# Reducing computational complexity of brute force algorithm in solving optimal placement of directional antenna

Aye Min Thike, Sergey Lupin

***Abstract*—Determining the optimal placement of directional antennas in wireless networks is a challenging problem, which includes maximum coverage with minimal overlap. This study explores some reduction methods to improve the efficiency of the brute force algorithm for determining this problem. We propose and evaluate three reduction methods: restriction of antenna rotation angles, limitation of potential positions of directional antennas, and early termination of unsatisfactory calculations. In our computational experiments, we apply these methods separately, in combination, and alongside a parallel computing approach with a brute force algorithm for calculating maximum coverage with minimal overlap. The parallel computing approach improves the brute force algorithm's performance using multi-core processing, while the combined reduction methods significantly improve efficiency by minimizing redundant and inefficient computations. Experimental results confirm that these methods successfully mitigate the limitations of the brute force algorithm, effectively reducing the computation time compared to the conventional brute force algorithm and improving scalability for solving optimal directional antenna placement problems.**

***Keywords*— optimal antenna placement, directional antenna, maximum coverage, wireless network design, brute force algorithm, computational efficiency, parallel computing**

## I. INTRODUCTION

Wireless network design plays an essential role in modern communication systems, as the performance of these systems depends on the strategic placement of antennas. Among the various factors influencing network efficiency, the optimal placement of directional antennas is especially crucial, as it directly impacts coverage, signal quality, and interference [1]. However, optimizing the placement of directional antennas is particularly complex due to their unique characteristics, including beam width, orientation angles, and positioning constraints. Unlike omnidirectional antennas, directional antennas reduce interference [2] and can transmit signals over longer distances, which is particularly beneficial in rural or remote areas where network nodes are often widely spaced.

Determining the optimal placement of directional antennas involves identifying the best locations and orientations to maximize coverage area while minimizing overlapping regions.

As the deployment area expands and the number of antennas increases, finding optimal placements [3] becomes increasingly complex.

Addressing this challenge requires advanced algorithms and efficient reduction methods to reduce the extensive search space within practical computation time. The optimal placement of directional antennas has critical applications in various fields, including IoT networks, military communication systems, cellular networks, and satellite communications [4], where maximizing coverage efficiency while minimizing interference is essential.

Urban networks require minimizing interference in densely populated areas, while sensor networks focus on maximizing coverage with limited resources. For Internet of Things (IoT) [5] networks, which involve many devices operating in varied environments, optimal antenna placement is essential to ensure reliable communication and minimize interference. These diverse requirements highlight the necessity for adaptable, computationally efficient placement algorithms.

Various algorithms have been developed to address the challenge of optimal placement of directional antennas, aiming to maximize coverage while minimizing overlap. Genetic Algorithms (GA) [6] simulate natural evolutionary processes, refining antenna positions through selection, crossover, and mutation. It allows them to explore large search spaces and identify near-optimal solutions. Simulated Annealing (SA) [7] uses a probabilistic approach, accepting suboptimal solutions temporarily to escape local minima, making it particularly effective for complex, non-linear optimization problems and finding global optima. Particle Swarm Optimization (PSO) [8] utilizes a swarm of particles that adjust their positions based on individual and collective experiences, enabling efficient exploration of high-dimensional solution spaces. Each algorithm offers unique advantages, providing effective placement for optimizing wireless network design.

However, existing methods often fail to balance computational complexity and placement accuracy. While heuristic and metaheuristic approaches provide approximate solutions, they lack the deterministic precision of brute force algorithm (BFA). The conventional brute force algorithm [9], which evaluates all possible placements and rotation combinations, provides exact solutions but becomes computationally impractical for larger deployment areas.

With the millions of potential positions and 360-degree rotations for each antenna, the computational complexity of the optimal placement problem grows exponentially, making brute force methods impractical for real-world applications.

