ENERGY OPTIMIZATION IN WSN USING ML ALGORITHMS

Abstract

As the evolution progresses from 5G (Fifth Generation) towards Beyond 5G and 6G (Sixth Generation), energy efficiency has become an important objective for wireless communication systems. Rising network density and complexity call for low-energy approaches to facilitate sustainability and longer lifetimes for networks, particularly for WSNs (Wireless Sensor Networks). This work advocates an energy-conscious routing mechanism based on ACO (Ant Colony Optimization), motivated by the natural foraging behaviour of ants. To cope with dynamic and resource-aware settings, ML (Machine Learning) mechanisms are embedded for smart decision-making and optimization. DT (Decision Trees), SVM (Support Vector Machine), KNN (K-Nearest Neighbour), and DL (Deep Learning) have been successfully implemented in wireless applications. ACO was selected for its low computational complexity and distributed and adaptive routing mechanism and has been tested previously in various WSN experiments for increased data delivery and energy efficiency. The findings presented are applied from these experiences to align with the changing demands of the 5G and B5G.

Key words: 5G, Network, ML, complexity, data delivery, routing, WSN, energy efficiency

1. Introduction

The evolution from 5G to beyond 5G (B5G) and future 6G wireless communication systems requires not only higher data rates and ultra-low latency but also tremendous improvement in energy efficiency. With network infrastructures growing denser to accommodate a vast array of applications, like the Internet of Things (IoT), smart cities, and autonomous systems, the energy efficiency of communication nodes becomes a serious issue. Energy efficiency, therefore, becomes a critical design target in today's wireless systems. Significantly, the environmental viability of Information and Communication Technology (ICT) equipment has been under pressure with its increasing carbon footprint, necessitating energy-conscious solutions more than ever before. Energy-efficient networks also save service providers operational expenses (OPEX), increase device battery life, and meet changing regulatory requirements. Specifically, Wireless Sensor Networks (WSNs), which are commonly used in remote or resource-limited scenarios, have limited energy supplies. Intelligent and energy-conscious routing strategies need to be developed to maximize network lifetimes and minimize operating expenses. Our project solves this problem by suggesting an energy-efficient routing method based on Ant Colony Optimization (ACO), inspired by the foraging behaviour of ants[1], which is a real-world application for future 6G environments.

To control the complexity and dynamic nature of such networks, Machine Learning (ML) has turned out to be a must-have. ML algorithms can process big data, learn patterns, and make decisions autonomously in real time. They assist in automating processes, minimizing human interference, and streamlining system performance even in unstable and dynamically changing environments. In the case of WSNs that are incorporated into 5G networks, ML enables processes like smart resource allocation, traffic forecasting, anomaly detection, and adaptive routing. Some ML algorithms have been successfully used in communication systems, such as Decision Trees (DT), Support Vector Machines (SVM), K-Nearest Neighbours (KNN)[2], and Deep Learning (DL) models. These algorithms perform classification, regression, and clustering processes well, providing flexible and scalable solutions for network optimization problems.
Among different optimization methods, Ant Colony Optimization (ACO) is notable because it is decentralized, adaptive path finding, and reliable in dynamic conditions. ACO, which mimics the real ants' behaviour of following pheromone trails to find and establish the best paths, is especially suited to routing issues in WSNs. It adaptively adjusts routes according to evolving network conditions with energy-efficient communication.

Several studies have proved that bio-inspired algorithms like ACO, Particle Swarm Optimization (PSO)[3], Genetic Algorithm (GA) and Artificial Bee Colony (ABC) are efficient in promoting energy efficiency in WSNs. For example, studies have shown that ACO-based protocols have greatly enhanced packet delivery as well as energy conservation compared to conventional approaches. The algorithms have been used in cases that include cluster-based routing, mobile sink optimization, and QoS-aware path choice in energy-limited environments. Following on from this available literature, our research suggests an improved ACO-based model designed for energy-efficient communication in 5G and B5G scenarios, combining ML-based optimization techniques to adapt to contemporary wireless challenges[4].

1.1 WHY ACO?

The choice of using Ant Colony Optimization as the fundamental routing mechanism in this research is grounded in its proven effectiveness in dynamic, distributed, and constrained environments—qualities that closely match Wireless Sensor Networks (WSNs) employed in 5G and B5 G [5]communication systems. ACO, being a nature-inspired metaheuristic algorithm that mimics the foraging behaviour of ants, has been proven to find optimal routes by laying down pheromone trails. This decentralized method enables each network node to make autonomous, localized routing decisions based on local information alone without global control or centralized coordination. The pheromone densities in shorter and more direct paths are higher, so ants are likely to choose these routes and thus amplify the pheromone densities—a positive feedback cycle[6]. The effectiveness of ACO-based routing has been established by its extensive employment in many applications that involve distributed control and dynamic networks with computational resource constraints, and therefore, it is ideally suited for the resource-constrained WSNs used in 5G and future wireless communication infrastructure.

1.2 Efficiency of ACO Over Alternative Techniques

ACO offers a set of advantages, making it significantly effective for energy-efficient routing[7] in wireless communication systems. Firstly, its energy-saving path finding minimizes unnecessary retransmissions and hence maximizes the network's overall lifetime. Through iteratively updating routes based on pheromone intensity and route cost, ACO only reinforces the most cost-efficient routes over time. Compared to other metaheuristics like Genetic Algorithms (GA), Particle Swarm Optimization (PSO)[8], and Artificial Bee Colony (ABC)[9], ACO has a lighter computational footprint, which is desirable for real-time processing in WSN nodes with limited processing capacities. In addition, ACO scalability and flexibility support dense and massive deployments, which are requirements paramount in 5G and B5 G scenarios. The ability of the algorithm to incorporate Quality of Service (QoS)[10] parameters such as delay, throughput, and energy consumption are included in its path selection factors, which provides further performance enhancement. Empirical findings have consistently shown that ACO-based routing protocols outperform traditional protocols in packet delivery ratio, end-to-end latency, and residual energy[11], thereby proving their superiority in efficiency and reliability.

