International Journal of Electrical, Electronics and Data Communication, ISSN(p): 2320-2084, ISSN(e): 2321-2950
Volume-7, Issue-4, Apr.-2019, http://iraj.in

PARALLEL APPLICATION FOR WIRELESS NETWORK TOPOLOGY OPTIMISATION

[1]AYE MIN THIKE, [2]SERGEY LUPIN

[1,2]High Performance Computing Department, National Research University Of Electronic Technology Zelenograd, Moscow, Russia
E-mail: [1]ayeminthike52@gmail.com, [2]lupin@miee.ru

Abstract - A parallel application realizing brute force method for solving the optimization task – distribution the towers with directional and omni-directional antennas in a wireless network is presented in this paper. The main purpose of optimization is maximize the network converge area and the level of populations having the access to network services. The example of wireless network topology uses two different types of antennas. Experiments aim on investigation the scalability of the algorithm's parallel implementation. In the computational experiments we use three different processors, including Intel Xeon Phi accelerator. For reducing the computational complexity of the algorithm the multi-thread application has been implemented. A parallel application demonstrates well level of scalability and efficiency.

Keywords - Wireless network topology, brute force algorithm, parallel implementation of algorithm, Open MP.

I. INTRODUCTION

Nowadays wireless networks have become an increasing popular choice because mobile computing devices such as laptops and Wi-Fi enabled phones are increasing in the workplace to access the various services. The engineering of wireless networks for enterprise environments remains a challenging task due to the problem of topological design. Every day the number of mobile clients is growing therefore we need to solve the optimization task - new towers distribution in a wireless network. Technologies of wireless networks are widely used in many applications: mobile communication, voice messages, navigation systems, health service, remote control, travel and entertainments, business, home networks, corporative management etc. Outdoor wireless networks use point to point technology with directional antennas for connecting two long distance nodes and omni-directional antennas with point to multi-point technology for distribution the signal to multiple clients. The engineering of network topology for directional antennas is more complexity due to additional argument – direction angle plus to coordinates [1].

In fact, the engineering of wireless network topology is an optimization task requiring introduction of various optimization algorithms. The cost of equipment, the level of security of access to network services, uniformity of a signal in the coverage area can be used as criteria of optimizations at this stage of the network design.

The topological optimization problems can be solved by enumerative methods. One of them is shrink and search algorithm (SSA) [2] for maximizing the network reliability of a distributed access network with constraint total cost. SSA jointly considers the upper bound of the network reliability for each branch in the combinatorial tree, and the cost constraint on the possible solutions. But this algorithm is not as such applicable for large size networks, such as networks having more than 20 nodes, because the size of the generated combinatorial tree is too large, which prohibits the exact calculation of network reliability. The enumerative methods discussed so far can be applied only to small networks.

Network coverage, connectivity, and lifetime are important issues in wireless sensors network (WSN). Open geographic density control (OGDC) is one of the popular coverage algorithms [3]. In this approach, the energy consumption can be reduced by controlling the density of the active sensors. This algorithm suitable for deterministic placement of the nodes, but this kind of placement is only possible for the small-scale networks.

The authors in work [4] proposed to solve the coverage problem as two sub-problems: floorplan and placement. The floorplan problem is to partition the service area into well-defined geometric cells and the placement problem is to assign the sensor devices into a set of cells. The two sub-problems are combining to an optimization problem of maximum coverage.

Article [5] presents the problem of area coverage as optimization tasks - minimum number of sensor nodes for maximum time duration in wide areas. Authors compared two types of deployment - random and deterministic. The disadvantages of deterministic deployment associate with the Area of Interest (AoI) limitation, and for random deployment with the non-equitable distribution of sensor nodes on AoI.

Article [6] describes the topological design as an optimization problem: finding a smallest set of nodes with maximum coverage. The authors present three optimization models. The first model is minimizing the sensor nodes number deployed into the area of interest to attain the high reliability level detection. The second model based on the determination of available nodes positions with nodes number constrain for attain perfect coverage, and the third - is to minimize the nodes number deployed in some locations that require low coverage and adjust the nodes number deployed in others locations that require higher coverage.