This study introduces three reduction methods to decrease the computational complexity of the brute force algorithm for calculating optimal placement while maintaining its exhaustive search capabilities. Complexity reduction methods are especially crucial for real-time and large-scale deployments, where achieving speed and precision is essential. Therefore, developing reduction methods that reduce computational complexity without compromising accuracy is imperative.

Parallel computing [10] effectively provides the limitations of brute-force algorithm by utilizing multi-core processors to execute complex calculations simultaneously. This approach [11] is especially advantageous for problems involving numerous antennas and large grid sizes, as it distributes the computational workload across multiple processors, significantly reducing computation times. By leveraging modern computational architectures, this method achieves substantially faster performance than the sequential execution of a brute force algorithm.

A key aspect of evaluating the quality of antenna placement involves calculating critical functions, such as maximum coverage and minimum overlap, which often involve complex geometric assessments. This paper presents reduction methods to improve the efficiency of the brute force algorithm in solving the optimal placement of directional antennas. Implementing these reduction methods has reduced computation times and resource usage while maintaining accuracy, making them crucial for real-time and large-scale deployments where speed and precision are essential.

## II. Related works

Optimizing the placement of directional antennas in wireless networks is a complex problem that has garnered significant research attention. This paper [12] discusses using a brute force algorithm to determine the optimal placement of antennas to maximize population coverage. Solving the computational intensity of brute force methods, the authors incorporate parallel computing techniques using OpenMP, significantly improving the efficiency of algorithm practicality for moderately sized networks. The paper highlights its contributions by presenting a clear mathematical framework and a replicable step-by-step methodology. While the algorithm demonstrates effectiveness, its scalability for extensive grid-based networks remains a notable limitation.

The authors [13] explore the application of a brute-force algorithm (BFA) to determine the optimal placement of omnidirectional antennas. A critical objective in optimizing wireless networks is maximizing the coverage area. BFA is used for solving discrete optimization, as it guarantees exact solutions by exhaustively evaluating all possible configurations. However, its computational complexity depends on the size of the solution space and the computational effort required to calculate criterion functions. Experimental results demonstrate that the tabular method accelerates the process, making it 17 times faster than directly computing the criterion value, ensuring no overlap in reception areas. This tabular approach is also highly suitable for parallel BFA execution. By using 12 threads, the computation time was reduced by more than sixfold, significantly improving efficiency.

The paper [14] presents a method for accurately determining the coverage area in wireless sensor networks (WSNs). The proposed algorithm considers the connectivity and uncovered areas within a region of interest (RoI), employing a disk model to analyze overlapping sensor ranges. The method efficiently calculates intersection arcs, uncovered arcs, and their corresponding areas using mathematical constructs such as Heron’s formula and polygon-based approximations. The authors validate their approach through MATLAB simulations, demonstrating its accuracy in scenarios with varying node configurations, including cases with cavities or hidden nodes. A notable strength of the study is its systematic approach to addressing uncovered sensing areas, which is critical for applications in dynamic environments.

The paper addresses the optimization of directional antenna placement to improve bandwidth utilization and reduce interference in wireless networks. By proposing an Integer Linear Programming (ILP) model, the study minimizes the number of antennas required to satisfy user bandwidth demands across multiple base stations. The authors [15] explore critical factors like antenna orientation, coverage limits, and bandwidth constraints, providing insights into their relationship through simulation-based evaluations using Cplex’s branch-and-bound algorithm. Their results highlight the importance of geometric alignment and the impact of base station distribution on achieving optimal network efficiency.

The study [16] presents topology optimization for wireless networks using brute force algorithm to maximize coverage, signal strength, and cost-efficiency. The brute force algorithm evaluates all possible positions of wireless network elements, such as directional and omnidirectional antennas, and selects the optimal solution. By using Intel Xeon Phi coprocessors and parallel computing, the authors demonstrate that brute force algorithm, often deemed impractical due to their high computational demands, can now be effectively utilized for such optimization tasks. The study includes a mathematical formulation of the optimization problem, complemented by visual representations such as grid-based settlements. The results are displayed through detailed tables and scalability graphs, which show the effect of grid size and core utilization on computational load and processing times.