1.3 Limitations

While it has numerous advantages, ACO is far from being limitation-free. One of the primary limitations is its slow convergence during the initial stages of route discovery, particularly in large networks. It may be objectionable in time-sensitive applications where communication in real time is crucial[12].

The ACO behaviour is highly parameter-sensitive, such as pheromone evaporation rate, heuristic influence, and updating frequency. Incorrect configuration can cause too much exploration or premature convergence to suboptimal routes.

Another concern is path stagnation, where high pheromone levels on suboptimal paths prevent the discovery of better alternatives. In addition, maintaining and updating pheromone tables at each node incurs memory and communication overhead, which can strain the limited resources of WSN nodes. In extremely dynamic network conditions[13], ACO may have to be extended or hybridized with machine learning techniques to remain effective and responsive.

2. Methodology

2.1 How ACO works:

Ant colony optimization is a bio-inspired algorithm that was introduced by Marco Dorigo in the time 1990[14]. Ants are a species that will always prefer to be in a group rather than individually. So, there should be a method for the ants to be in a group; they will use pheromone acid to communicate with each other. Pheromone is a chemical secreted by the ants that helps them to communicate with other ants[6]. So, they will go through multiple paths for the least and efficient way to reach their destination, and the remaining ants also follow the same path using the pheromonic which has been released. Similarly, in this case of a Wireless sensor network, we follow this Ant colony optimization technique for a good and energy-efficient path. So, the pheromone level is the deciding factor for the optimized path[15].

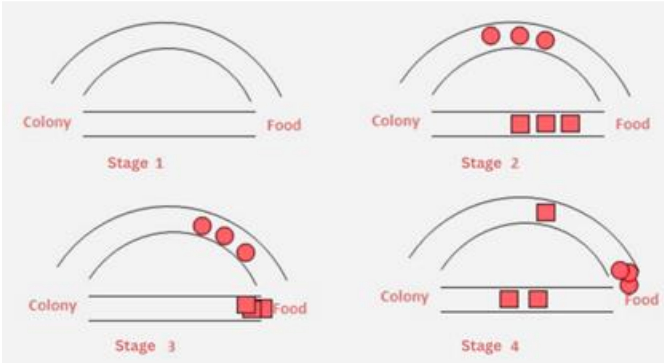

Fig.1 Process of ants searching for food

Figure 1 describes how ants efficiently search for food[16]. This process all 4 levels, describes the same; stage 1 shows that all the ants are in their nest and there will be no pheromone level in the environment. So, the path will be empty. In the stage 2 Ants will begin their search for the food, assuming there is a probability of 0.25 along each path, from the above figure we can say that the curved path is very long and will take more time to reach the destination from that path, So, ants will automatically change their path which is very efficient and not that much time taking[17]. In stage 3, the probability of selecting a good path is very high because the ants came across the shortest path, leaving a pheromone level, which makes them select the shortest path very quickly, so now they will reach the starting point on the same path through which they came from, which makes the role of pheromone level crucial in this stage. Finally, in the stage 4 more number of ants pass through the same path leaving the pheromone level, so we will assume that there is more concentration of pheromone level but, due to the evaporation the pheromone concentration will also gradually decrease in the longer paths reducing the probability of selecting that path in the future. We can justify that the optimization has been achieved in stage 4[18], which brings an end to the search for food.

2.2 Describing the parameters:

Apart from the real-world environment, it is not much like how we implement them all in practice, so we use some key parameters for optimization. If I take an example of ants when they don't use these parameters, it would be difficult for them to find the good path from all[19]. Similarly, in practice, these parameters help to build an optimized path. That saves us from just wandering over there.

Let us compactly see the significance of each parameter:

2.2.1 Alpha (α) This Parameter is treated as the sense of history, similarly to how our Google account keeps tracking all our information, this parameter also keeps track of the pheromone level. When ants return to their destination, this α It will help them to move to their destination (Their initial point).

2.2.2 Beta (β): Beta helps to calculate the shortest distance, so without beta the ants might take longest pheromone route even it is unnecessary. This parameter is very important in our wireless sensor network, as all the electronic devices are battery oriented, it is very important to take a right decision while traveling over the nodes, which will minimize the energy and will give us the shortest possible path among all the routes. So, beta will save energy by taking the least path.

2.2.3 Gamma (γ): Gamma's main task is to avoid all the low-energy nodes while travelling. There will be so many devices whose power will get reduced after N number of transmission's so in the next iteration, gamma will not allow selecting the low-powered devices and make to pick healthier nodes for the routing. Gamma will help ants, and here in this case nodes, not just to select the shortcuts but also be aware of which route is being selected and what is the left-out energy in their surroundings. Because in the wireless sensor network, information must be passed securely without any distortion, if any node picks a bad node, it will face challenges moving to the next node.

2.3 Formulas used in the ACO algorithm:

2.3.1 Transition probability formula:

$$P_{ij} = \frac{\tau_{ij}^{\alpha} * n_{ij}^{\beta} * E_j^{\gamma}}{\sum_{k \neq} visited\ \tau_{ik}^{\alpha} * n_{ik}^{\beta} * E_k^{\gamma}} \quad (1)$$

The transition probability formula itself says that it is used to calculate the probability of a node moving from point i to j based on the previous results. In nature, this factor helps the ants to decide their next move, and in the Wireless sensor network, this helps to make a clean decision on the next hop and parallelly with the less cost. [20]This formula will always prefer the shortest distance for fast communication and on the other side it avoids low energy nodes for clear transmission of data without any distortions.

