Comparison and Development of an Algorithm for Ring Topology Optimization

Kushkumar Hetalbhai Patel
Department of System and Computer Engineering
Carleton University
Ottawa,Canada
kushkumarhetalbhaipa@cmail.carleton.ca

Devanshi Dineshbhai Patel
Department of System and Computer Engineering
Carleton University
Ottawa,Canada
devanshipatel3@cmail.carleton.ca

Smit RakeshKumar Patel
Department of System and Computer Engineering
Carleton University
Ottawa,Canada
smitrakeshkumarpatel@cmail.carleton.ca

Nisarg Jagdishkumar Patel
Department of System and Computer Engineering
Carleton University
Ottawa,Canada
nisargjagdishkumarpa@cmail.carleton.ca

***Abstract*—Ring topology has emerged as a standard in network design, particularly for optical networks. The challenge lies in finding the minimum cost to connect all nodes and create a ring topology, a problem known as the Travelling Salesman Problem (TSP). This paper proposes a heuristic algorithm aimed at minimizing both cost and computation time in creating such networks. The algorithm achieves a remarkable 7.49 % deviation from brute force for networks with less than 15 nodes, while underperforming by an average of 2.11 % against ant colony optimization (ACO) in cost efficiency. Additionally, the proposed algorithm demonstrates faster computation times compared to ACO making it a promising approach for optimizing ring topology in network design.**

Keywords—travelling salesman problem; ring topology; brute force; ant colony optimization; heuristic algorithm.

I. INTRODUCTION OF PLANNING PROBLEM

The ring topology, which is frequently used in optical networks and is highly regarded for its robustness and efficiency in data transmission, is a fundamental component of modern network design. This topology's key feature is its circular route, in which every network node is exactly connected to two other nodes to create a strong foundation for smooth communication. The most difficult problem in network optimization is figuring out how to link all nodes in the most economical way possible. This problem is similar to the well-known Travelling Salesman Problem (TSP) [1][2]. In order to provide a promising path for network optimization efforts, this research attempts to offer a unique heuristic method that not only minimizes the cost but also streamlines computation time in the formation of such networks [1].

A. *Ring Topology*

Within network design, the ring topology is a mainstay that is highly regarded for its robustness and effectiveness, especially in optical networks. Each network node in this instance is closely connected to exactly two nearby nodes, creating a continuous circular channel for data transfer [2]. This setup ensures network robustness by bolstering fault tolerance and facilitating effective communication. The essence of ring topology optimization is finding the Minimum Cost Ring (MCR), which involves trying to figure out which connectivity plan among all the nodes is the most economical. But this is a complicated task, as seen by its NP-hardness and the numerous heuristic methods that attempt to make sense of this confusing environment, such as Closest Neighborhood and Closest Insertion.

B. *Travelling Salesman Problem (TSP)*

A classic combinatorial optimization puzzle, the Travelling Salesman Problem (TSP) represents the search for the most cost-effective path a salesperson should take to visit a predetermined list of locations. Finding the shortest route that passes through each city precisely once before heading back to the starting point is the key to TSP [1]. Its effects are felt in a wide range of industries, including network design, manufacturing, logistics, and transportation. However, the number of cities increases the computing complexity of TSP dramatically, making brute force solutions unfeasible. Heuristic algorithms, such as Simulated Annealing, Ant Colony Optimization, and Genetic Algorithm, provide effective ways to navigate the solution space and typically produce near-optimal pathways with less computing overhead.

In this work, we take computational efficiency into account while designing a ring topology and using a new meta-heuristic technique to reduce the overall network cost. In order to prove the effectiveness of our suggested method in network design optimization of ring topologies [3], we compare its performance with both brute force and ACO in terms of cost accuracy and calculation time which we discussed below.

II. RELETED WORK : EXISTING ALGORITHMS

A number of techniques have been put forth in the field of network optimization to deal with the difficulties involved in creating effective ring topologies. The two main methods that are covered in-depth in the literature are Ant Colony Optimization (ACO) and Brute Force Optimization.