Some authors discussed the implementation of a genetic algorithm. The algorithm [7] specifies the positions and the number of sensor nodes to be deployed in the AoI in order to achieve optimal coverage. The application of this algorithm overloads the nodes by calculations that always leave the nodes in active state, the thing that quickly exhausts the network. The remedy proposed for this is the repetitive deployment of sensor nodes in locations where coverage has reached a minimal threshold, which is not always possible especially in danger zones.

The genetic algorithm is one of the strongest heuristics methods, which based on the principle of natural selection. It can be applied for the solution of optimizations tasks which can't be resolved under standard algorithms, particularly when criterion function is not continuous, not differentiable, stochastic or significantly nonlinear However, a major disadvantage of genetic algorithms is related to difficulties of their adaptation to task formulation. Paper [8] presents using genetic algorithm for finding the optimal location of the nodes in the network providing minimization of the energy consumed by sensors for wireless transmission.

Paper [9] describes the different strategies, which are used to maximize coverage in WSN. Algorithms are classified into three categories: force based, grid based and geometry computational based. Force based deployment depends on the mobility of sensor nodes, where sensor nodes are forced to move away or approach each other until they reach the perfect coverage. The force based strategy uses a virtual force as a repugnant and attractive force. This strategy provides coverage and connectivity on AoI, but it is extremely dependent on mobility and complex computations, which quickly depletes the network. The grid deployment based on the determining the precise positions of the sensors. In the strategy the coverage is evaluated as the ratio between the points of the covered grid and the total number of points of the grid in the AoI. The accuracy of the assessment is determined by the size of each grid, the smallest in size is the most accurate in the assessment. Authors discuss three types of grids which using in the deployment to provide the best coverage: Triangular, Square and Hexagonal grids. The triangular model is the best type due to the smallest area of overlap and the hexagonal grid is the worst of all, since it has the largest area overlap. Computational geometry is also frequently used to optimize coverage in WSN. Paper [10] presents two geometric computational approaches to provide the best coverage: Voronoi diagram and Delaunay triangulation. The Voronoi diagram bases on the division of the AoI on locations (sites) so that the points in a polygon are closer to the site in the polygon than the other sites. The Voronoi diagram can be used as a sampling method to determine coverage, and measure the holes in WSN. The Delaunay triangulation is used to evaluate the worst and best case of coverage [11]. The complexity of these approaches lays in the use of computational complexity geometry methods, which quickly depletes the network.

Paper [12] presents a Flower Pollination Algorithm (FPA) was introduced as a solution for deployment and coverage problem. A deployment approach based on FPA was proposed for the sensor placement problem for WSN. This approach can find the optimal placement topology in terms of Quality of service (QoS) metric. But FPA uses too many parameters to calculate for specify the positions of active nodes, which quickly depletes the network.

II. BRUTE FORCE ALGORITHM

As mentioned in publications, many methods of networks topology design are quite difficult to automate [13]. Our investigation aimed to approve that the problem of topology synthesis can be solved using parallel application of brute force algorithm (BFA). The quality of topology can be evaluated by various criteria. In this paper we present results for one of them.

BFA require significant computational resources and so was not been widely used in the past. Newest computer technologies such as multi-core processors and accelerators allow expanding the area of implementation of BFA and made it applicable for optimization tasks.

Brute force algorithm allows comparing all decision options and choosing from them the solution most corresponding to criterion of an optimization. The analyzing territory is present in the form of matrix structure where elements are settlements with known population.

Notations

Δx, Δy	Axial steps
N_x, N_y	Number of axial steps
N_{step}	Number of possible antenna positions
$\Delta\alpha$	Angle step
α_{step}	Number of angle directions
X_i, Y_i, α_i	Antenna's placement parameters
F_1	Criterion function
X_{ant_i}, Y_{ant_i}	Coordinates of *i*-antenna placement

Using BFA for wireless network topology optimization can be formulated as following:

1. Let $\{A_N\}$ be a finite set of network elements which can be allocated in possible positions $\{P_M\}$. Usually M >> N.
2. Find the distribution of elements $\{A_N\}$ at the positions $\{P_M\}$ corresponding to an extrema of some functional F. The solution of the problem is

the N-dimensional vector $\{D_N\}$ which elements belong to $\{P_M\}$ [14]:

$$\{D_N\} = (P_1, \dots, P_N), P_i \in \{P_M\}, \forall\, i = 1, N$$

$$F(D_N) \to \max\,(\min)$$

Thus, we will find the set of antennas positions which provide maximum value of selected criterion.

