Łukasz Szklarski

Sensor Network Deployment Optimization for Improved Area Coverage Using a Genetic Algorithm

Security Dimensions. International & National Studies nr 3 (19), 150-181

2016

SECURITY DIMENSIONS
INTERNATIONAL & NATIONAL STUDIES
NO. 19; 2016 (150–181)

Sensor Network Deployment Optimization for Improved Area Coverage Using a Genetic Algorithm

Łukasz Szklarski, M.A.
ITTI Ltd, POLAND

ABSTRACT

Ensuring optimal coverage is a central objective of every sensor deployment plan. Effective monitoring of the environment helps to minimize manpower and time, while enhancing surveillance capability. In this paper, a solution for improved area coverage was presented. A lattice of a pre-defined parameter has been used as an input for the algorithm. For the purpose of the research the blanket deployment strategy has been adopted. Then, a genetic algorithm has been proposed and implemented to find an optimal solution. The proposed approach has been tested and the conclusions have been drawn. The results proved that the proposed genetic algorithm could already provide satisfactory results, usually finding only suboptimal solutions.

ARTICLE INFO

Article history
Received: 01.08.2016 Accepted 23.09.2016

Key words
network of sensors, genetic algorithm, sensors deployment, area coverage

1. Introduction

The paper is motivated by the new developments in the area of sensor networks and on-going research devoted to deployment plans, which are to guarantee the most efficient performance of the network. The aspect to maintain a superior quality of the network coverage seems to be of paramount importance for many researchers, especially due to the critical role of sensors, i.e. CBRN substance detection, threat detection, etc. Therefore, research often delves into the development of optimized deployment plans based on a range of mathematical algorithms (random search algorithm, exhaustive search algorithms, genetic algorithm, etc.). The key factor underpinning many efficient networks is high percentage of area coverage achieved via minimal number of sensors. As a consequence, such networks are easier to implement, especially if one has a limited budget at one's disposal. Additionally, proper data fusion algorithms that networks incorporate may allow for a range of desired effects, including reduction of false alarm, detection of faulty network components, or decrease redundant data. The data redundancy issue is taken into account by many in the process of designing a given sensor network, especially when one aims at the overall cost reduction. Avoiding redundant data implies such a sensor placement, which do not cover the same area as other sensors.

The hereby paper, therefore, approaches the subject of sensor network deployment in a comprehensive manner. First of all, the focus is put on the presentation of network superiority over standalone sensor placement with clear explanation of the features ensuring improved efficiency including data fusion, network management, connectivity and scalability. Furthermore, the focus is shifted towards the description of related works, which are connected with deployment issues, coverage problems, and genetic algorithm. To provide a reader with better understanding of sensor network placement, the paper features a characteristic of three basic deployment strategies including *blanket deployment*, *barrier deployment*, and *target-oriented deployment*. However, the central subject of the research paper is to fully delve into the blanket coverage strategy. Consequently, in the following sections of the paper, a step-by-step approach of genetic algorithm creation is depicted. This is followed by a section summarizing the overall results of the selected approach. The paper concludes with a clear-cut guidelines on the future research needed to improve the proposed sensor network deployment strategy.

2. Network advantages

While considering the implementation of sensor network, it seems essential to touch upon the advantages the network has over the independent sensors deployed in a given area. First of all, there is an issue of management and operating sensors. While sensors are not integrated within the network, it may be a daunting task to monitor the output coming from tens or hundreds of sensors deployed over a given area. The sensor network seems to have an edge over standalone sensors, in this respect, thanks to the proper network management software.

Furthermore, it may be indicated that the network's major advantage lies with regard to the possibility of data fusion[1]. Harnessing the sensor network enables, therefore, the reduction of false-positive and false-negative alarms through the application of various algorithms, e.g. fuzzy logic, neural network or Bayesian network algorithm. Data fusion may also provide other benefit such as automatic detection of faulty network components, which in certain types of standalone solutions is problematic to determine. Faulty components are not always tantamount to a complete breakdown of a given device. It is also possible that certain sensors provide incorrect readout of the environmental data, which might be difficult to establish if they are randomly and independently deployed.

What is more, incorporating sensors in a network paves the way for researchers to explore the potential of heterogeneous sensors deployment. It is assumed that thanks to such approach, sensor networks will be capable of detecting a wider range of threats (e.g. thanks to sensors based on different technologies), and the detection process itself will be facilitated[2].

