Improving Energy Efficiency and Reliability by Optimizing Gateway Placement in Dense LoRaWAN Networks

Jui Mhatre[1], Ahyoung Lee[1*], Hoseon Lee[2], Tu N. Nguyen[1], and Kun Suo[1]
[1]Department of Computer Science, Kennesaw State University, Marietta, GA 30060, USA.
[2]Department of Electrical and Computer Engineering, Kennesaw State University, Marietta, GA 30060, USA
Email: [1]jmhatre1@students.kennesaw.edu, {[1*]ahyoung.lee, [2]hoseon.lee, [1]tu.nguyen, and [1]ksuo}@kennesaw.edu.
([*]Corresponding Author)

***Abstract*—Energy efficiency has become a top priority for NextG networks, driven by the need to reduce costs, comply with regulations, and mitigate the environmental impact of telecommunications infrastructure. LoRaWAN (Long Range Wide Area Network) plays a significant role in NextG networks due to its unique features involving low power consumption, cost-effectiveness, and broader coverage. As technology advances, there is an exponential rise in network traffic, resulting in increased interference within networks and reduced coverage. By strategically positioning gateways, network operators can ensure that the network meets the requirements of various applications and environmental conditions while managing deployment costs effectively. This paper presents two algorithms to improve the gateway placement strategy in LoRaWAN. The First uses the analysis of the connected components (CC-Place), and the second is based on Simulated annealing (SA-Place). We show that these approaches are energy-efficient for denser networks and those with broader coverage. Moreover, our results also show that our proposed algorithms make these networks more reliable by reducing the collision probabilities. We compare these algorithms with Graph-based gateway placement in [1] and intelligent, fuzzy C-Means (FCM) cluster-based algorithm in [2]. We evaluate the number of gateways each algorithm uses and their collision probabilities. We show that our algorithms propose fewer gateways, thus lowering the initial network cost.**

***Index Terms*—LoRaWAN, Gateway Placement, Energy efficiency, Connected Component Analysis, Simulated Annealing, Reliability.**

I. Introduction

LORAWAN (Long Range Wide Area Network) technology has gained significant traction in various IoT applications due to its long-range communication capabilities and low power consumption. LoRa is used widely in IoT owing to its low power consumption attribute. LoRa uses a chirp spread spectrum, making it robust to interference. LoRaWAN plays a crucial role in next-gen networks by providing low-power, long-range connectivity for IoT devices, supporting diverse applications, and enabling cost-effective and scalable network deployments. Its unique features make it well-suited for various use cases and contribute to next-gen networks' overall success and growth. LoRaWAN gateways (GW) are used to extend the range of LoRaWAN networks, which are designed to have a long-range capability. The Deployment of the GW which serve as communication hubs in LoRaWAN networks, is crucial for ensuring network coverage, reliability, and scalability.

Gateway placement is important for extending the coverage range of LoRaWAN networks. Since LoRaWAN operates in unlicensed spectrum bands, the number and placement of gateways significantly affect the network's coverage area. By strategically deploying gateways in optimal locations, coverage can be maximized, ensuring that sensor nodes located in remote or inaccessible areas can effectively communicate with the network. In dense urban environments or areas with a high concentration of devices, collisions can occur due to overlapping transmissions, leading to degraded network performance. Proper gateway placement helps alleviate this issue by ensuring that nearby gateways adequately serve nodes, reducing the likelihood of collisions and improving overall network capacity. Furthermore, energy efficiency is critical in LoRaWAN deployments, especially for battery-operated sensor nodes. Suboptimal gateway placement can increase energy consumption among nodes due to longer communication distances or inefficient routing paths. Energy consumption can be reduced by strategically placing gateways to minimize communication distances and optimize routing paths, thereby extending the sensor nodes' battery life and enhancing the network's sustainability. The optimal gateway placement in LoRaWAN networks is essential for maximizing coverage, minimizing collisions, and improving energy efficiency. This paper underscores the importance of considering various factors, including coverage requirements, network density, and energy constraints when designing and deploying LoRaWAN gateways to ensure the reliable and efficient operation of IoT applications.