Let's define criteria of an optimality take into consideration the following parameters:

1. Power (P) of received signal is inversely proportional to the square of distance (R) between receive point and the antenna (P^0 - output power of transmitter):
$$P_i(R) = P_i^0/R^2 \quad (1)$$
2. The transfer of information between network nodes is possible if:
$$R \le R_{max} \quad (2)$$
3. There is the set of cities $\{T_L\}$, which are characterized by the location coordinates (X,Y) and the amount of residents (C):
$$T_i = (X_i, Y_i, C_i) \quad (3)$$

The criteria of wireless network topology quality are maximizing:

- the level of populations having the access to network services (F_1);
- the level of signal which accepted by the city's residents (F_2);
- the level of the distributed signal (F_3);
- the level of populations having the access to network services with the consideration of the space distribution of cities' residents (F_4).

In our experiments we use criterion F_1, which formulated as following:

$$F_1 = \sum_{i=1}^{L} D(T_i) \cdot C_i \to \max \quad (4)$$

where $D(T_i) = 1$ if the city T_i is in coverage zone of any antenna from $\{D_N\}$, C_i - the amount of inhabitants in i-th city.

2.1. Computational complexity of tasks

The computational complexity of the brute force algorithm will be determined by the number of antennas N and number of their possible positions M.

During the brute force algorithm application execution all possible placements for N antennas have been synthesized. Calculation of functional value F is performed at each point of (M^N)-dimensional space of decisions and computational complexity of the algorithm is defined as O(M^N) [15].

The complexity of calculation of all criteria is approximately the same, so our work focuses on computational aspects of the task those relate only with the forming of vector $\{D_N\}$, rather than criteria.

This formulation of the problem allows implementing a brute force algorithm as a multi-threaded application.

2.2. Experimental task

In our experiment the computational task is determined by the following parameters:

- maximum number of antennas in topology – 3;
- number of antennas range types – 2;
- number of steps across axis (N_x,N_y) – 10÷30.

Using these parameters, the number of options for equipment allocation can be calculated using the grid dimension: $M = 30^2 = 900$. If we assume that angle step for directional antennas rotation equal 10°, then $K = 360/10 = 36$. The maximum number of analyzed variants of decisions is: $(M * K)^N = (900 * 36)^3 \approx 34 * 10^{12}$. In this case the computational time of topology design is very longer therefore to reduce it we need to use parallel application.

Example comprises two types of antennas employed to transmit the signal (Table 1).

Table1. Parameters of Antennas

Antenna type	Range	R_{max} (km)	α_{bw} (degree)
ODA	A_1	10	-
	A_2	20	-
DA	A_1	10	20
	A_2	20	10

The antenna (A_1) can transmit signal to the maximum range equals R_{max}=10 km whereas the second type of antenna (A_2) provides reliable communication at R_{max}=20 km.

III. PROGRAM REALIZATION

The serial application has been implemented using programming language C++. Fig.1 and 2 show flowchart and code fragment of BFA serial application.