The authors [17] propose innovative graph-based gadgets to model complex constraints and develop algorithms for optimizing the network topology. The study focuses on optimizing the backbone topology in airborne networks using directional antennas. The objective is to maximize concurrent bandwidth from source nodes to a sink node while considering critical constraints such as radio compatibility, directional link capacities, and relay node placements. The authors demonstrate that uniform capacity can be efficiently solved using flow-based methods. However, non-uniform capacity presents NP-complete challenges, making them computationally intensive. To address this complexity, the authors introduce a heuristic iterative rounding algorithm, which provides practical scalability and achieves near-optimal solutions in simulations-based evaluations.

The paper [18] describes using a genetic algorithm (GA) to optimize the placement of base stations in cellular networks. The study focuses on maximizing signal coverage, minimizing interference, and reducing financial costs while adhering to constraints such as traffic demand and handover requirements. The authors employ the HATA propagation model and the weighted sum method to evaluate multiple objectives, demonstrating the flexibility of GA in multi-criteria optimization problems. Their approach dynamically adjusts base station parameters, including location, transmission power, and antennas tilt, to achieve optimal network performance. The proposed algorithm is validated through simulations conducted over a 20x20 km² area with 3450 test points. The results demonstrate significant improvements, achieving 97% coverage while ensuring efficient handover and minimized interference.

The paper [18] describes using a genetic algorithm (GA) to optimize the placement of base stations in cellular networks. The study focuses on maximizing signal coverage, minimizing interference, and reducing financial costs while adhering to constraints such as traffic demand and handover requirements. The authors employ the HATA propagation model and the weighted sum method to evaluate multiple objectives, demonstrating the flexibility of GA in multi-criteria optimization problems. Their approach dynamically adjusts base station parameters, including location, transmission power, height, and antenna tilt, to achieve optimal network performance. The proposed algorithm is tested through simulations conducted over a 20x20 km² area with 3450 test points. The results demonstrate significant improvements, achieving 97% coverage while ensuring efficient handover and minimizing interference.

This study [20] introduces a modified brute-force approach to optimize antenna placement in wireless networks. It reduces computational complexity by precomputing overlap areas and distances between antenna positions, storing them in a multidimensional matrix for quick access. This method retains the exhaustive nature of the brute force algorithm while significantly improving efficiency, achieving a 12-16x speedup over the original algorithm. The authors emphasize reducing redundant computations to address the NP-hard nature of the antenna placement problem and demonstrate the scalability of their approach for larger grids and configurations. The study provides a detailed mathematical foundation for overlap calculations and implements this method through C++ programming.

This paper addresses the challenge of optimizing directional antenna placement in wireless networks using a brute force algorithm enhanced with complexity reduction methods. While the brute force algorithm guarantees exact solutions, it suffers from high computational complexity. By implementing early termination of unsatisfactory calculations, restricting rotation angles and potential positions of directional antennas, and using parallel computing, we significantly reduce the computational burden of brute force algorithm for calculating critical functions.

## III. Problem condition

The problem involves determining the optimal placement of directional antennas on a (6×6) km grid while maximizing coverage and minimizing overlap. The grid consists of 36 cells, each measuring (1×1) km$^2$, allowing 36 possible placement options for each antenna. The numerical task includes three directional antennas ($N = 3$), each antenna with a 6 km range ($R$) and a 30° beam width ($\theta$). The brute force algorithm evaluates all possible placements and orientations, calculates coverage and overlap areas for each placement and orientation, and determines the optimal placement and orientation that achieves maximum coverage with minimal overlap.