Let us now examine the significance of each term:

In Ant Colony Optimization, the term τ_{ij}^{α} Stands for the pheromone between two points i and j. This pheromone level helps ants to decide which path would be worth taking, it is like a memory of which routes have given them the efficient path before, so that they can reach the source on the same path. The higher the level of pheromone on a path, the more attractive it becomes. On the other hand, the parameter n_{ij}^{β} (where $n_{ij} = \frac{1}{d_{ij}}$) Reflects how the next best path is based on its distance. It's usually calculated as the inverse of the distance between two nodes. $\frac{1}{d_{ij}}$, meaning shorter paths are considered more favourable. Together, these factors guide the decision-making process, balancing experience with efficiency in finding the next best move. Assume initially the distance is very small, then the 1/d ratio will become more, making it a good path. In the second case, if let distance is more than 1/d ratio becomes very small, making it a less efficient path. The denominator also has some role in this formula; it will ensure that the sum of probabilities is 1, and only unvisited nodes are to be considered to avoid unwanted loops in the system.[20]

2.3.2 Pheromone update rule:

$$\tau_{ij} = (1-\rho) * \tau_{ij} + \Delta\tau_{ij} \qquad (2)$$

The above formula is used to update the pheromone level. In the previous formula, we saw how the probability of selection is being taken, and here we need to update the level from hop to hop. As ants travel over too much area in nature, it is very difficult for them to keep the memory of a previous long route, which is unnecessary, and can be resolved by this pheromone update rule. In the context of WSN, selecting the next hop is the key factor in the optimization. So, the task is how the formula is getting updated dynamically over time, how it is adapting to the network changes, and so all[20].

Considering the challenges in the Wireless sensor network:

The WSN is dynamic, so there is a high chance of nodes getting damaged or their battery getting reduced, then it shouldn't take the help of the nodes in optimization, and all these things will happen within seconds. Without this, there is a high probability of nodes getting used regularly, which will make them inefficient in future use, so the term (1-ρ) ensures that previously travelled nodes might be used less this time.

$$\Delta\tau_{ij} = \frac{1}{L} \qquad (3)$$

Here, L denotes the length of the efficient path.

2.3.3 Path Length calculation formula:

$$L = \sum_{k=1}^{N-1} d_{k,k+1} \qquad (4)$$

The path length formula is used to sum all the distances we got from node k to k+1 for the N number of nodes. So, the output shows the quality of the routing, or we can say how well the routing has taken place. In the context of a wireless sensor network, this calculation helps to identify how well the transmission of data took place. This will mention the length of the routing it has taken, which can conclude the shortest total path. The main motto of this formula is to optimize the length of the routing, as in our project, both the routing length and cost of traveling are the key factors.

2.3.4 Energy consumption calculation:

$$E_{used} = d * C_{tx} * C_{rx} \qquad (5)$$

We have been continuously saying that energy is a key parameter that needs to be optimized, and here comes the main formula, which is the energy consumption calculation[21]. His formula will calculate the energy between two end devices. As the data is getting transmitted in the network, there would be a lot of energy intake and out-take, which needs to be stored while optimizing the energy itself. Using tE_{used}we can monitor the network lifetime and avoid any unnecessary distortions or node failures.

Below is the demonstration of formula (5)

The mentioned energy consumption calculation formula will calculate the total energy consumed while travelling from one node to another node. Because getting the shortest path may be easy for the nodes that are placed in the geographical area, but how much energy they are consuming and what would be the leftover energy with which the remaining nodes and all data can transmit would be an important thing. Here 'd represents the distance between the transmitter and the receiver nodes, and as the distance increases, the power consumption also increases, so we must keep a note of every node on how much power it is consuming and how much is left, as all are battery-oriented. C_{tx} denotes the amount of energy consumed to send one bit of data[21], and this factor depends on which type of device we are using and the manufacturing quality from device to device. C_{rx} Denote the energy consumed while receiving one bit of data, which depends on the type of receiver we are using, and again, the manufacturing quality and all other parameters like the surrounding temperature, the climate, and the region. In general, the units for both C_{tx} and C_{rx} Are joules/bit.

3. Algorithmic Description

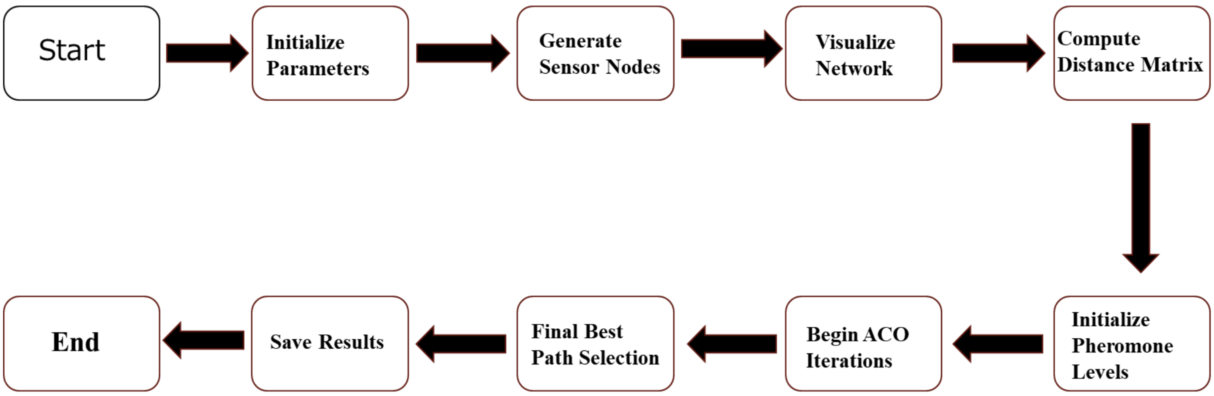

Fig. 2 Block diagram of ACO implementation

The above block diagram is the flow of implementation of our ACO code, where we divided the things into 10 parts[22]. Except Start and end there are 8 main steps which are very crucial in the implementation part, In the beginning of the code we will initialize the parameters that will be used in the formulas and also the number of nodes we are going to take and the area size in which we are going to perform this process, other parameters like initial energy, number of iterations will be declared in the beginning itself. As we are performing in the MATLAB software[23], we need to generate some source nodes using the rand function, which is used to create some random numbers that are uniformly distributed in the range of [0,1]. Then in step 4, we visualize the network, which means creating a 2D interface where I can see all the nodes and perform some operation on them. Step 5 will calculate the initial distance between each pair of nodes, which is very important in the routing and efficient path perspective. After that, we will initialize the pheromone levels based on which we are calculating the distance now. Then we will start the iterations one by one based on the area size and number of nodes, The software will take some time to give the best result from all the possible routes. Then we will end up with the best path, which consumes the least amount of energy. We can try this by increasing the number of nodes and area size, so we can conclude that this code can be able to give perfect information under any circumstances[23]. Finally, if you want, you can save all the results that are generated and end the simulation.