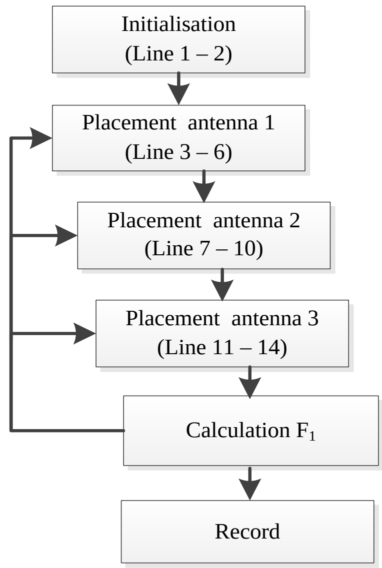

Fig.1. Flowchart of serial application

```
1:   N_step = N_x*N_y
2:   α_step = 360/Δα
3:   For I_1= 0 to N_step do
4:       X_1= [(I_1 / N_x)]
5:       Y_1= I_1 - (X_1*N_y)
6:       For α_1 = 0 to α_step do
7:           For I_2= 0 to N_step do
8:               X_2= [(I_2 / N_x)]
9:               Y_2= I_2 - (X_2*N_y)
10:              For α_2 = 0 to α_step do
11:                  For I_3= 0 to N_step do
12:                      X_3= [(I_3 / N_x)]
13:                      Y_3= I_3 - (X_3*N_y)
14:                      For α_3 = 0 to α_step do
15:                          Calculation F_1
16:                      end
17:                  end
18:              end
19:          end
20:      end
21:  end
```

Fig.2. Serial realization of BFA

3.1. Parallel application

As marked above, brute force algorithm has high computational complexity. So for the large number of allocations the parallel programming was employed. Parallel programming is being employed in the environment of Intel Parallel Studio 2017 and OpenMP Library [16]. In order to provide the parallelization in the software execution, OpenMP technology ensures operation of the sequential programs using compiler directives, environment variables, and library routines. This allows the multithreaded applications to be designed for multi-core or multiprocessor systems having shared memory. Generally, at the first stage of parallelization the translator selects the loop operator and provides performance the next iteration of this cycle in a new thread. Therefore, the function “omp parallel for” from OpenMP library can be used for distribution of the iterations of cycles.

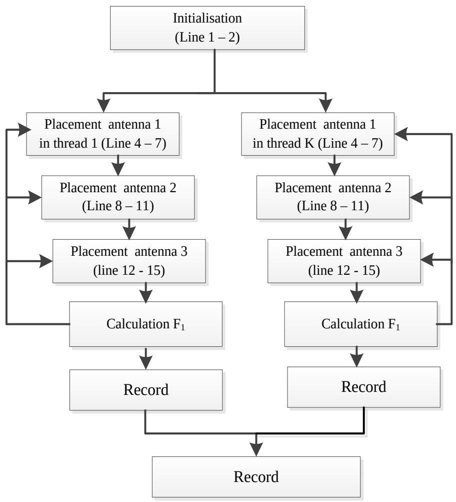

Fig.3. Flowchart of parallel application

Fig.3 and 4 show flowchart and code fragment of BFA parallel application. In BFA all calculations in threads are independently by data, because each thread estimates its own variants of topology. We define the value of global record after finishing all threads and comparing their local records.

```
1:   N_step = N_x*N_y
2:   α_step = 360/Δα
3: #program omp parallel for num_threads (core_number)
 private (I_1, I_2, I_3, α_1, α_2, α_3, X_1, Y_1, X_2, Y_2, X_3, Y_3, F_1)
4:   For I_1= 0 to N_step do
5:       X_1= [(I_1 / N_x)]
6:       Y_1= I_1 - (X_1*N_y)
7:       For α_1 = 0 to α_step do
8:           For I_2= 0 to N_step do
9:               X_2= [(I_2 / N_x)]
10:              Y_2= I_2 - (X_2*N_y)
11:              For α_2 = 0 to α_step do
12:                  For I_3= 0 to N_step do
13:                      X_3= [(I_3 / N_x)]
14:                      Y_3= I_3 - (X_3*N_y)
15:                      For α_3 = 0 to α_step do
16:                          Calculation F_1
17:                      end
18:                  end
19:              end
20:          end
21:      end
22:  end
```

Fig.4. Parallel realization of BFA

The accelerator Intel Xeon Phi has been used as a hardware platform for implementation of the multi-threaded parallel application. Intel Xeon Phi is the multi-core coprocessor with shared memory. Model 7120P, used in experiments, contains 61 cores [13]. It is possible to use two models of interaction with the coprocessor:

- offload: the application is run on the host and parts of code are unloaded to the coprocessor;
- native: the code is carried out initially on the coprocessor, all code and dependences have to be loaded on the device. This method was used in the experiments.