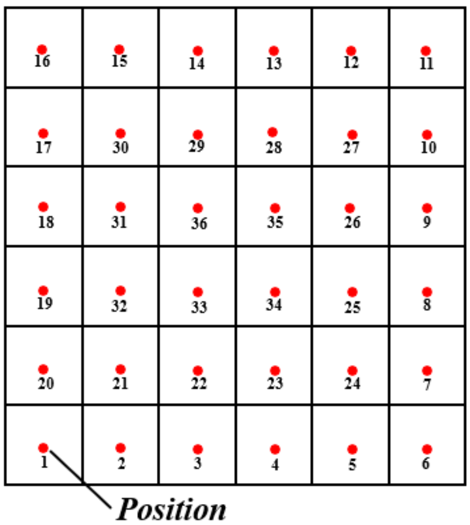


Fig. 1 Positons of antennas

The antenna is located at the center of a selected cell, and its orientation is adjusted to cover the desired area within the grid. Each antenna position on the grid is assigned by coordinates ($x$, $y$) and defined as a corresponding number to facilitate optimal placement determination. In Figure (1), the grid positions are sequentially numbered in a clockwise pattern, starting from the bottom-left corner at (x = 0, y = 0). The positions along the bottom edge (x = 0 to 6, y = 0) are numbered from 1 to 6. The positions along the right edge (x = 6, y = 1 to 6) are numbered from 7 to 11. The numbering continues along the top edge (x = 5 to 1, y = 6) from 12 to 16 and down the left edge (x = 0, y = 5 to 1) from 17 to 20. This numbering pattern repeats inward for subsequent layers, ensuring a systematic arrangement for placement evaluation.

Let $N$ denote the number of antennas, $M$ is the total number of possible positions for each antenna, and $\theta$ the beam width. Since the antennas rotate in increments of $\theta$ over 360°, the number of possible orientations per antenna is given by ($\boldsymbol{O} = \mathbf{360/}\ \boldsymbol{\theta}$). The total number of variants $V$ is determined as ($\boldsymbol{V=M^N * O^N}$) using the brute force algorithm, where ($M^N$) represents all possible position combinations and ($O^N$) represents all possible orientation combinations for $N$ antennas. This exhaustive approach evaluates every configuration to determine the optimal placement and orientation for maximum coverage with minimizing overlap.

## IV. Reducing the Computational Complexity of Brute Force Algorithm

Reduction methods are essential for improving computational efficiency in determining the optimal placement of directional antennas. The first approach is restricting the range of rotation angles based on antenna positions. In a conventional brute force method, antennas rotate through 360° at each placement to evaluate coverage, which is highly time-consuming. By considering the deployment area and antenna orientation, restricting rotations to a specific angular range decreases unnecessary evaluations. Instead of testing all possible angles, the algorithm focuses only on those contributing to effective coverage, significantly reducing computational complexity for coverage and overlap calculations. For example, antennas positioned along the bottom edge of the grid (x = 0 to 6, y = 0) only require rotations between 0° — 180° to cover the grid. This targeted approach reduces the search spaces, accelerates calculations, and enhances the efficiency of the placement process.

The second method focuses on strategic placement based on antenna range. Directional antennas with long-range coverage don’t need to be placed in every grid cell. Instead, their placements are selected depending on their coverage radius. For example, a 6 km range antenna can effectively cover a 6×6 grid or the entire area when positioned along the grid’s borders. Similarly, a 5 km range antenna is placed one cell inward from the edges to ensure coverage of the required area without unnecessary overlap. This method avoids ineffective calculations for coverage areas, is placed across multiple cells to maximize coverage area, and reduces the number of potential placements. As a result, it improves the computational efficiency of the brute force algorithm, optimizes the placement process, and achieves critical functions.

A third method improves efficiency by terminating unsatisfactory calculations early when identifying overlapping areas between antennas. The process begins by computing the coverage area and storing point numbers for the coverage area of the first antenna in an array. For other antennas, the algorithm calculates their coverage and checks whether any point numbers already exist in the array. If point numbers are the same, this means finding an overlapping area, the current subtask is immediately terminated, and the algorithm proceeds to the next potential placement. This early termination strategy reduces redundant calculations for coverage area and computational overhead. It improves the efficiency of the brute-force algorithm by minimizing unnecessary iterations, making it particularly beneficial for large-scale antenna placement scenarios.