This paper proposes a gateway placement algorithm for LoRaWAN, which finds an optimal number of gateways and their location coordinates. This placement reduces the number of gateways to be installed, reduces capital expenditure (CAPEX), and reduces the number of collisions. This helps reduce the power consumed to retransmit the packets and increases the network's reliability. The solutions involve finding an appropriate number of clusters and placing the GWs in the clusters. All the devices in the given cluster can access the GW and vice versa. We have two solutions: The first is a heuristic approach using Connected Components Analysis Placement (CC-Place), and the second uses a machine learning-based Simulated Annealing [3] technique based Placement (SA-Place). We also compare the existing state-of-the-art algorithms in [1], [2].

The rest of the paper is organized as follows: Section II describes the detailed study and analysis of the LoRAWAN gateway placement problem. Section III describes a model of the network under consideration, and we formulate the problem to solve, followed by the CC-Place algorithm and SA-Place in section IV. Analysis of results is discussed in section V followed by conclusion in VI.

II. Related Work

We studied various existing approaches to find optimal gateways and their locations. Graph-based gateway placement for better performance in LoRaWAN deployments [1], where a graph is created, and distances are found. The nodes with the highest centrality are chosen as a gateway, and choosing continues until the coverage is achieved using the shortest path algorithm. This paper presents a graph-based gateway placement approach to reduce collision probability and increase reliability in LoRaWAN deployments. It reduces the required number of gateways by up to 40% and collision probability by up to 70%. LoRaWAN gateway placement model for dynamic IoT scenarios (DPLACE) in [4] computes the number of LoRaWAN gateways based on the gap statistics method. A heuristic algorithm for gateway location selection in large-scale LoRa networks is proposed in [5].

Optimal gateway placement based on fuzzy C-Means for low power wide area networks in [2]. It considers the gap statistics method to find the number of LoRa gateway. The paper proposes an optimal LoRa gateway placement algorithm called PLACE, which reduces CAPEX and OPEX by 36% compared to grid and random placement methods. Reference [6] has four clustering algorithms that were used to deploy The network GWs are K-Means; its three versions are Minibatch K-Means, Bisecting K-Means, and Fuzzy c-Means (FCM). As robust gateway placement for scalable LoRaWAN [7] gateway positions for a LoRaWAN as a GEOMETRIC SET COVER problem where the sensors are points that need to be covered, and the gateways are unit disks whose radius equals the sending range of a gateway. It optimizes by converting into integer linear programming. An ILP solution for VORONOI COVER minimizes the number of chosen gateways. Methodology for LoRa gateway placement based on bio-inspired algorithms for a smart campus in a wooded area [8] uses evolutionary particle swarm optimization of gateways using integer linear programming.

In [6], the paper proposes a strategy for planning LoRaWAN gateways' (GWs) number and location in smart agriculture. It compares the performance of four clustering algorithms and suggests the optimal number of GWs for different scenarios. It considers clustering-based placement of LoRaWAN GWs and other solutions will have different results. The efficacy of Particle Swarm Optimization (PSO) for gateway placement in LoRaWAN networks is investigated in [9]. It proposes a modified PSO approach incorporating gateway distancing measures to achieve optimal locations for gateways. A comparative study of gateway positioning strategies in a LoRaWAN network is done in [10], but it does not provide a specific answer to the question about LoRaWAN gateway placement. Reference [8] proposes an empirical and statistical methodology using Evolutionary Particle Swarm Optimization (EPSO) to optimize the placement of LoRa gateways in a smart campus area. Another reference, [11], discusses the optimization of multiple gateway location selection in LoRaWAN networks, but it does not provide specific details on the placement of LoRaWAN gateways. The LINGO modeling program is used to test the model, and the findings suggest that six gateways at optimal locations can provide signal coverage for all end nodes and effectively manage the capacity of the LoRaWAN gateway. The paper proposes a mathematical model to optimize the selection of multiple gateway locations for LoRaWAN networks, considering factors such as the spatial distribution of clients, radio signal propagation, and the capacity of the gateways. The discussion of finding a sufficient number of sink nodes (gateways) for connectivity in a systematic grid distribution of outdoor sensor nodes in [12]. The study analyzes the performance of LoRa shadowed radio links operating in urban and semi-urban centers, incorporating an examination of node grid distributions and a determination of optimal sink node placements. The optimal placement of IoT gateways at the University of Zululand's main campus, but it does not specifically mention LoRaWAN gateway placement in [13]. Chosen gateway placement methods improve network performance. Results show that they are better than random placement for optimizing gateway placement in IoT networks.