The benefits that come with sensor network can be further attributed to either wired or wireless sensor networks. In case of the former, the data transmission is much faster and it is less susceptible to potential

[1] Y. A. Vershinin, *A Data Fusion Algorithm for Multisensor Systems*, [in:] *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, 2002, vol. 1, p. 341-345; J. S. S. Z. L. L. a. L. S. Y. Chen, *Data Fusion in Wireless Sensor Networks*, [in:] *Electronic Commerce and Security, 2009. ISECS '09. Second International Symposium on,* 2009, p. 504-509; D. Izadi, J. H. Abawajy, S. Ghanavati, T. Herawan, *A Data Fusion Method in Wireless Sensor Networks*, "Sensors" (Basel, Switzerland), 2015, vol. 15, no. Issue: 2, p. 2964–2979.

[2] L. Lazos, R. Poovendran, *Coverage in Heterogeneous Sensor Networks*, [in:] *Network Security Laboratory (NSL)*, Department of Electrical Engineering, 2006; B. K. a. C. C. J. K. Jae-Joon Lee, *Impact of Heterogeneous Deployment on Lifetime Sensing Coverage in Sensor Networks*, [in:] *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on,* 2004, p. 367-376.

jamming. In case of the latter, on the other hand, the implementation process of the network is not as time-consuming as wired network. Additionally, wireless sensor networks are easily scalable[3]. Therefore, the replacement of faulty components or extension of the network with additional components does not require a network configuration from the scratch.

Current applications for sensor networks can be found not only in military context, for example, in security and tactical surveillance over a wide area, but they also expand to social purposes, such as habitat monitoring or disaster intervention. Regardless the purpose, both economic and technical factors need to be taken into consideration. Although, along with evolution of sensor technology, their price decreases, it is still desired to cover the area of interest with the minimal number of sensors.

3. Related work

Efficient sensor deployment has been the subject of researchers' considerations for many years now[4]. One of the main problems that is often discussed is the problem with the coverage. Coverage is the most crucial metric measuring the performance of sensor network and it indicates if the sensor field is well monitored. Each sensor has its sensing capability and quality, which denote the ability of a single sensor to cover certain area. When there are multiple sensors located in a given area, the problem of network sensing coverage arises. In general, coverage problems may be classified into point (target) coverage, area (blanket) coverage and barrier coverage[5].

---

3 J. S. S. Z. L. L. a. L. S. Y. Chen, *op. cit.*, p. 504-509.

4 D. Izadi, et. al., *op. cit., p. 2964-2979;* B. K. a. C. C. J. K. Jae-Joon Lee, *op. cit.,* p. 367-376; J. Beutel, K. Römer, M. Ringwald and M. Woehrle, *Deployment Techniques for Sensor Networks*, [in:] *Signals and Communication Technology*, 2009, p. 219-248.

5 Y. Yoon, Y.-H. Kim, *An Efficient Genetic Algorithm for Maximum Coverage Deployment in Wireless Sensor Networks*, [in:] *IEEE Transactions on Cybernetics*, 2013, vol. 43, no. Issue: 5, p. 1473-1483; B. Wang, *Coverage Problems in Sensor Networks: A Survey*, [in:] *ACM Computing Surveys (CSUR)*, 2011, vol. 43, no. Issue 4, p. Article No. 32; H. Kim, S.-w. Han, *An Efficient Sensor Deployment Scheme for Large-Scale Wireless Sensor Networks*, [in:] *IEEE Communications Letters*, 2015, Vol. 19, Issue 1, p. 98-101; V. Sharmaa, R. Patelb, H. Bhadauriaa, D. Prasadc, *Deployment Schemes in Wireless Sensor Network to Achieve Blanket Coverage in Large-Scale Open Area: A Review*, "Egyptian Informatics Journal", 2016, vol. 17, no. Issue 1, p. 45–56.

Genetic algorithm, widely discussed in many works, is frequently used to solve complex problems[6]. To the population (set) of potential solutions, the principle of survival of the fittest is applied. In result, a solution best answering the problem is selected[7]. The algorithm finds its application in numerical and combinational optimizations, machine learning and engineering design. The paper presents the usage of genetic algorithm in finding the best solution to the optimal sensor placement problem.

In the following paper, due to the nature of the research, focus will be given to blanket coverage problem. The solution presented is based on the genetic algorithm and it ensures optimal sensor deployment for the area of interest with the use of minimal number of sensors required.