Parallelization [21] is another approach to improve the efficiency of brute force algorithms by distributing coverage calculations across multiple processors. Parallel computing is applied to coverage area calculations for various antennas in this study, significantly reducing processing time for critical functions. This method is particularly effective in solving large-scale deployments when increasing the grid sizes and antenna numbers. By leveraging parallel processing, the approach improves scalability and efficiency in solving optimal placement of directional antenna.

Combining multiple reduction methods is the most effective to improve the computational efficiency of the conventional brute force algorithm without compromising the accuracy of optimal antenna placement. This integrated approach enhances scalability and performance, making it well-suited for solving complex wireless network design challenges in real-world applications.

## V. Implementation of Complexity Reduction Methods for Optimal Placement of Directional Antenna

Optimizing wireless network designs requires calculating critical functions, such as maximizing coverage while minimizing overlap. However, the brute force algorithm is commonly used for this task but has high computational complexity due to exhaustive searches and redundant calculations. Various complexity reduction methods are applied to solve these challenges, including early termination of unsatisfactory calculations, restricting rotation angles, and limiting potential positions of directional antennas. These methods effectively reduce the search space and eliminate redundant computations, significantly decreasing computation time without compromising accuracy. This study presents the implementation and evaluation of these methods, demonstrating their effectiveness in improving the computational efficiency of the brute force algorithm for optimal directional antenna placement.

For an antenna located at position $(x_i, y_i)$, the distance to a grid point (x, y) is calculated as:

$$d_i(x,y) = \sqrt{(x-x_i)^2 + (y-y_i)^2} \qquad (1)$$

Coverage area of points falls within the antenna's coverage range $R$, if:

$$d_i(x, y) \leq R \quad (2)$$

To reduce computational complexity, antenna orientations are limited based on their placements. The possible orientation range for an antenna is:

$$\theta_i = \{\, \theta : \theta_{min} \leq \theta \leq \theta_{max} \} \quad (3)$$

The coverage area $C_i$ of the ***i***-th antenna consists of the set of grid points (x,y) that satisfy its range and orientation constraints. Mathematically, this can be expressed as:

$$C_i = \{\, (x, y) | d_i(x, y) \leq R, \theta_{min} \leq \theta \leq \theta_{max} \} \quad (4)$$

Here:

$d_i(x, y) \leq R$ – ensures the coverage point is within the antenna's range.

$\theta_{min} \leq \theta \leq \theta_{max}$ – ensures the coverage point falls within the specified beam orientation.

Let $C_i^{(k)}$ represent the coverage area of antenna ***i***-th in a given subtask ***k***, defined by a specific placement and beam orientations. The total coverage for ***N*** antennas in subtask ***k*** is:

$$C_{total}^{(k)} = \bigcup_{i=1}^{N} C_i^{(k)} \quad (5)$$

Overlap occurs if a point is covered by more than one antenna:

$$\exists (x, y) \in C_i^{(k)} \cap C_j^{(k)} \text{ for i} \neq \text{j} \quad (6)$$

If overlap occurs, the variant ***k*** is discarded. The set of valid variants ***V***, where no overlap exists, is:

$$V = \{k | C_i^{(k)} \cap C_j^{(k)} = \emptyset \; \forall \text{ i} \neq \text{j}\} \quad (7)$$

Among valid configurations, the optimal subtask $F^{(k_{opt})}$ is selected as:

$$F^{(k_{opt})} = \arg\max_{k \in V} |C_{total}^{(k)}| \quad (8)$$

Where: $F^{(k_{opt})}$ represents optimal placement and beam orientation of the antennas.