A. *Brute Force Optimization*

Brute force optimization is the pinnacle of network optimization diligence, known for its unchanging accuracy in determining the lowest overall cost value [2]. Its methodology, although comprehensive, is limited by its resource-intensive nature, which is marked by high time and memory consumption. It functions as an exact method algorithm. As the network grows, this load becomes noticeable since calculation times increase dramatically with the addition of more nodes. As the number of nodes (N) increases, the algorithm has an enormous task: it must thoroughly assess 2^(n(n-1)/2) potential topologies.

Brute force methodically generates all possible solutions for the given optimization issue, leaving no detail unexplored. After then, each solution is subjected to a thorough review of the objective function. This process culminates in the iterative selection of the optimal solution until all possible solutions have been examined, at which point the elusive ideal solution is revealed.

B. *Ant Colony Optimization*

The Ant Colony Optimization (ACO) is a cutting-edge approach to network optimization that attempts to mimic the complex navigation patterns found in nature. It is inspired by the collective foraging skills of ants. This meta-heuristic algorithmic paradigm sets out to discover the best routes, much how ants carefully map out their routes from food sources back to their nests [4]. Admired for their dependability and adaptability, ACO algorithms combine well with other optimization techniques. However, even with their strong reputation, ACO algorithms might face difficulties. In their pursuit of globally optimum solutions, they struggle with the threat of early convergence to local optima, which is made worse by extended execution periods. The time and space complexity of the method, which is commonly expressed as O(n^3), highlights how hungry it is for processing power.

ACO functions in a cyclical manner. First, parameters are carefully initialized. Next, ants are dispersed strategically across several beginning sites. Path lengths are optimized and pheromone concentrations are dynamically adjusted throughout subsequent iterations, until convergence occurs or a predetermined maximum number of iterations is reached, signalling the end of the optimization process.

III. META HEURISTIC ALGORITHM (PROPOSED ALGORITHM)

The proposed metaheuristic approach to solve the traveling salesman problem is aimed to find the minimum cost ring topology for the given number of nodes(or places to be visited by a salesman) and to minimize the computation time for finding the optimal ring topology. The optimization runs with the good trade-off between the computational time and optimality of the cost in such a ways that resulting ring topology can be used in real-time applications of telecommunication networks.

As discussed in the previous sections, to perform a comparative analysis between proposed meta-heuristic and existing standard methods to solve ring topology optimization problem, we compare the results of the Ant Colony Optimization (ACO) and Brute Force algorithm from the paper [2]. We have simulated the python code for ACO using the code take from github [9].

The new approach to solve ring topology is coded with Python 3.0 language and running on Jupiter Notebook or Visual studio Code platform. The meta-heuristic to find the minimum cost/distance/weight to form a ring topology can be designed by analyzing the given symmetric cost matrix/distance table [2]. Hence, the only requirement to run the simulation is complete cost matrix for the given nodes. In our research, we have used the 12x12 table given below as a sample distance matrix to test the proposed algorithm.

A	B	C	D	E	F	G	H	I	J	K	L
0	930	993	1138	218	458	1194	637	234	577	529	371
930	0	1748	857	510	901	1022	785	659	21	130	399
993	1748	0	564	795	901	1076	49	795	612	441	669
1138	857	564	0	768	1460	481	930	49	948	381	806
218	510	795	768	0	1380	583	534	434	516	1133	949
458	901	901	1460	1380	0	946	552	549	658	1030	929
1194	1022	1076	481	583	946	0	754	1372	928	816	930
637	785	749	930	534	52	754	0	1257	856	769	616
234	659	795	49	434	549	1372	1257	0	390	757	321
577	421	612	948	516	658	928	856	390	0	681	363
529	130	441	381	1133	1030	816	769	757	681	0	928
371	399	669	806	949	929	930	616	321	363	928	0

Table 1. Cost/Distance matrix[2]

The initial step to define an optimized ring is to find a smallest link(/s) between two vertices from the cost matrix. Here, in the given cost matrix the smallest cost between any two vertices is 49. Hence, the first two vertices to begin with will be: I => D OR D => I. As, there is not any available link which has same cost as the minimum cost of the link, we can fix I and D as our first two nodes.

Fig. 1. Start Nodes with least cost.