4. Simulation Setup

Parameter	Value	Significance
No. of nodes	1000	Represents the overall number of nodes deployed in the network
Area size	150	Area of the region where the nodes will spread
Initial Energy	100	Each node will begin with a fixed amount of strength
No. of ants	20	A wide variety of synthetic agents(ants) are used in the Ant colony optimization.
No. of iterations	100	Range of times the algorithm will run
Alpha (α)	1	The parameter that has an impact on the pheromone level
Beta (β)	2	This parameter determines the influence of heuristic facts
Gamma (γ)	1.5	It aids in adapting the routing preference
Decay	0.5	Refers to the price of pheromone evaporation
Early stopping threshold	10	A barrier that stops the iterations if no rule is obeyed
Transmission cost	0.1	The power utilized to transmit a packet
Reception cost	0.05	Power used to receive a packet

These are the parameters that we are going to declare at the beginning of the code, and we keep on varying the count of the nodes and sometimes the area to observe the difference in the energy consumption[24]. Beginning with the No. Of Nodes, this is dynamic for sure, as we are trying to match the results and compare the difference that is arising from count to count, so this will display the number of nodes that will get initialized here. Then, coming to the area size, we kept it constant at 150x150, and observing the results, it just depicts how well the nodes get spread in the given area. Then the focus will shift to the Initial energy, which needs to be specified for every node, and we took 100 as the starting energy for each node. We limited the number of iterations that can be performed is 100, as we think that this algorithm can perform the operations within a few iterations. As we discussed the significance of each term in the Methodology section here, we will assign some numerical to those values. They are alpha, α, β , and γ, whose initial values are already specified in the above table, and we took values based on one research paper where they specified the values between the range of 3-5 for an optimal result, but we took some basic values in our project. Decay refers to the price of pheromone evaporation. It avoids the set of rules getting stuck in suboptimal or antiquated paths. Then comes the term Early stopping threshold that restricts the iterations if some rules get disobeyed, like a constant value for 5-6 times, which reflects that there is no more efficient value in the next iterations and will stop the entire simulation then which will save our time and as well as system's performance[25]. At last, the two parameters will specify the amount of energy to transmit and receive the packets, as we know transmission takes slightly higher energy, so we specified a bit more value for the transmission cost and less for the receiver.

Now we will observe the results for both stable and unstable node cases.