4. Deployment strategies (Coverage problems)

There is a number of deployment strategies, however, for the purpose of this paper, the following coverage types are shortlisted:

A. Blanket deployment

As proposed by Gage[8], the blanket coverage type refers to wide area monitoring using numerous sensor nodes. The sensors can be distributed with the application of random deployment algorithms[9]. It is also feasible to apply deterministic deployment algorithms, however the complexity of this solution is far larger than random algorithm.

[6] Y. Xu, X. Yao, *A GA Approach to the Optimal Placement of Sensors in Wireless Sensor Networks with Obstacles and Preferences*, [in:] *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference,* 2006, vol. 1, p. 127-131; D. W. Gage, *Command Control for Many-Robot Systems*, [in:] *Proceedings of AUVS-92,* 1992, vol. 10, no. Issue: 4, p. 28-34; S. S. Dhillon, K. Chakrabarty, *Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks*, [in:] *IEEE Wireless Communications and Networking Conference (WCNC)*, 2003, p. 1609-1614; B. Liu, D. Towsley, *A Study of the Coverage of Large-Scale Sensor Networks*, [in:] *IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS)*, 2004, p. 475-483.

[7] J. Beutel et. al., *op. cit.,* p. 219-248.

[8] Y. Yoon and Y.-H. Kim, *op. cit.,* p. 1473-1483.

[9] M. Cardei and J. Andwu, *Coverage in Wireless Sensor Networks,* [in:] *Handbook of Sensor Networks,* 2004, Chapter 19.

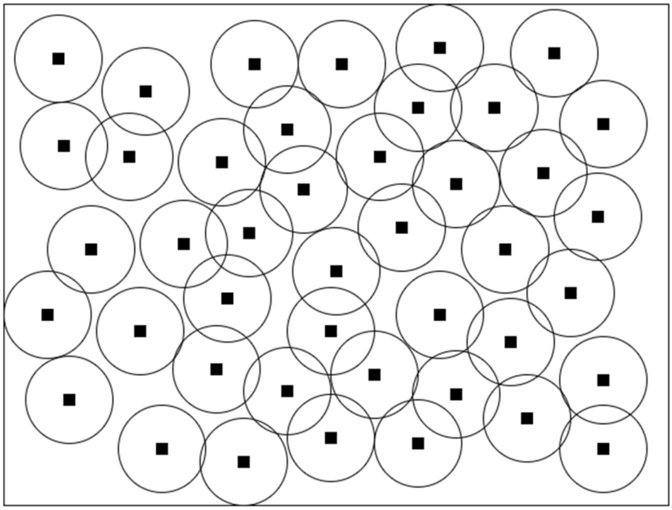

Figure 1. Blanket coverage type

As can be noticed in the Figure 1, the blanket deployment aims to cover the whole area of interest. One of the major problems, which can occur during this kind of deployment is high probability of generating blind spots in the network sensing area. Moreover, when inappropriate deployment algorithm is used, sensors deployment can be focused around several points in the area, providing smaller coverage and more redundant data. This problem can be solved by sensors relocation algorithms. However, in this case, mobility of the sensors is required.

B. Barrier deployment

While blanket deployment is focused on the coverage of the whole area, the barrier deployment strategy is concentrated on the area's contour. This contour can, for example, consist of straight lines laid along the border[10], as presented in the Figure 2.

[10] S. S. Dhillon, K. Chakrabarty, *op. cit.*, p. 1609-1614.

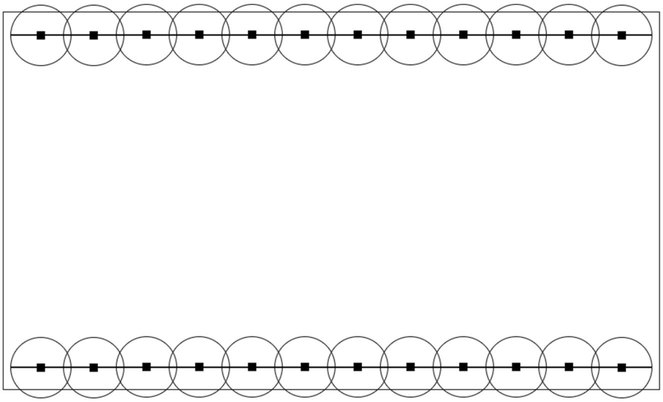

Figure 2. Barrier coverage type

In this case, sensors are deployed using deterministic algorithm (point to point), the same deployment strategy could be used in case the line is curve.