The calculation of criterion F1 (maximizing the level of populations having the access to network services) is more complexity procedure for directional antennas due to additional argument – direction angle plus to coordinates. Fig.5 and 6 show flowchart and code fragment of this procedure.

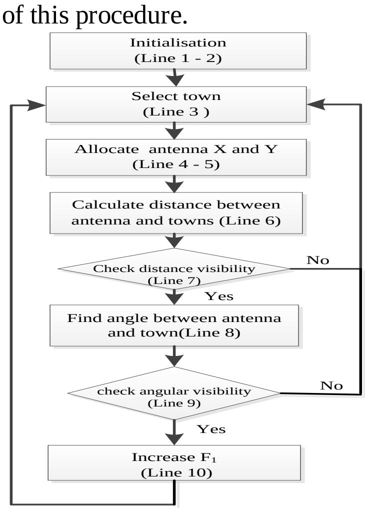

Fig.5. Flowchart of criterion calculation

$\Delta x = (x_{max} - x_{min}) / N_x$
$\Delta y = (y_{max} - y_{min}) / N_y$
For T=0 to N_t do
  $X_{ant\_i} = (\Delta x * Y_i) + \Delta x / 2$
  $Y_{ant\_i} = (\Delta y * X_i) + \Delta y / 2$
  $R = \sqrt{(X_{ant\_i} - X_T)^2 + (Y_{ant\_i} - Y_T)^2}$
  if ($R \le R_{max}$)
    $\alpha_{bat} = \mathrm{atan}((X_{ant\_i} - X_T) / (Y_{ant\_i} - Y_T))$
  if$(\alpha_{ant} - \alpha_{bw}/2) \le \alpha_{bat} \le (\alpha_{ant} + \alpha_{bw}/2)$
    $F_1 = F_1 + P_T$
end

Fig.6. Criterion calculation

At first we calculate the distance between antenna and town and check visibility by distance. After this for visible towns we calculate an angle between antenna and town and then check angular visibility. For visible town we increase criterion value by adding its population.

IV. EXPERIMENTS' RESULTS

For estimation the efficiency of BFA parallel implementation we used three computing platforms with parameters presented in the Table 2.

Table2. Computing platforms

Platform	CPU Type	Installed memory	Physical CPU
1	Intel®Core(TM) i7-3770K CPU 3.5GHz	8 Gb	4
2	2* Intel®Xeon® CPU E5-2630 v2 2.6GHz	32 Gb	12
3	Intel®Xeon Phi 7120p	16 Gb	61

The chosen platforms allow evaluating parallel applications from the point of view of loading existing compute cores. To reduce notations, we will use only the platforms' enter numbering in the tables with the experimental results. Intel Parallel Studio uses some of the system resources when the parallel applications running, so the results presented below contain these overhead expenses. Next figures and tables bellow are shown the results of the multi-threaded application's running on platforms 1 and 2.

Table 3. Results for platform 1

Number of threads	Execution time (Sec)	Acceleration
1	2081.36	-
2	1064.31	1.96
3	752.648	2.76
4	563.837	3.69

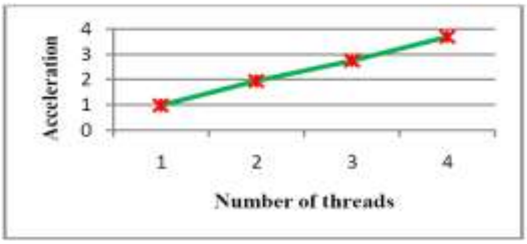

Fig.7. Acceleration of application on platform 1

Table 4. Results for platform 2

Number of threads	Execution time (Sec)	Acceleration
1	2479.41	-
2	1244.81	1.99
3	858.74	2.89
4	645.658	3.85
5	531.973	4.67
6	444.016	5.58
7	379.873	6.53
8	332.937	7.45
9	294.712	8.41
10	263.678	9.4
11	238.865	10.38
12	219.065	11.32