4.1 Visualization for Stable Nodes

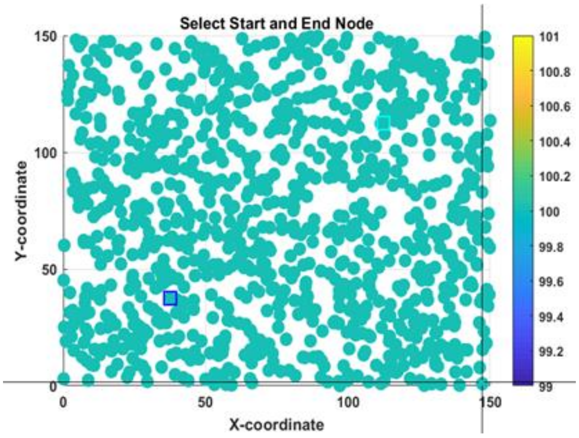

Fig. 3 Initial Stage of Iteration

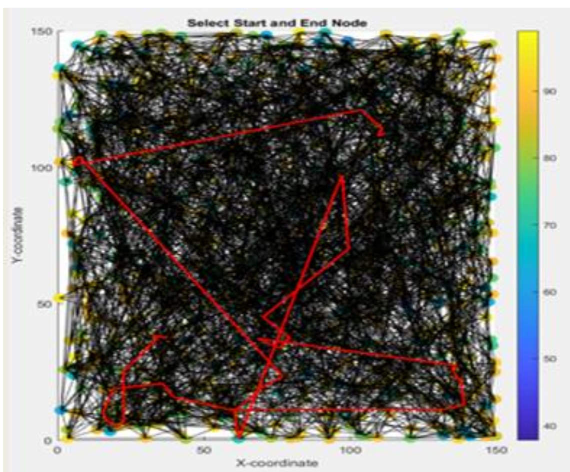

Fig. 4 Searching for an efficient path

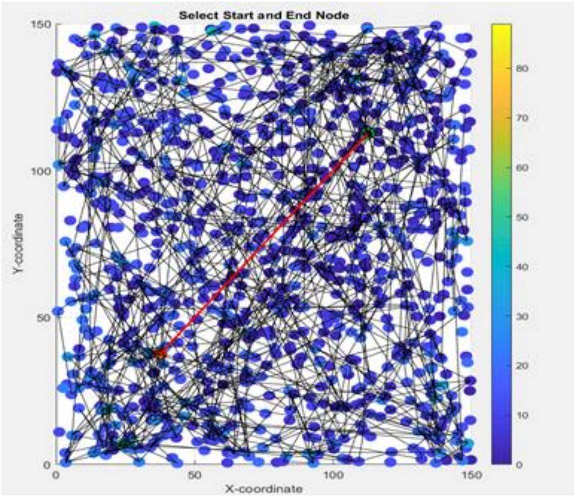

Fig. 5 Efficient path from all nodes

The above three figures represent the transitions when finding an energy-efficient path in Stable node cases[23]. In the figure 3 we have a certain number of nodes and among them only 2 nodes are highlighted using square boxes which will act as transmitter and receiver even when we increase the number of nodes or area size no parameter can change the position of those two nodes and we will too point only those 2 nodes even for n simulations. This is a practical application where we have n number of nodes in the network, but only 2 devices have fixed communication between them. This term is known as point-to-point communication in our Network field[26]. Figure 4 will then show all the possible routes that can be possible to go from source to sink, and that red line shows the current route that it is travelling. Subsequently, the number of paths will get reduced after a few iterations, which is represented in Figure 5. And at last, the path which got selected will use a smaller number of nodes and avoid unnecessary detours, showing the algorithm’s success in minimizing energy use. The overall visual flow demonstrates how the ACO approach improves routing step by step. These results can help improve energy management in sensor networks or communication systems.

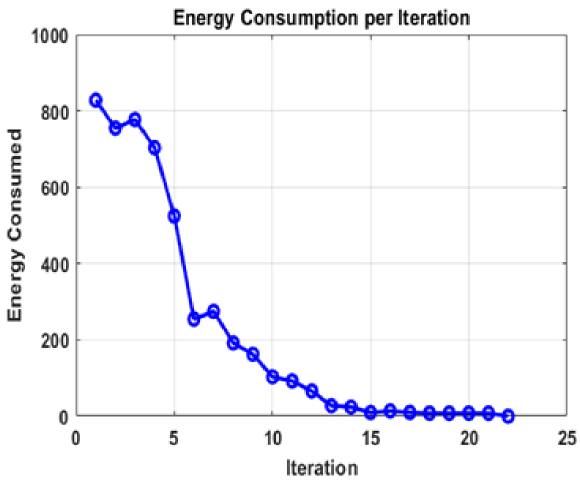

Fig. 6 Energy vs Iteration Graph

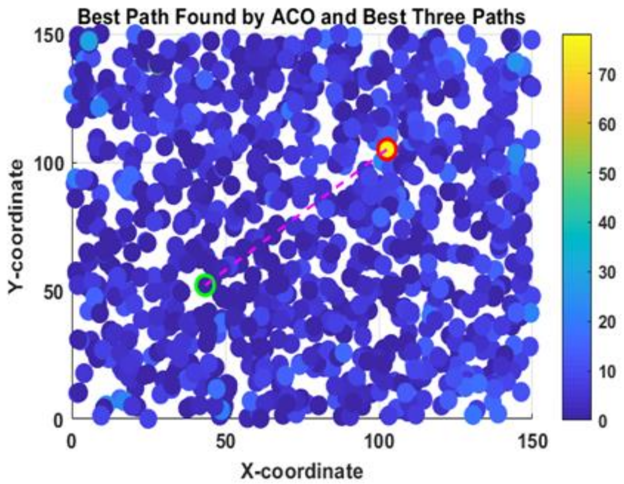

Fig. 7 Efficient path found by ACO

Figure 6 is representing the energy per iteration graph which is high at the beginning and will get reduced simultaneously as the number of iterations gets increased because if a certain region is occupying more energy, then ACO will not take that path again to reduce the time complexity of the algorithm it will select a new region where an efficient path is possible. So, like this, it will take a few iterations and will stop only when it encounters the same energy level 4-5 times, which means the path energy can't be reduced more from that point.[27] Figure 7 shows the final selected path with minimal energy and path cost, which is represented using a red dotted line. Moreover, the path will almost remain the same for n number of nodes as the nodes are fixed there and there will not be much change in the energy consumption too; it will also be a constant value.

4.2 Visualization for Unstable Nodes.

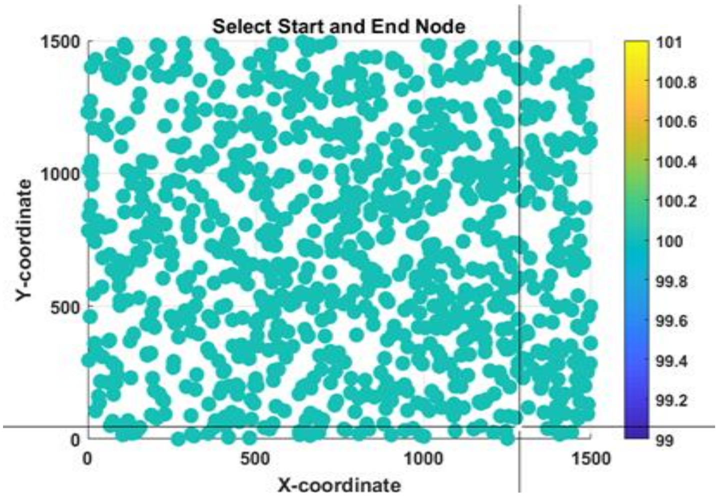

Fig .8 Selecting nodes in the Network.

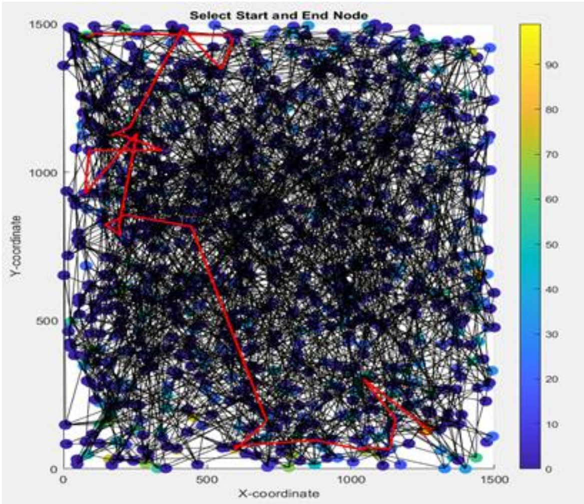

Fig .9 Searching for an efficient path.