We identified some challenges in existing work based on the above literature survey. Most of the papers present clustering-based solutions. They are not immune to noisy data and outliers, but in the case of networks, even the outlier sensor should be connected to achieve total coverage. Moreover, energy efficiency in LoRaWAN has not been discussed. In [6], they discuss energy efficiency due to higher interference, and collision is shown here. However, they don't consider other transmission parameters, which are also responsible for differences in power consumption. Our proposed approach compares collision probabilities of state-of-the-art techniques and spread factor allocations. Also, we present evaluations of the number of gateways used; this gives an idea about the initial installation cost for network operators. Spread factor allocation and energy efficiency are also discussed.

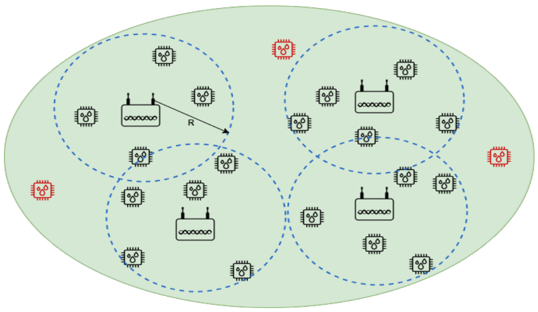


Fig. 1: LoRa Network with sensors connected to gateways. If sensors are out of gateway range, then they are disconnected from the entire system (e.g., the red color sensors).

III. System Model

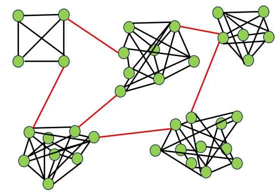
LoRaWAN is a star of star networks with N sensors, as shown in Fig. 1. Each sensor is connected to a LoRaWAN gateway. There are M gateways that are placed to cover all sensors in the network. The GW are further connected to the backhaul network. All GW can connect sensor in distance of R. All sensor within the gateway range of R can connect to it. Any sensor that is out of the gateway's range will not be able to connect to the network unless it is within the range of another gateway. During the uplink process, when sensors send data to the gateway, the data packet is retransmitted if there are collisions, thus increasing the power consumption. Network Operators (NO) aim to design the network with full coverage lower power consumption. The gateway placement problem is finding the minimum number of gateways and their appropriate locations to achieve full coverage. We aim to optimize the solution to this problem by using the minimum number of gateways to achieve total coverage and lower the collision probabilities. Additionally, locations where the gateways are installed are crucial. Signals that are modulated with a higher spreading factor are less prone to errors and can travel longer distances than signals with a lower spreading factor. Therefore, sensors located closer to the gateways do not require higher spreading factors. Thus, the appropriate gateway placement helps lower the energy consumption.

IV. Proposed solution

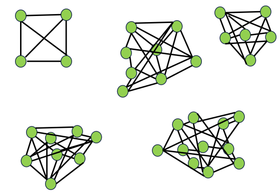
To find the minimum number of gateways and their locations, we propose two solutions: first, a heuristic algorithm based on connected component analysis (CCA), and second uses, a simulated annealing algorithm (SA).