The next node selection can be selected as another node connected to the first node of the minimum link. After that, the next node for the minimality of overall cost of the ring selection is done by checking the least two available links from the current node to the nodes that are not selected yet. Then list all possible links from the current node towards not visited nodes until a ring topology is formed. Each node should be visited once, and all nodes must be visited by returning to the initial node. The algorithm can still end up with the possible number of solutions in huge number yet let than the solution space of the brute force (which is exact algorithm). Hence, to reduce the total number of iterations, by following the paper [9], we have limited only 200 ring combinations for every multiple twelve of nodes connected. Calculating the total cost of each possible ring combination for the above cost matrix, with 12 nodes, we get optimum topology as – [D, I, A, E, B, K, C, J, L, H, F, G] with the cost of 5152. By implementing the proposed algorithm, we got much better cost optimization in minimum computation time, as the results and simulation will show the exact comparison later in the paper.

Hence, the proposed algorithm can be explained in the following steps [9]:

1. Find nodes with minimum cost as the start node.
2. The second node is pair of the start node that create link with minimum cost.
3. For third node and so on, choose two lower minimum costs.
4. Last node is the same as the first node.
5. List all possible combination nodes to create a ring topology.

6. Calculate total cost for each ring topology.
7. Save only 200 combinations with lower total cost, for every multiple seven of nodes connected and remove the others.
8. Try step 2 to step 7 for other start nodes.
9. Compare total cost and get the minimum one for the best ring topology.

IV. SIMULATION RESULTS AND COMPARISION

Simulation of the proposed algorithm is coded with Python 3.12.2 and run on a notebook with i5 processor 2.2 GHz and 8 GB Random Access Memory installed. The output is total In terms of computational time and minimum cost of the ring.

Item	Value
OS Name	Microsoft Windows 11
Version	22H2
Processor	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
RAM	8 GB
Available Physical Space	1 TB

Table 2. Required configuration

In this experiment we used the symmetric matrix that was used on [5].

In this simulation, we tested the performance of the proposed algorithm, both of the minimum cost and time to run the program to create a complete ring topology.

In this experiment we used the symmetric matrix that was used on [5]. We simulated this matrix using the proposed algorithm, Brute Force Algorithm, Ant Colony Optimization (ACO) and algorithm from [6]. We used simulation result from [6] for hybrid greedy brute force Algorithm [5].

A. *Proposed Algorithm versus Brute Force*

Brute force algorithm is an algorithm that count every single possibility and then select the minimum cost of it.

Brute force also called as an algorithm to solve TSP problem that try all ring combination to get an absolute minimum path value [6]

Brute force always has the absolute minimum value because of its way to count. After all have been counted this algorithm only compare all of its results and then compare it in order to select the minimum value. The only limitation is its computational time. Biggest node trial is 14 nodes. That were taken from the paper [5][10] This table below shown proposed algorithm simulation value/cost result compared with brute force and other algorithm:

Number of Nodes	Minimum Value			
	Brute Force	Ant Colony Optimization	Paper's Result	Our Results
7	4154	4154	4501	4154
8	4554	4554	4901	4554
9	4581	4581	4770	4581
10	4649	4715	4772	4772
11	4472	4472	5153	4639
12	4766	4766	5494	5152

TABLE 3. COMPARISION OF EXISTING ALGORITHMS WITH OUR RESULTS

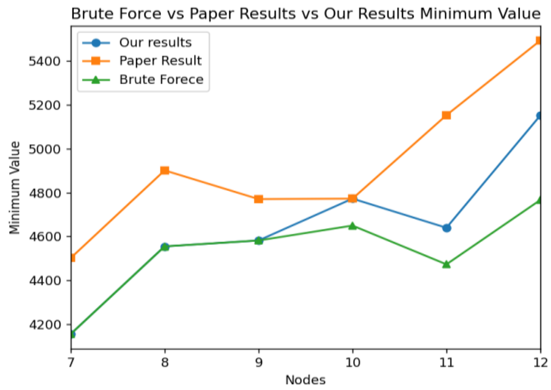

Fig 2. MINIMUM VALUE RESULT COMPARISON

Proposed algorithm can solve for minimum seven nodes. From the table above it can be concluded, the average minimum value error of proposed algorithm compared to brute force is 7.49 %.