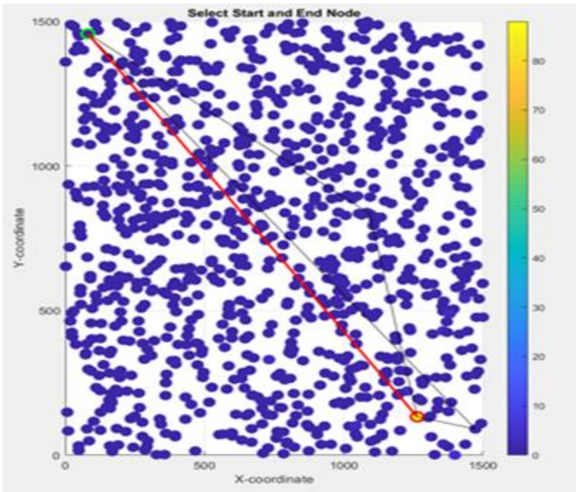

Fig. 10. 3 Best Paths in the Network

The above 3 figures are stages of finding an energy-efficient path in unstable cases, that is, where the nodes can be selected by the user dynamically. So, in this case, we can get different energy levels based on the distance and the path that has been selected. In Figure 8, we can see that dynamic nodes were selected from the network, and after selecting the nodes, the source node will be marked as green and the sink node will be marked as red, between which the routing will take place. And then in Figure 9, this is an intermediate stage of routing where the algorithm is searching for an efficient path from all the possible routes, and the red line shows the present node where it is traveling at that instant. Same as in the case of stable nodes, where it is searching for different possible nodes. This is the algorithm exploration phase. In Figure 10, we see how the algorithm filters and focuses on only the top three most efficient paths from all possible options. The light black lines are the 2nd and 3rd most efficient paths, which are possible after the 1st path. This can be useful in cases where, if the original path has some distortions or any node failure, in that case we can go with the 2nd and 3rd best paths. Let us explore the results of this unstable node case.

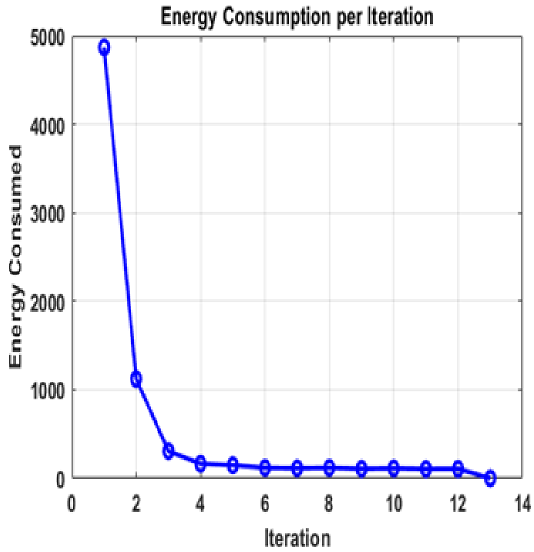

Fig .11 Energy vs Iteration graph

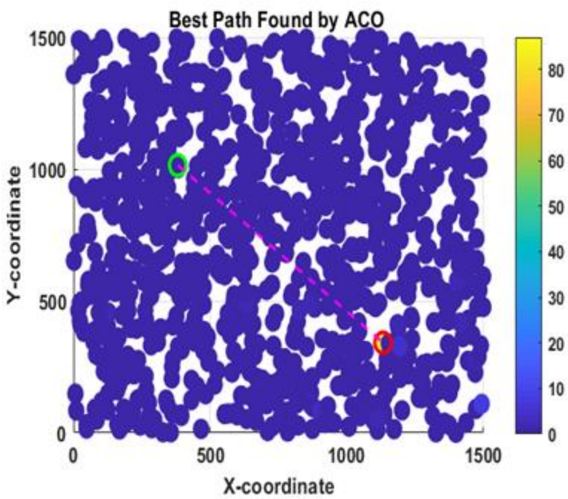

Fig.12 The best path for unstable nodes

Figure 4 represents a graph showing how energy consumption decreases drastically in the first few iterations and then stabilizes, indicating that the algorithm is learning and improving its choices. Figure 12 displays the best path chosen by the algorithm after all iterations, connecting the starting point and the destination point. This path uses the least amount of energy[28], as calculated through repeated simulations. The overall sequence of images shows how the ACO algorithm effectively narrows down its choices to find the most optimal route. It mimics the behaviour of ants, using trial and error along with feedback from previous results. Together, these visuals support the theory that bio-inspired algorithms can offer efficient solutions for energy-based pathfinding in complex networks.

5. Results and Discussion

From all the previous experience and the outputs, we are concluding that Ant colony optimization has been successfully implemented in the wireless sensor network. It succeeded in both finding the best path and energy energy-efficient path, improving the overall routing performance. It is capable of giving the best path even when there are more number of nodes in the network, even though it is taking some extra time to simulate[29] but at last it is giving a proper output, which we can verify from all the iterations the code will iterate continuously for 15-20 times and whenever it got a constant energy value for more than 5 times it will assume that there is no other way to minimize the energy and will route in that selected path. The minimization of the energy will help in increasing the lifespan of each node in the wireless sensor network.