A. Connected Component Analysis based Gateway Placement(CC-Place)

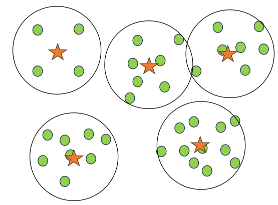
The gateway placement algorithm based on Connected Component Analysis (CC-Place) in Algorithm 1 can be visualized as in Fig. 2. The network with sensors forms the vertices V of a graph G. Gateways that we intend to place have a predefined range of R, and only the sensors in that range can connect to it. If the distance between two sensors



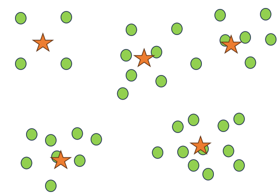
(a) Graph showing the sensors in the network.



(b) Connected components formed from the graph.



(c) Clustering and centroid detection of CC.



(d) Gateway Placement at the centroids.

Fig. 2: Comparison of the number of gateways and collision probability during S1 using CC-Place and Shortest Path-based Gateway placement [1] with their SF distributions.

$d(m, n)$ is less than or equal to the 2*R, an edge e_{mn} exists in the set E. Since the path loss of the signal directly relates to the distance traveled by the signal, signal strength is highly correlated to the distance. Hence, we have considered the distance a measure of the neighborhood. Fig. 2a shows a connected graph with all possible edges and all sensors as the vertices in our network. Next, we find connected clusters as described and shown in the second section (Fig. 2b). The function on line 11 ($connectedComponentAnalysis()$) generates components.

Algorithm 1 Connected Component Analysis for Gateway Placement (CC-Place)

1: $V \leftarrow$ Sensor Locations.
2: $E \leftarrow$ generateEdges(R)
3: $CC \leftarrow$ connectedComponentAnalysis(V, E, R)
4: **for** $c\ gets\ CC$ **do**
5: $\quad X \leftarrow CC$[:0]
6: $\quad Y \leftarrow CC$[:1]
7: $\quad$ centroidsX.append(sum(X)/len(X))
8: $\quad$ centroidsY.append(sum(Y)/len(Y))
9: $\quad$ print(centroidsX[-1], centroidsY[-1])
10: **end for**
11: func: connectedComponentAnalysis(V, E, R):
12: visited $\leftarrow$ []
13: cc $\leftarrow$ []
14: **for** v $\leftarrow$ range(count(V)) **do**
15: $\quad$ **if** visited[v] **then**
16: $\quad\quad$ component $\leftarrow$ []
17: $\quad\quad$ cc.append(DFS(component, v, visited))
18: $\quad\quad$ return cc
19: $\quad$ **end if**
20: **end for**
21: return component
22: return True;

It uses a depth-first search (DFS) algorithm to traverse the vertices internally. Additionally, we check that the component size is not greater than $Max_{Sensors}$ to ensure that the number of sensors in the cluster is within the load capacity that the GW can handle. We also check whether the sensor is within the range of the gateway, assuming that the gateway is placed at the cluster's centroid. When adding a new sensor to a component, it should be connected to all other sensors, meaning that it should be at most $2{*}R$ apart from all sensors. Once the components are selected, they form a cluster of sensors sharing the same gateway. The next task is to find the optimal location of the gateway. We choose to place the gateway at the centroid of the component (Fig. 2c). The final location of gateways looks like shown in Fig. 2d. The next step is spread factor allocation to each sensor. This parameter allocation further controls the signal transmission, data rate, and collision probability. Thus, it affects the energy efficiency of the network transmission. Spread factor allocation is done in a heuristic approach as described in [1].

