineering Physics Department, Institut Teknologi Bandung, Indonesia. Currently, she is lecturing in Computer Engineering Department of Universitas Kristen Maranatha and School of Computer Science Bina Nusantara University. Her research areas are included Electrical and Energy System, Control and Artificial Intelligence, Transportation and Smart Decisions and Wireless Communication.



Nana Rachmana graduated with a Bachelor's degree in Electrical Engineering from ITB in 1983, obtained Master by Research degree from Royal Melbourne Institute of Technology, Australia in 1990, and a doctoral degree from School of Electrical Engineering and Informatics, ITB in 2011. He has been a Professor on Telecommunication Engineering since 2017 and a lecturer at the School of Electrical Engineering and Informatics, ITB since 1984. His research interest includes Telecommunication Networks, Telematics Services, Content-Centric Network (CCN), Software-Defined Network (SDN), Protocol Engineering and Tele-traffic Engineering. He has authored or coauthored over 100 published articles. Prof. Dr. Nana Rachmana Syambas is a member of IEEE, VTS, and ITS.



Faisal Husni received his bachelor degree in Telecommunication Engineering Program from School of Electrical Engineering and Informatics, ITB in 2021 and master degree in April 2023 from Electrical Engineering Master Program in the same school.. After graduation he joined Ula as software engineer. He is currently with Samsung R&D Institute Indonesia.



Gelar Pambudi Adhiluhung received his bachelor's degree in Telecommunication Engineering Program from School of Electrical Engineering and Informatics, ITB in 2021 and master degree in April 2023 from Electrical Engineering Master Program in the same school. He is now with OY! Indonesia as Site Reliability Engineer.



Hafizh Mulya Harjono received his bachelor's degree with cum laude in Telecommunication Engineering Program from School of Electrical Engineering and Informatics, ITB in 2021 and master degree in April 2023 from Electrical Engineering Master Program in the same school also with cum laude. He is now with OY! Indonesia as Site Reliability Engineer.