The implementation of brute force algorithm calculates the critical function using complexity reduction methods as follows:

1. Initialize the grid size and parameters of antennas;
2. Generate all possible placement variants;
3. Generate subtasks for antenna beam orientations $\theta$;
4. For each subtask ***k***, calculate the coverage sets $C_i^{(k)}$ for all antennas ***i=1,2, …, N***;
5. Check for overlap using equation (6);
6. If overlap is found: discard subtask ***k*** and return to step (2); otherwise, retain valid variants and calculate critical functions;
7. Compare all valid variants and select the optimal solution $F^{(k_{opt})}$ using equation (8);
8. Output the optimal solution $F^{(k_{opt})}$.

These approaches improve efficiency by systematically filtering out invalid variants, reducing search space, and eliminating redundant calculations while selecting the optimal placement with minimal overlap and maximum coverage.

## VI. EXPERIMENTAL RESULTS

In this study, we implemented and tested complexity reduction methods using the C++ programming language in the Visual Studio environment, configured in 64-bit release mode. The experiments were conducted on a personal computer with an Intel Core i3 9th generation processor operating at a clock frequency of 3.1 GHz. The tested methods included restricting rotation angles (**C*RM*$_1$**), limiting potential placements of directional antennas (**C*RM*$_2$**), early termination of unsatisfactory calculations (**C*RM*$_3$**), parallel computing (**C*RM*$_4$**), and the conventional brute force approach. The experiments evaluated different test variants for computing critical functions: (1) conventional brute force, (2) **C*RM*$_3$** only, (3) **C*RM*$_2$** and **C*RM*$_3$**, (4) **C*RM*$_1$**, **C*RM*$_2$**, and **C*RM*$_3$**, and (5) **C*RM*$_4$**. All variants produced identical results, achieving a total of 34 coverage points. Table 1 presents the point number for optimal antenna placements, beam orientations, total coverage points, and computation times for each test variant, highlighting significant differences in performance.

Table 1. Experimental results using various complexity reduction methods

| Var No | Method | Computing time (sec) | Acc | Optimal placement (Point numbers, $\theta_{min}$, $\theta_{max}$) | Coverage points |
|---|---|---|---|---|---|
| 1 | BFA | 190.48 | - | 17, 0° to 30°<br>9, 180° to 210°<br>1, 0° to 30° | 1-14<br>17-36 |
| 2 | Only C*RM*$_3$ | 152.45 | 1.25 | | |
| 3 | C*RM*$_2$ + C*RM*$_3$ | 28.15 | 6.77 | | |
| 4 | C*RM*$_1$ + C*RM*$_2$ + C*RM*$_3$ | 2.31 | 82.46 | | |
| 5 | C*RM*$_4$ | 49.61 | 3.84 | | |

When compared to the conventional brute force approach (variant 1), the acceleration (Acc) values for the other variants were as follows: method (2) – 1.25x faster, method (3) – 6.77x faster, method (4) – 82.46x faster and method (5) – 3.84x faster. These results demonstrate substantial improvement in computational efficiency achieved by complexity reduction methods while maintaining accuracy in calculating critical functions. We calculated the experimental task using multi-core parallel processing (OpenMP) with four physical cores, and hyperthreading wasn't used in this experiment. The computation time for method (5) depends on the number of processor cores utilized for parallel computing, with increasing cores leading to further reductions in computation time. However, parallel computing further accelerates computation but is limited by hardware capabilities.

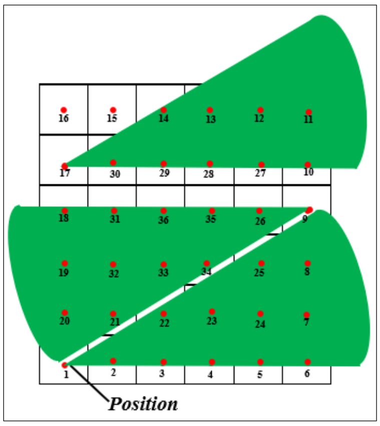


Fig. 2 Visual representation of the optimal placement of directional antennas

Figure 2 visually shows the optimal placement of directional antennas, achieving maximum coverage without overlap area. It illustrates the precise placement and beam orientations required to maximize coverage across the coverage area.