B. Simulated Annealing based Gateway Placement (SA-Place)

Algorithm 2 Simulated Annealing for Gateway Placement (SA-Place)

```
curr_centers ← initial random clusters
curr_labels ← cluster allocations to all data points based on current_centers
current_max_dist ← maximum distance (point, current_centers)
max_iterations ← num
for iter ← max_iterations do
    neigh_sol ← curr_solution(add/remove cluster)
    neigh_labels ← cluster allocations to all data points based on neigh_sol
    neigh_max_dist ← maximum distance (point, neigh_centers)
    delta_dist = neigh_max_dist - curr_max_dist
    if if delta_distance < 0 or rand() < math.exp(-delta_dist / current_temp) then
        curr_(centers,labels,max_dist) ← neigh_(centers, labels,max_dist)
        if current_max_dist < best_max_dist then
            curr_(centers,max_dist) ← best_(centers, max_dist)
        end if
        update curr_temp
    end if
    if best_max_dist ≤ max_dist_threshold then break
    end if
end forreturn , best_centers, best_max_dist
```

The simulated annealing process is a versatile optimization technique that can be applied to a wide range of problems where finding the global optimum is difficult due to the presence of multiple local optima or complex search spaces. We use it to find an optimal number of gateways and their locations. The main aim is to minimize the maximum distance distance between data points and cluster centers. This can be learnt in continuous manner and find cluster centers and updating them in each iteration to perform minimization. Details of the SA algorithm are mentioned in Algorithm 2. It is performed as follows:

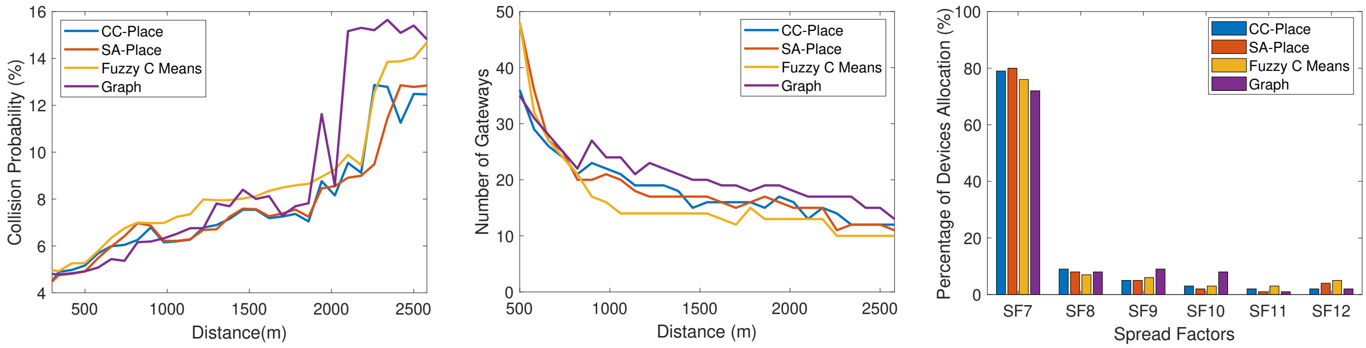
1) Initialize SA algorithm using parameters such as initial temperature (1.0), cooling rate (0.95), max_iterations (10,000).
2) Initialize the clusters using random numbers and create labels for all data based on the cluster centers. Find the max cluster distances based on currently selected ones.
3) Initialize the best solution as the current clusters.
4) Iteratively learn the new clusters as follows:
   - Intermediate or neighboring clusters are created by randomly adding or deleting one cluster from the current cluster set and finding the maximum distance for each point with new cluster centers.
   - If newly generated new clusters are nearer to data points, replace the current clusters with these new clusters and discard the previous ones.
   - If they are nearest to any current clusters, label them as the best ones.
   - This update takes place with acceptance probability. It calculates the probability of accepting a worse solution (neighbor solution) based on the criterion. This probability is used in simulated annealing to determine whether to accept or reject the neighbor solution. The algorithm agrees with the worse solution if the likelihood is greater than a random number between 0 and 1. The algorithm avoids getting stuck in suboptimal solutions by accepting moves that increase the objective function value, especially at high temperatures. The algorithm becomes less likely to receive worse solutions as the temperature decreases, leading it toward convergence.