Table 1 Energy Consumption per Iteration for Unstable Nodes

Iteration	50 Nodes	100 Nodes	1000 Nodes	10000 Nodes
Iteration 1	43.5564	106.488	5518.28	44958.6
Iteration 2	40.8557	82.6454	696.267	13732.3
Iteration 3	39.3899	59.9766	261.611	1869.03
Iteration 4	29.1818	55.9907	184.066	800.91
Iteration 5	20.8999	46.7174	188.322	576.268
Iteration 6	23.2925	46.2075	170.851	284.948
Iteration 7	18.4094	31.5694	163.703	268.071
Iteration 8	15.9522	22.9233	161.556	306.112
Iteration 9	15.0348	16.9678	168.639	197.303
Iteration 10	11.8701	20.1718	158.27	218.657
Iteration 11	12.5847	10.8489	158.048	187.259
Iteration 12	10.5471	12.3214	158.011	174.729
Iteration 13	nan	9.9828	nan	171.733
Iteration 14	nan	10.1885	nan	159.261
Iteration 15	nan	9.6852	nan	159.261

Table 2 Energy Consumption for Stable Nodes

Iteration \ Nodes	50	100	1000	10000
Iteration 1	111.9943	129.9943	747.3598	7763.4458
Iteration 2	65.1194	109.5547	484.9062	6614.1246
Iteration 3	37.5798	68.6618	545.4321	4994.877
Iteration 4	24.5948	64.3692	285.9873	4425.886
Iteration 5	14.7489	52.9842	386.7012	4667.6100
Iteration 6	13.4988	34.0871	300.1219	3370.9800
Iteration 7	11.0623	24.1069	240.4569	2308.8280
Iteration 8	11.3900	22.6061	151.8144	2235.9240
Iteration 9	10.6566	16.9629	121.2326	1885.3180
Iteration 10	11.8097	12.0308	169.0948	2314.8460
Iteration 11	10.3412	12.7709	188.8862	1151.9450
Iteration 12	–	10.9409	105.3975	1078.9510
Iteration 13	–	12.5627	118.8800	1076.9740
Iteration 14	–	10.6566	64.6195	576.0028
Iteration 15	–	–	97.4777	825.5008
Iteration 16	–	–	65.8248	477.8020
Iteration 17	–	–	49.1730	592.7244
Iteration 18	–	–	50.3536	462.7785
Iteration 19	–	–	26.9376	461.8278
Iteration 20	–	–	18.8500	395.4795
Iteration 21	–	–	10.7419	242.3114
Iteration 22	–	–	–	371.6652
Iteration 23	–	–	–	263.0242
Iteration 24	–	–	–	231.4771
Iteration 25	–	–	–	179.3835
Iteration 26	–	–	–	229.3330
Iteration 27	–	–	–	100.0284
Iteration 28	–	–	–	162.1941
Iteration 29	–	–	–	191.8936
Iteration 30	–	–	–	134.9009
Iteration 31	–	–	–	90.0371
Iteration 32	–	–	–	87.1735
Iteration 33	–	–	–	75.2007
Iteration 34	–	–	–	10.7877

The above two tables represent the Energy consumption values per iteration in both stable and unstable cases. We had checked the energy consumption values for 50, 100, 1000, and 10000 nodes at an area of 150x150 in MATLAB. What we observed is the ACO algorithm is capable of giving a fruitful information based on the difficulty and increasing number of nodes, The table 2 is all about the energy consumption in stable network for an increasing number of nodes there we can observe that it is succeeded in giving a constant result even when we increase the node count i.e. around 10.7, if I increase the number of nodes from 50 to 1000 or 10000 nodes it is taking approximately 20-25 minutes to process all the nodes because in the real world there will be Lakhs of nodes and connected devices through network where more number of nodes will take more time to transmit the data, which is exactly represented in our simulation part, The simulation time is getting increased as we increase the number of nodes in the network, We had verified with other algorithm techniques like Dijkstra's algorithm where the result is on point for up to 150-200 nodes but once we give 1000+ nodes in the setup it is not able to give the efficient path just showing a blank screen[30].

Our project aim is not only to align with the wireless sensor network but also in the VLSI field,[31]where efficient routing will be able to reduce the material cost and circuit complexity. According to Moore's law, the number of transistors is increasing every 2 years, which also increases the power consumption in the circuit, which is exactly what we represented through this project, wherever the count of nodes increases the power dissipation too will increase then it will be a problem for those devices which are running with the source of battery, then the real necessity for energy efficient path will arises which transmits the data to the destination in energy efficient way[32].

6. Conclusion

This project presents the implementation of Ant Colony Optimization (ACO) in Wireless Sensor Networks (WSNs) using MATLAB to address the challenge of energy-efficient routing. ACO, inspired by the behaviour of real ants, dynamically discovers optimal paths for data transmission, reducing energy consumption and extending network lifespan. Compared to traditional algorithms like Dijkstra's, ACO offers better adaptability, scalability, and performance in real-time scenarios. The results demonstrate that ACO effectively balances load, handles unstable nodes, and is a robust solution for optimizing communication in energy-constrained sensor networks. This ACO technique is also useful in VLSI and some embedded techniques where an efficient way of routing will help the device to consume less power, and as well as it will also save time during the traveling. So, we conclude that ACO worked properly to find an efficient path in the wireless sensor Network, and this work can be progressed in the future as well.

References

[1] P. R. McMullen, "Ant-Colony Optimization for the System Reliability Problem with Quantity Discounts," *American Journal of Operations Research*, vol. 07, no. 02, pp. 99–112, 2017, doi: 10.4236/ajor.2017.72007.

[2] M. Suyal and P. Goyal, "A Review on Analysis of K-Nearest Neighbor Classification Machine Learning Algorithms based on Supervised Learning," Jul. 01, 2022, *Seventh Sense Research Group*. doi: 10.14445/22315381/IJETT-V70I7P205.

[3] A. Jones, "Particle Swarm Optimization," 2023, doi: 10.13140/RG.2.2.32162.20163.

[4] 'Machine Learning Algorithms from Wireless Sensor Network's Perspective', Wireless Sensor Networks - Design, Applications and Challenges. IntechOpen, Oct. 18, 2023. doi: 10.5772/intechopen.111417. R. Chandra Gangwar and R. Singh, "Machine Learning Algorithms from Wireless Sensor Network's Perspective," *Wireless Sensor Network's-Design Applications and Challenges*, Oct. 2023.

[5] F. Zhang *et al.*, "A Benchmark Of Transparent Data Encryption For Migration Of Web Applications In The Cloud," *Journal of Computer Engineering & Information Technology Short*

Communication A SCITECHNOL JOURNAL All articles published in Journal of Computer Engineering, vol. 34, no. 1, pp. 121–128, 2012, doi: 10.4172/2324-9307.1000326.