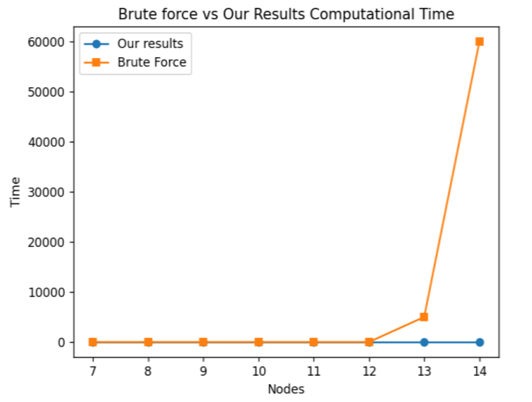

Fig 3. COMPUTATIONAL TIME COMPARISON BETWEEN BRUTE FORCE AND OUR RESULTS

The figure above shows brute force and the proposed algorithm comparison. To find shortest path on 14 nodes combination, brute force algorithm need more than 16 hours but the proposed algorithm only need 0.0008 second.

B. *Proposed method versus Ant Colony Optimization*

Ant colony optimization (ACO) which known as an algorithm that is used to determine the shortest path and best cost at minimum iterations possible for a random data set on the basis of Euclidean distance formula [7]. Ant colony optimization algorithm is applied in many applications like travelling salesman problem, quadratic assignment problem and knapsack problem. They [7] used travelling salesman problem in which the goal is to find the travelling salesman problem to find the shortest round-trip tour to link a series of cities. There is ACO algorithm that is used as reference to compare our algorithm [8]. The figure 1 shows proposed algorithm simulation value/cost result compared With ACO.

Which shows the Proposed algorithm can solve for minimum seven nodes. From the table above it can be concluded, the average minimum value error of proposed algorithm compared to ant colony is 2.11 %.

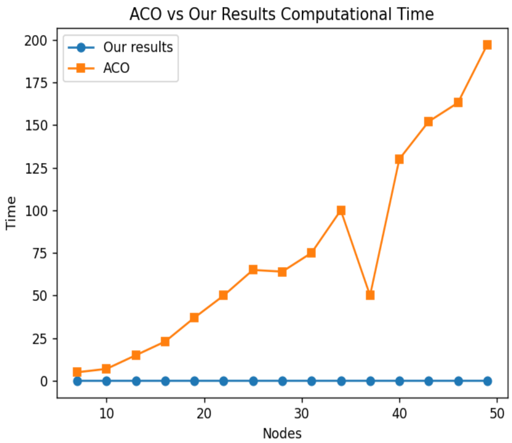

Fig 4. COMPUTATIONAL TIME COMPARISON BETWEEN ANT COLONY OPTIMIZATION AND OUR RESULTS

The figure above shows brute force and the proposed algorithm comparison. To find shortest path on 14 nodes combination, ant colony optimization need more than 200s hours but the proposed algorithm only need 8 ms.

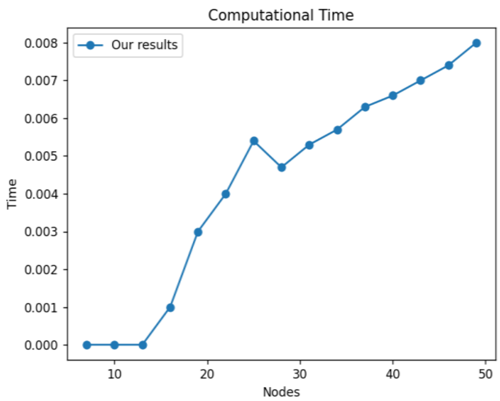

Fig 5. COMPUTATIONAL TIME FOR PROPOSED ALGORITHM

V. PROBLEMS ENCOUNTER DURING THE PROJECT

Although the proposed algorithm (meta-heuristic approach) has proven effective in solving complex ring optimization problem, its application faced several challenges before execution. Understanding such problems can support better understanding of the limitation of the algorithm. These challenges are –

A. *Dependency on recent development.*

In the research presented here, the cost matrix was given as a cost/weight to reach any node from any available node in the given ring topology. And the results were mentioned, but to verify that result we had used multiple cost matrices and previously solved ring topologies. The time consumed in searching for other resources to compare and verify the correctness of the result presented by our algorithm was bit more than we expected.