V. Performance Evaluation

We evaluate the performance of our algorithms using the dataset from Würzburg, Germany, which comprises 10,000 sensors as described in [7]. Our evaluation is conducted in two scenarios, as discussed below. Performance evaluation is carried out for two scenarios,

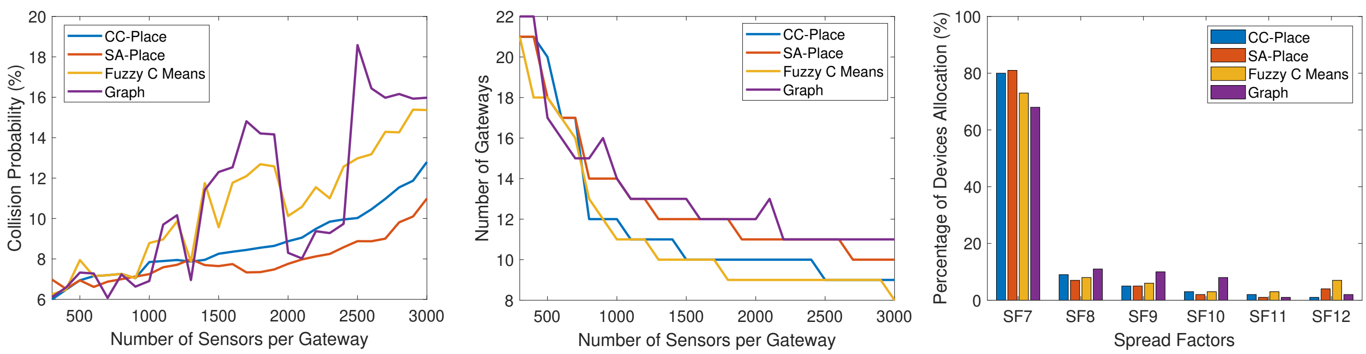
1) **Scenario $S1$:** We studied what should be the optimal distance between gateways and sensors because if they are larger, then larger SFs are required, which in turn causes a higher collision probability as ToA for each message increases.
2) **Scenario $S2$:** We increase the number of sensors and evaluate the collision probabilities and gateway counts selected by both algorithms.

We have compared the results of these scenarios using our CC-Place and SA-Place algorithms with a Graph-based shortest path algorithm [1], and clustering algorithm using Fuzzy C-Means [2]. Implementation can be found in [14]. We extracted results for the number of GWs with their exact locations in Fuzzy C- Means (FCM) using the implementation mentioned in [15]. The collision probability and SF allocation are performed using the implementation in [1].



(a) Collision Probability Evaluation (b) Number of Gateways Evaluation (c) SF Distribution Evaluation

Fig. 3: Comparison of the number of gateways, collision probability, and spread factor distribution for varying distance thresholds between the sensors and the gateways from 300m to 2,580m. The maximum number of connections to the gateway is kept constant at 300 sensors per gateway.



(a) Collision Probability Evaluation (b) Number of Gateways Evaluation (c) SF Distribution Evaluation

Fig. 4: Comparison of the number of gateways, collision probability, and spread factor distribution for varying thresholds of the number of sensors per gateway. The maximum range the gateway can transmit to is kept constant at 800m.

A. Scenario S1: Evaluation of network for increasing the maximum allowed distance between sensors and gateways.

We evaluated the results for Scenario $S1$ in Fig. 3. It shows a comparative analysis for finding an optimal number of gateways and locations for increasing the distance between sensors and gateways using the CC-Place, SA-Place, FCM, and graph-based shortest path algorithms. The predicted gateways should be able to transmit up to given distances. All sensors are part of at least one gateway's scope.