Combining complexity reduction methods significantly reduced execution times compared to the conventional brute force approach. Parallel computing provided additional speedup, demonstrating the benefits of leveraging hardware resources for improved performance.

## VII. CONCLUSION

The study highlights the effectiveness of complexity reduction methods in optimizing directional antenna placement, significantly improving the computational efficiency of the conventional brute force algorithm without maintaining accuracy. By combining reduction methods such as restricting rotation angles, limiting potential positions of directional antennas, and early termination of unsatisfactory calculations, execution times were significantly reduced, making these methods suitable for large-scale deployments. These approaches are applicable in various fields, including wireless sensor networks, telecommunications, and surveillance systems, where optimal antenna or sensor placement is critical for performance. Additionally, leveraging multi-core architectures and distributed computing can improve scalability for real-time and large-scale network optimization challenges.

## REFERENCES


[1] Hong-Ning Dai, Kam-Wing Ng, Minglu Li and Min-You Wu, “An overview of using directional antennas in wireless networks”, International journal of communication systems, 2013, Vol. 26, pp. 419-448, doi: 10.1002/dac.13481.

[2] Songwei Li, Chenyuan He, Mushuang Liu, Yan Wan, Yixin Gu, Junfei Xie, Shengli Fu, Kejie Lu, “The Design and Implementation of Aerial Communication Using Directional Antennas: Learning Control in Unknown Communication Environments”, IET Control Theory & Applications, 2019, Vol. 13 Iss. 17, pp. 2906-2916, doi: https://doi.org/10.1049/iet-cta.2018.6252.

[3] A. M. Thike, S. Lupin, H. Tun, O. Kostrova and R. Chakirov, “Topology optimisation of wireless internet infrastructure”, 2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW), St. Petersburg, Russia, 2016, pp. 362-364, doi: 10.1109/EIConRusNW.2016.7448194.

[4] Wang, Q., Dai, H.-N., Zheng, Z., Imran, M., Vasilakos, A.V., “On Connectivity of Wireless Sensor Networks with Directional Antennas”, Sensors 2017, 17, 134. doi: https://doi.org/10.3390/s17010134.

[5] M. Rizkinia, T. Baba, K. Shirai and M. Okuda, “Local Spectral Component Decomposition for Multi-Channel Image Denoising”, in IEEE Transactions on Image Processing, vol. 25, no. 7, pp. 3208-3218, July 2016, doi: 10.1109/TIP.2016.2561320.

[6] Xiaopeng Yang, Yuqing Li, Feifeng Liu, Tian Lan, Long Teng, Tapan K. Sarkar, “Antenna position optimization method based on adaptive genetic algorithm with self-supervised differential operator for distributed coherent aperture radar”, IET Radar, Sonar & Navigation, 2021, Vol. 15, doi: https://doi.org/10.1049/rsn2.12055.

[7] M. Bekar, C. J. Baker and M. Gashinova, “Sparse 2D MIMO Antenna Designs using Simulated Annealing”, 2023 20th European Radar Conference (EuRAD), Berlin, Germany, 2023, pp. 371-374, doi: 10.23919/EuRAD58043.2023.10289537.

[8] X. Zheng, “Application of Swarm Intelligence Optimization Algorithm in Antenna Design”, 2021 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Xi'an, China, 2021, pp. 773-779, doi: 10.1109/ICITBS53129.2021.00193.

[9] A. M. Thike, S. Lupin, B. H. Than and R. Chakirov, “The computational aspects of estimating the efficiency of wireless networks' topology”, 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg and Moscow, Russia, 2017, pp. 582-586, doi: 10.1109/EIConRus.2017.7910621.

[10] Aye Min Thike, Lupin, S.A., Fedyashin, D.A, “Using MPI library for parallel implementation of a brute-force algorithm”, Software & Systems, 36(4), 2023, pp. 607–614 (in Russ.). Doi: 10.15827/0236-235X.142.607-61.