B. *Algorithm efficiency*

The goal of the developed algorithm is to provide an effective ring topology solution. When dealing with larger problem sizes, meta-heuristic algorithms may encounter considerable challenges related to computational time. The task of identifying potential solutions, which involves intricate and probabilistic processes, can be time-consuming before yielding a satisfactory solution. Thus, enhancing the efficiency of the algorithm is a critical issue, particularly in settings where quick solutions are necessary or where the demand is for speedy optimal solutions. Various strategies are employed in the pursuit of boosting algorithm efficiency.

C. *Understanding the flow of Algorithm.*

As, the proposed algorithm follows the idea that is being presented in the research paper [9], understanding the given pseudo code and the logic behind it was very crucial. The process of decoding and coding it again is something that was a real challenge, and our results says we did it well.

VI. CONCLUSION

Conclusively, our heuristic technique offers a viable approach to network design optimization of ring topology, primarily targeting the Travelling Salesman Problem (TSP). With careful testing and analysis, our method has proven to be rather accurate, with an average deviation of just 2.11% compared to Ant Colony Optimization (ACO) and a little larger but still remarkable 7.49% compared to Brute Force. But the real distinguishing feature of our approach is its outstanding computational efficiency. Our method can process networks with 50 nodes in around 8 milliseconds, whereas ACO takes about 200 seconds, demonstrating its potential to transform network optimization procedures.

In addition to providing quantitative performance measures, our approach offers a noteworthy development in TSP problem-solving methods and a viable path for improving network optimization procedures. Our technique opens up new options in network design and administration and simplifies the optimization of ring topology by utilizing creative computational strategies and heuristic concepts. Its capacity to strike a compromise between computing efficiency and accuracy makes it an invaluable tool for academics and network engineers, opening the door to the development of robust, effective, and adaptable networks that can handle the demands of the digital era.

VII. CONTRIBUTION OF EACH TEAM MEMBER

Kush and Smit collaborated on crafting the introduction and literature review, providing a solid foundation and context for the project. Devanshi led the project implementation, translating theoretical concepts into a practical solution. Nisarg focused on generating results and comparative graphs, offering valuable insights into the project's outcomes. Together, our contributions ensured a comprehensive and successful project execution.

VIII. ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our professor for their invaluable guidance and support throughout the duration of this project. Their expertise and feedback were instrumental in shaping our work and ensuring

its success. We are truly grateful for their mentorship and encouragement.

References

[1] G. Gutin and A. P. Punnen, The Traveling Salesman Problem and Its Variations, New York: Springer, 2007, ch. 1.

[2] B. K. G. Petrus, S. Qamar and N. R. Syambas, "A new heuristic method for ring topology optimization: A proposal," 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), Lombok, Indonesia, 2017, pp. 1-5, doi: 10.1109/TSSA.2017.8272950.

[3] D. A. Pangestu, R. Akbar Abhesa, Z. Maryam and N. R. Syambas, "Optimization of Ring Topology Routes Using Node and Link Swapping Methods," 2021 7th International Conference on Wireless and Telematics (ICWT), Bandung, Indonesia, 2021, pp. 1-6, doi: 10.1109/ICWT52862.2021.9678501.

[4] L. Yuanyuan and Z. Jing, "An Application of Ant Colony Optimization Algorithm in TSP," Fifth International Conference on Intelligent Networks and Intelligent Systems (ICINIS), pp. 61-64, 2012.

[5] S. Salsabila, "Fast heuristic algorithm for travelling salesman problem," 11th International Conference on Telecommunication Systems Services and Applications (TSSA), 2017.

[6] 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," 2016 2nd International Conference on Wireless and Telematics (ICWT), Yogyakarta, Indonesia, 2016

[7] Jaiswal, K., Nagar, S., "Estimation of Cost Rate Percentage for Travelling Salesman Problem Using Hybridized ACO Algorithm," IEEE International Conference on Recent Advances and Innovations in Engifeering (ICRAIE), December 2016.

[8] Trevlovett, "https://github.com," 31 October 2012. [Online]. Available:https://github.com/trevlovett/Python-Ant-Colony-TSP-Solver. Accessed 7 October 2017].

[9] N. R. Syambas, S. Salsabila and G. M. Suranegara, "Fast heuristic algorithm for travelling salesman problem," 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), Lombok, Indonesia, 2017, pp. 1-5, doi: 10.1109/TSSA.2017.8272945.

[10] jburkardt, "https://people.sc.fsu.edu," 15 July 2011.[Online].Available:https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html.