Fig. 3a shows the collisions that take place in the network when the number of gateways and their locations are predicted by algorithms in Fig. 3b for increasing the maximum distance of sensors from the gateways that they can serve. We observe that the FCM algorithm predicts fewer gateways for a given distance from sensors than other algorithms. But, it faces higher collisions when compared to CC-Place and SA-Place. This is because the gateways are sparse for FCM. Most of the sensors can connect to one GW. However, since the GW count increases, sensors can be connected to more than one gateway for CC-Place and SA-Place. Thus, it increases the spectral availability and reduces collisions. For the Graph algo., the gateway count and the collision probabilities are higher because the gateway location chosen is one among the sensors. This may not always be the optimal location, when the sensor is an outlier and farther than all other sensors; this makes it farther from the gateway than CC-Place, where the location chosen is a centroid of the sensor group in the cluster.

Fig. 3c shows spread factor distribution in sensors based on distances from the gateway. This allocation is mainly dependent on the distance. The lower the distance between the and the gateway, the lower the spread factor is allocated. In CC-Place and SA-Place, the average distance between the gateway is lower than in FCM and graph-based algorithms. The gateway locations control spread factor distribution. The more optimally placed the gateway, the lower the spread factors allocated. Thus, we observe that SA-Place and CC-Place have more sensors allocated to SF7. However, if the sensor is beyond a distance that the signal with SF7 cannot reach, it becomes essential to allocate SFs with values greater than SF7. Lower SF sends more chirps per second, allowing faster data encoding. Higher SF means fewer chirps per second, leading to less data encoded. Higher SF requires more transmission time (airtime) when sending the same amount of data. Longer airtime means the modem consumes more energy [16]. We observe that lower spread factors are allocated to sensors in a CC-Place-based and SA-Place algorithm-based network as compared to the Shortest path and FCM. This lowers the energy consumption due to gateway placement using our proposed algorithms. Thus, we can safely say that energy efficiency is paramount for the network with gateways that can connect to farther sensors. Cc-place and SA-Place algorithms should be considered for GW placement.

B. Scenario S2: Performance evaluation of network for increasing the maximum allowed sensors per gateway.

The evaluation results for scenario $S2$ are shown in Fig. 4. It shows how the collisions are affected (Fig. 4a) when the number of GWs is predicted using each algorithm. We show the predicted number of GWs for increasing the number of sensors served per gateway in Fig. 4b. Thus, the algorithms place GW to serve sensors that are not more than the specified threshold. We observe the improvement as fewer GWs are predicted in SA-Place and CC-Place than in the graph-based algorithm. FCM proposes the lowest gateways to serve the sensors in the network. This implies that more sensors per gateway are served, increasing collisions. This is observed in Fig. 4a. SA-Place assigns more gateways than FCM but maintains lower collision probabilities. This is because the number of sensors assigned to each gateway is lower than the FCM. The CC-Place shows improved collision probability compared to FCM, but the number of gateways is comparable to FCM and is the lowest among all algorithms under consideration. This improvement is also observed due to the location of gateways, which the algorithm predicts. Thus, for scenarios with a dense network with each GW serving more sensors, the SA-Place algorithm should be used for GW placement if collision probability is the primary concern. However, if CAPEX is to be lowered by reducing the number of GW, then the CC-Place algorithm proves more useful. Moreover, spread factor distribution in Fig. 4c shows that gateways are placed optimally for denser networks using CC-Place and SA-Place by more sensors using lower spread factors. Since a higher number of sensors transmit using lower spread factors, the overall energy consumption of the network is reduced.

VI. Conclusion

Gateway placement is essential in the network design step. The algorithms help assign and place GW in appropriate locations based on the network requirements and characteristics. This paper proposes the connected component analysis-based (CC-Place) and simulated annealing-based (SA-Place) algorithms to find the optimal number of gateways and locations in the network. We compare our work with the existing heuristic graph-based approach in [1] and the clustering-based fuzzy c-means algorithm in [2]. We evaluate GW placements for the networks in two scenarios: Based on the strength of GW signals to farther distances and for dense networks when the GW needs to support the load from more sensors. We show that our proposed algorithms are energy efficient in transmitting farther distances. They control the GW count as well as the number of collisions. Similarly, we also show that SA-Place is a reliable solution for denser networks, as it has lower collision probabilities, and the CC-Place algorithm can lower the initial network setup cost by installing a reduce GW.