[6] J. M. Chalissery, A. Renyard, R. Gries, D. Hoefele, S. K. Alamsetti, and G. Gries, "Ants sense, and follow, trail pheromones of ant community members," *Insects*, vol. 10, no. 11, Nov. 2019, doi: 10.3390/insects10110383.

[7] S. Okdem and D. Karaboga, "Routing in wireless sensor networks using an Ant Colony optimization (ACO) router chip," *Sensors*, vol. 9, no. 2, pp. 909–921, Feb. 2009, doi: 10.3390/s90200909.

[8] J. Revathy and S.K. Jayanthi, "A Diabetes Diagnosis Model Using Optimized Long Short-Term Memory Based On Improved Particle Swarm Optimization ," Erode, Dec. 2024.

[9] K. Mahawar *et al.*, "Employing artificial bee and ant colony optimization in machine learning techniques as a cognitive neuroscience tool," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-94642-6.

[10] Z. Mostafa *et al.*, "Characterizing QoS Parameters and Application of Soft-QoS Scheme for 3G Wireless Networks."

[11] T. Bhowmik and I. Banerjee, "An Improved ACO Based Energy Efficient Routing Algorithm in WSNs," Coimbatore: IEEE, 2018, pp. 1–6.

[12] A. Ahuja, S. Das, and A. Pahwa, "An AIS-ACO hybrid approach for multi-objective distribution system reconfiguration," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy*, Pittsburgh: IEEE, 2008, pp. 1–1.

[13] N. Singla and N. Sharma, "A Dynamic Network Reconstruction Approach using ACO." [Online]. Available: www.iosrjournals.orgwww.iosrjournals.org

[14] V. Maniezzo, L. Maria Gambardella, and F. de Luigi, "5 Ant Colony Optimization."

[15] Yinglin. Wang, Xuelong. Li, and Hongming. Cai, *PIC 2014 : proceedings of 2014 IEEE International Conference on Progress in Informatics and Computing : May 16-18, 2014 : Shanghai, China*. IEEE, 2014.

[16] GeeksforGeeks, "Introduction to Ant Colony Optimization."

[17] W. Zhu, W. Cai, and H. Kong, "Optimal path planning based on ACO in intelligent transportation," *International Journal of Cognitive Computing in Engineering*, vol. 6, pp. 441–450, Dec. 2025, doi: 10.1016/j.ijcce.2025.02.006.

[18] K. Mahawar *et al.*, "Employing artificial bee and ant colony optimization in machine learning techniques as a cognitive neuroscience tool," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-94642-6.

[19] B. A. Livingston, S. Pichler, E. E. Kossek, R. J. Thompson, and T. Bodner, "An Alpha, Beta and Gamma Approach to Evaluating Occupational Health Organizational Interventions: Learning from the Measurement of Work-Family Conflict Change," *Occup Health Sci*, vol. 6, no. 4, pp. 513–543, Dec. 2022, doi: 10.1007/s41542-022-00122-y.

[20] S. Zhang, X. Liu, and M. Wang, "A novel Ant colony optimization algorithm for the shortest-path problem in traffic networks," *Filomat*, vol. 32, no. 5, pp. 1619–1628, 2018, doi: 10.2298/FIL1805619Z.

[21] Q. Song, L. Yu, S. Li, N. Hanajima, X. Zhang, and R. Pu, "Energy Dispatching Based on an Improved PSO-ACO Algorithm," *International Journal of Intelligent Systems*, vol. 2023, 2023, doi: 10.1155/2023/3160184.

[22] Wikipedia - The Free Encyclopedia, "Ant colony optimization algorithms."

[23] R. Seidlová and J. Poživil, "IMPLEMENTATION OF ANT COLONY ALGORITHMS IN MATLAB."

[24] X. Wei, "Parameters Analysis for Basic Ant Colony Optimization Algorithm in TSP," *International Journal of u-and e-Service*, vol. 7, no. 4, pp. 159–170, 2014, doi: 10.14257/ijunnesst.2014.7.4.16.

[25] D. DeSantis, B. D. James, C. Houchins, G. Saur, and M. Lyubovsky, "Cost of long-distance energy transmission by different carriers," *iScience*, vol. 24, no. 12, Dec. 2021, doi: 10.1016/j.isci.2021.103495.

[26] M. Erdélyi, J. Abonyi, M. Kutatók, and N. Szimpóziuma, "Node Similarity-based Graph Clustering and Visualization," 2006. [Online]. Available: https://www.researchgate.net/publication/228920373

[27] H. Gunigari and S. Chitra, "Energy Efficient Networks Using Ant Colony Optimization with Game Theory Clustering," *Intelligent Automation and Soft Computing*, vol. 35, no. 3, pp. 3557–3571, 2023, doi: 10.32604/iasc.2023.029155.

[28] 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," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jul. 2020. doi: 10.1088/1742-6596/1566/1/012045.

[29] S. Haldenbilen, O. Baskan, and C. Oz, *An Ant Colony Optimization Algorithm for Area Traffic Control*. 2013.

[30] T. Sachithraj and P. Sharan, "COMPARISON OF DIJKSTRA ALGORITHM WITH ANT COLONY OPTIMIZATION ALGORITHM USING RANDOM TOPOLOGY IN ALL-OPTICAL NETWORK USING RWA." [Online]. Available: http://www.ijret.org

[31] J. Shalf, "The future of computing beyond Moore's Law," Mar. 06, 2020, *Royal Society Publishing*. doi: 10.1098/rsta.2019.0061.

[32] S.D. Saranya and M. Balasingh Moses, "Enhancing Economic Operation and Planning of Power Systems through Artificial Intelligence: Implementation of Optimal Power Flow in Chennai Utility Bus System," Jul. 2024.