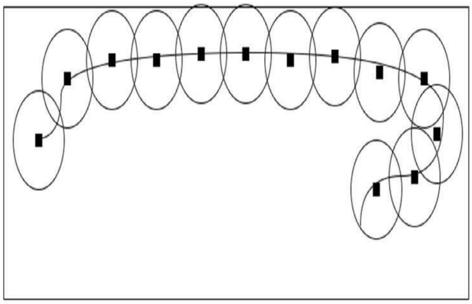

Figure 3. Barrier coverage type (curve)

This kind of deployment strategy is relevant, for example, for border surveillance, dangerous substance monitoring or critical infrastructures protection[11].

C. Target oriented deployment

Some scenarios require special attention to a specific points of interest (PoIs). This deployment plan focuses on deploying sensors mainly in the most important areas. It is mostly used when the area is known well and the sensors should be deployed precisely[12].

[11] B. Liu, D. Towsley, *op. cit.,* p. 475-483.

[12] C. Zhao, Z. Yu, P. Chen, *Optimal Deployment of Nodes Based on Genetic Algorithm in Heterogeneous Sensor Networks*, [in:] *IEEE International Conference on Wireless Communications*, 2007, p. 2743-2746.

The deployment strategies can be also further divided into two types considering the mobility of the sensors network[13]. Either *stationary* or *mobile* deployment can be identified. In the former, sensors are deployed in fixed positions in the area. This kind of deployment can realize any of the previously shortlisted plans. However, usually larger amount of sensors is required to fulfil the chosen plan than in case of mobile network. The latter type, on the other hand, affects scenarios where sensors are mounted on mobile platforms and can be moved automatically or manually. For example, unmanned aerial and ground vehicles can be used as such platforms. Mobility of sensors enables dynamic relocation of network nodes. The network can be adapted to frequently changing conditions such as weather. Moreover, deployment plan can be also changed dynamically, if necessary. For instance, a mobile group of sensors can be send to relocate and change required coverage from blanket to barrier. However, this usually requires more complex algorithms for data fusion, automatic positioning and dynamic relocation. There was also research done about self-positioning of mobile sensors network[14].

5. Proposed method

In this chapter, the development of the algorithm for network sensors deployment is described in a step-by-step manner. First, preliminary assumptions and basic algorithms are discussed. Then, objective function of the optimization task is described. Finally, the development process of genetic algorithm (GA) and sample deployments are presented.

A. Assumptions

The main objective of the blanket deployment strategy is to cover as high percentage of the area of interest as possible while applying predetermined number of sensors. The preliminary assumption for the proposed deployment is to harness homogenous point detection sensors. Moreover, as noted above, the set of predefined potential deployment points (e.g. lattice of a given parameter) is used as the algorithm's input. The sample set of input points in 2D coordinate system is presented in the Figure 4. The distance unit is arbitrary, however it is assumed to be the same in all calculations. Such a definition of the problem creates an immense search space. The

[13] B. K. a. C. C. J. K. Jae-Joon Lee, *op. cit.*, p. 367-376.

[14] D. W. Gage, *op. cit.*, p. 28-34.

number of possible deployment options can be calculated using the following formula:

$$N = \binom{l^2}{n} \quad (1),$$

Considering the equation above, *l* is a lattice size, *n* is a number of deployed sensors and *N* is a number of possible solutions to the problem. For instance, in case of a sample 10x10 lattice incorporating 7 sensors, there is *N* = *16007560800* possible deployments. This factor, connected with high objective function complexity, makes exhaustive search impossible to perform in short period of time. Fast and efficient algorithm has to be used in order to quickly find an optimal solution. The development process and algorithm's design will be depicted in the following sections.