[11] Aye Min Thike, Sergey Lupin, “Parallel Application for Wireless Network Topology Optimisation”, International Journal of Electrical, Electronics and Data Communication, vol.7, Iss.4, pp. 14-20, April 2019, ISSN(p): 2320-2084, ISSN(e): 2321-2950.

[12] A. M. Thike, S. Lupin, B. H. Than, H. H. Linn and K. N. Z. Lin, “Wireless network topology optimization — Decisions for directional antennas”, 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Moscow and St. Petersburg, Russia, 2018, pp. 1467-1471, doi: 10.1109/EIConRus.2018.8317374.

[13] Ай Мин Тайк, Лупин С.А., Телегин П.Н., Шабанов Б.М, “Ускорение расчета критериальной функции в задаче размещения всенаправленных антенн”, Программные продукты и системы. 2024. Т. 37. № 3. С. 197–206. doi: 10.15827/0236-235X.142.

[14] Hesham Adnan Alabbasi, Alaa Hussain Altimimy, “An Approach to Calculate Exact Coverage Area for Connected Wireless Sensor Network”, Journal of University of Babylon for Engineering Sciences, 2018, Vol. 26, No. 7, pp. 182-194.

[15] Feng Pan and S. Kasiviswanathan, “Efficient placement of directional antennas in infrastructure-based wireless networks”, 2011 - MILCOM 2011 Military Communications Conference, Baltimore, MD, 2011, pp. 1796-1801, doi: 10.1109/MILCOM.2011.6127572.

[16] A. M. Thike, S. Lupin and Y. Vagapov, “Implementation of brute force algorithm for topology optimisation of wireless networks”, 2016 International Conference for Students on Applied Engineering (ICSAE), Newcastle Upon Tyne, UK, 2016, pp. 264-268, doi: 10.1109/ICSAE.2016.7810200.

[17] G. Hadynski, S. B. Lee, G. Rajappan, R. Sundaram, X. Wang and F. Zhou, “Optimization of directional antenna network

topology in Airborne Networks", 2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE, San Jose, CA, USA, 2010, pp. 68-73, doi: 10.1109/MILCOM.2010.5680269.

[18] Ouamri Mohamed Amine, Abdelkrim Khireddine, "Base station placement optimisation using genetic algorithms approach", International Journal of Computer Aided Engineering and Technology, 2019, Vol. 11, Iss. 6, pp 635-652, doi: https://doi.org/10.1504/IJCAET.2019.102495.

[19] B. Joukar, M. Mohammadi and M. J. Ostovari, "Combination of Smart Antenna and Ant Colony Optimization in Multi-Purpose Wireless Sensor Networks", 2019 27th Iranian Conference on Electrical Engineering (ICEE), Yazd, Iran, 2019, pp. 1315-1320, doi: 10.1109/IranianCEE.2019.8786612.

[20] Aye Min Thike, Sergey Lupin, "Comparative Analysis of Calculation Methods for Over lapping Area in Antenna Placement Problem using Brute Force Algorithm", 2025 Confer ence of Young Researchers in Electrical and Electronic Engineering (EICon2025), January 28 - 30, 2025, St. Petersburg, Russia.

[21] Segura, C., González, Y., Miranda, G., León, C, "Parallel Hyperheuristics for the Antenna Positioning Problem", Distributed Computing and Artificial Intelligence - 7th International Symposium, DCAI 2010, Vol. 79, pp. 471-479, doi:https://doi.org/10.1007/978-3-642-14883-5_61.

Paper received 14 March 2025.

Sergey Lupin, Ph.D., Professor, National Research University of Electronic Technology «MIET», (e-mail: lupin@miee.ru);

Aye Min Thike, Ph.D., Doctorate Student, National Research University of Electronic Technology «MIET», (e-mail: ayeminthike52@gmail.com).