VII. Acknowledgement

This research was partly supported by the US NSF Grants SHF-2210744, CPS-2103459, AMPS-2229073, and CNS-2103405.

References

[1] F. Loh, N. Mehling, S. Geißler, and T. Hoßfeld, "Graph-based gateway placement for better performance in lorawan deployments," in *2022 20th Mediterranean Communication and Computer Networking Conference (MedComNet)*. IEEE, 2022, pp. 190–199.

[2] N. Matni, J. Moraes, D. Rosário, E. Cerqueira, and A. Neto, "Optimal gateway placement based on fuzzy c-means for low power wide area networks," in *2019 IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, 2019, pp. 1–6.

[3] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[4] N. Matni, J. Moraes, H. Oliveira, D. Rosário, and E. Cerqueira, "Lorawan gateway placement model for dynamic internet of things scenarios," *Sensors*, vol. 20, no. 15, p. 4336, 2020.

[5] K. Grochla and K. Połys, "Heuristic algorithm for gateway location selection in large scale lora networks," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2020, pp. 777–782.

[6] F. P. Correia, S. R. d. Silva, F. B. S. d. Carvalho, M. S. d. Alencar, K. D. R. Assis, and R. M. Bacurau, "Lorawan gateway placement in smart agriculture: An analysis of clustering algorithms and performance metrics," *Energies*, vol. 16, no. 5, p. 2356, 2023.

[7] F. Loh, D. Bau, J. Zink, A. Wolff, and T. Hoßfeld, "Robust gateway placement for scalable lorawan," in *2021 13th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, 2021, pp. 71–78.

[8] H. A. Cruz, S. C. Ferreira, J. P. Araújo, F. J. Barros, F. S. Farias, M. C. Neto, M. E. Tostes, A. A. Nascimento, and G. P. Cavalcante, "Methodology for lora gateway placement based on bio-inspired algorithmsfor a smart campus in wooded area," *Sensors*, vol. 22, no. 17, p. 6492, 2022.

[9] C. N. Nyirenda, "On the efficacy of particle swarm optimization for gateway placement in lorawan networks," 2021.

[10] C. N. da Silva, P. F. F. de Abreu, J. D. da Silveira *et al.*, "Estimating the number of gateways through placement strategies in a lorawan network," IEEE, pp. 1–6, 2022.

[11] C. Kaewta, C. Savithi, and E. Naenudorn, "An optimization of multiple gateway location selection in long range wide area network networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 2, pp. 1011–1020, 2023.

[12] N. B. Zainal, M. H. Habaebi, I. Chowdhury, M. Islam, and J. I. Daoud, "Gateway sink placement for sensor node grid distribution in lora smart city networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 2, pp. 834–842, 2019.

[13] S. Mnguni, P. Mudali, N. Sibeko, and M. O. Adigun, "Lora gateway placement at the university of zululand: A case study," in *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*. IEEE, 2019, pp. 1–6.

[14] J. Mhatre. Cc-place. (Date last accessed:2023). [Online]. Available: https://github.com/JuiMhatre/lora_graph_gw

[15] G. LaCIS. Cmeans_gw_allocation. (Date last accessed:2023). [Online]. Available: https://gitlab.com/gercomlacis/cea/lorawan/Cmeans_GW_Allocation

[16] Q. T. T. C. Partner. How spreading factor affects lorawan® device battery life. 18-11-2019. [Online]. Available: https://www.thethingsnetwork.org/article/how-spreading-factor-affects-lorawan-device-battery-life