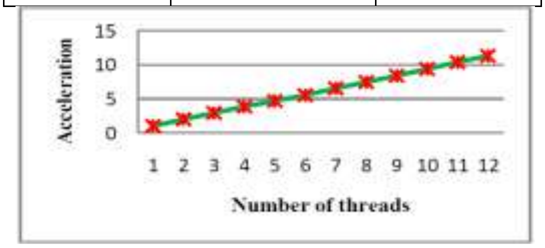

Fig.8. Acceleration of application on platform 2

The Intel Xeon Phi, which is part of platform 3, allows the use of 60 physical cores, so testing the scalability of the application on this platform is most informative. Results of the execution multi-threaded application on Intel Xeon Phi present in Table 5, Fig.9.

Table 5. Results for platform 3

Number of threads	Execution time (Sec)	Acceleration
1	5076.32	-
5	1077.65	4.71
10	551.091	9.21
15	361.059	14.06
20	284.327	17.85
25	221.206	22.95
30	187.965	27.3
35	158.102	32.11
40	137.321	36.97
45	126.93	39.99
50	117.043	43.37
55	103.032	49.27
60	95.9942	52.88

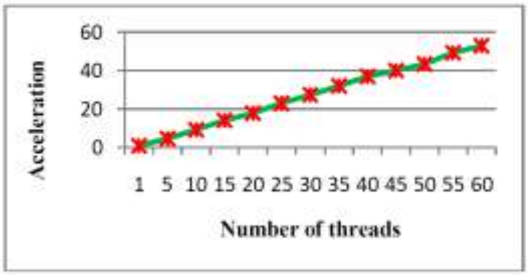

Fig.9. Acceleration of application on platform 3

The results of the analysis of efficiency of parallel implementation of the algorithm in the environment of Intel Parallel Studio are presented on Fig.10 (Locks and Waits Analyze). The application efficiently uses all computing resources of the platform 2 (12 physical cores) and provides 11.32 times faster computing.

All examples confirm that the BFA can be effectively realized as multi-threaded application. Algorithm is invariant in relation to different criteria and this allows changing the optimization purposes without essential rebuilding the application. Moreover, with a little changes BFA allows finding not one but all existing solutions of the optimization task.

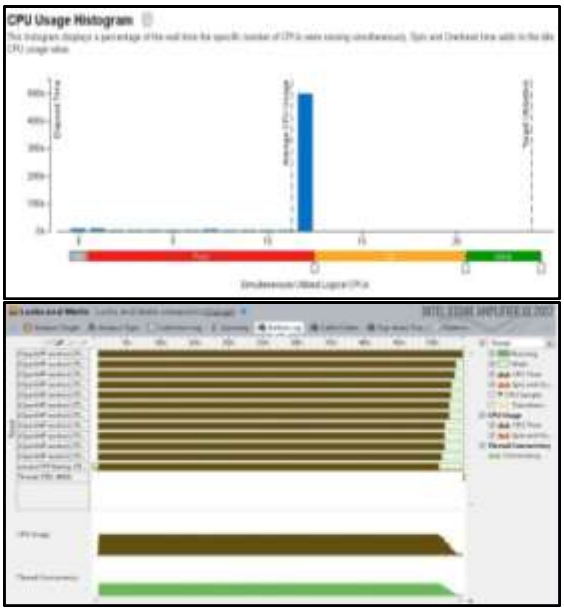

Fig.10. Results of Locks and Waits Analyze

The presented results show that the application has good scalability across the entire range of studies and multi-core accelerators can be considered as the most appropriate platform for its implementation.

CONCLUSIONS

Presented results confirmed that brute force algorithm can be used for distribution the directional antennas in wireless network topology optimization process. Of course, this task has high level of complexity, but parallel computations allow applying algorithm also for solving the practical tasks. Future research will be aimed to decrement K value by means of the limitation the antennas angle rotation in near border positions.

ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Basic Research (RFBR) as a part of the research project No 16-07-01055\165.

REFERENCES

[1] Aye Min Thike, Sergey Lupin, Ba Hla Than, Htun Htun Linn, Kyaw Nay Zaw Lin, “Wireless network topology optimization — Decisions for directional antennas”, 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 29 january- 1 february 2018, Moscow, Russia.
[2] Fang-Ming Shao, Xuemin(Sherman) Shen, and Pin-Han Ho, Reliability optimisation of Distributed Access Networks With
[3] Constrained Total Cost, IEEE Transactions on Reliability, 54(3), (2005), 421-430.
[4] C. Giolli, A. Scrivani, G. Rizzi, F. Borgioli, G. Bolelli, and L. Lusvarghi, “Failure mechanism for thermal fatigue of thermal barrier coating systems”, Journal of Thermal Spray Technology,vol.18,pp.223–230,2009.
[5] S.J. Habib, Modeling and simulating coverage in sensor networks Computer Communications, Computer Communications 30 (2007) 1029-1035.
[6] Adda Boualem, Youcef Dahmani, Abdelkader Maatoug ,Cyril De-runz, “Area Converage Optimization in Wireless Sensor Network by Semi-random Deployment”, 7th International Conference on Sensor Networks(SENSORNETS 2018), pp.85-90, 22-24 January 2018.
[7] Fahad Alduraibi, Noureddine Lasla, Mohamed Younis, “Coverage-based Node Placement Optimization in Wireless Sensor Network with Linear Topology”, IEEE ICC 2016 Ad-hoc and Sensor Networking Symposium, 22-27 May 2016, Kuala Lumput, Malaysia.
[8] Omar Banimelhem, Moad Mowafi, Walid Aljoby, “Genetic Algorithm Based Node Deployment in Hybrid Wireless Sensor Networks”, Communications and Network, no.5, pp.273-279, 8 August 2013.
[9] Amol P.Bhondekar, Renu Vig, Madan Lal Singla, C Ghanshyam, Pawan Kapur, “Genetic Algorithm Based Node Placement Methodology For Wireless Sensor Networks”, Proceedings of the International MultiConference of Engineers and Computer Scientists 2009, vol 1, pp.106-112, 18-20 March 2009, Hong Kong.
[10] Xingfa Shen, Jiming Chen, Youxian Sun, “Grid Scan: A Simple and Effective Approach for Coverage Issue in Wireless Sensor Networks”, 2006 IEEE Internaional Conference on Communications, 11-15 June 2006, Istanbul, Turkey.
[11] Franz Aurenhammer, “Computational Geometry - Some Easy Questions and their Recent Solutions”, Journal of Universal Computer Science, vol. 7, no.5, pp.338-354, 28 May 2001.
[12] Seapahn Megerian, Farinaz Koushanfar, Miodrag Potkonjak, Mani B.Srivastava, “Worst and Best-Case Coverage in Sensor Networks”, IEEE Transactions on Mobile Computing. vol.4, no.1,pp. 84-92, 17 January 2005.
[13] Faten Hajjej, Ridha Ejbali and Mourad Zaied, “An Efficient Deployment Approach for Improved Coverage in Wireless Sensor Networks based on Flower Pollination Algorithm”, Computer Science & Information Technology(CS & IT), pp. 117-129, 2016.
[14] Aye Min Thike, Sergey Lupin, Hein Tun, Olga Kostrova, “Topology Optimisation of Wireless Internet Infrastructure”, 2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW), pp. 362-364, 2-3 February, 2016, St. Peterburg, Russia.

[15] Aye Min Thike, Sergey Lupin, Yuriy Vagapov, "Implementation of Brute Force Algorithm for Topology Optimisation of Wireless Networks", 2016 International Conference for Students on Applied Engineering (ICSAE), 20-21 October, 2016. Newcastle upon Tyne, UK.
[16] Aye Min Thike, Sergey Lupin, Roustiam Chakirov, Yuriy Vagapov, "Topology Optimisation of Wireless Sensor Networks", 2016 International Conference on Design, Mechanical and Material Engineering (D2ME 2016), vol 82, 8 - 10 September, 2016, Auckland, New Zealand.
[17] OpenMP. [Online]. Available: https://en.wikipedia.org/wiki/OpenMP.

★★★