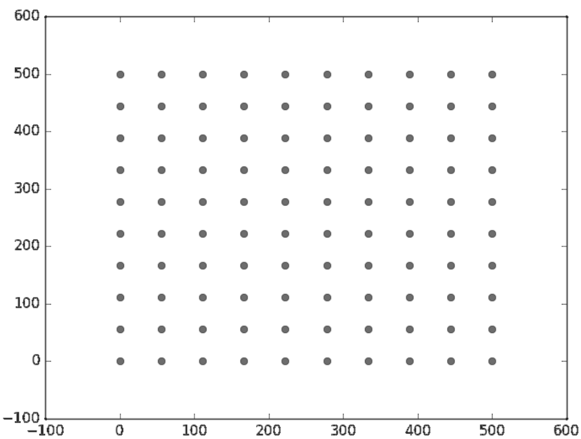

Figure 4. Sample set of possible sensor deployment points

B. Preliminary approach

The preliminary approach assumed the implementation of the exhaustive search and random (based on Monte-Carlo method) algorithms in order to design an objective function for the optimization task. These simple solutions were, then, verified against a genetic algorithm by taking execution time and overall quality into consideration.

The exhaustive search algorithm (ESA) tests every possible solution. The main asset of this solution is that it always finds the optimal result from the search space. The crucial drawback, however, is its lack of time-efficient approach (e.g. evaluation of 8x8 lattice and 7 sensors took approximately 8 hours to compute[15]). ESA will be used in validation and verification processes in order to compute simple problem solution and compare it with genetic algorithm output.

The random search algorithm (RSA) was also designed and implemented. RSA tests random solutions from the search space and computes them until a sufficient quality or maximum iterations is reached. Due to the uncomplicated implementation, this algorithm can be used in order to find a good starting point for different algorithms (e.g. GA). There are several approaches for RSA implementation. The most common one implies the selection of a random element from the full set of possible solutions and computing its quality measure. The other assumes making random changes in actual solution, memorizing previous solutions in order to reduce redundant computations. The main advantage of RSA is that it can quickly find a suboptimal solution. However, in its basic form, there is no control of the optimization process and algorithm can fall in the loop, testing the same solution several times.

These two algorithms were defined and applied for objective function development and validation of the genetic algorithm approach.

C. Objective function definition

The objective function constraints and solution search method are the main aspects of the optimization tasks. The constraints in the analyzed task are defined with the use of the lattice that identifies all possible sensors' positions. The most challenging and critical aspect is objective function definition, via which a given solution quality is calculated. In the genetic algorithm approach, objective function is commonly referred as a fitness function[16].

---

15 All calculations performed within the hereby described research were done on a computer with Intel i5-2410M processor.

16 J.-H. Seo, Y.-H. Kim, H.-B. Ryou, M. Jo, *Optimal Sensor Deployment for Wireless Surveillance Sensor Networks by a Hybrid Steady-State Genetic Algorithm*, [in:] *IEICE Transactions on Communications E91-B(11)*, 2008, p. 3534–3543.

Objective function used in the deployment optimization task has to ensure that optimal solution provides the largest area coverage. This can be achieved through maximizing the distances between individual sensors. For example, the following objective function can be utilized:

In the equation above *Q* stands for the quality, *si* is an individual sensor, *n* is the number of sensors in the current deployment and *distances* is a function returning vector of Euclidean distances between a given *si* and every other sensor. This objective function was tested by using an exhaustive search algorithm (7x7 lattice, 5 sensors). The output deployment is presented in the Figure 5.

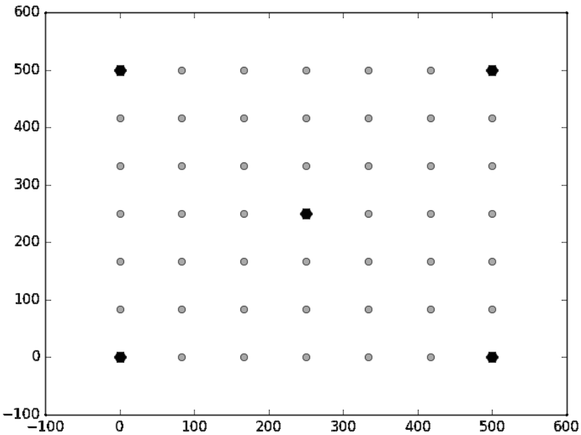

Figure 5. Deployment of five sensors using unmodified objective function

The above mentioned objective function definition has one major drawback. Namely, most sensors are deployed on the edges of the area. However, in some cases, there is a need to focus deployment in the center of the area. In order to overcome this issue, an additional factor should be added to the objective function. This factor is responsible for keeping the distance between sensors and area's edges and it is modelled as an additional multiplier of the equation (2). Figure 6, presents an improved deployment concept. It must be added that the algorithm can be further manipulated by changing the added weight factor. The higher it is, the more centralized deployment output will be produced as a result.

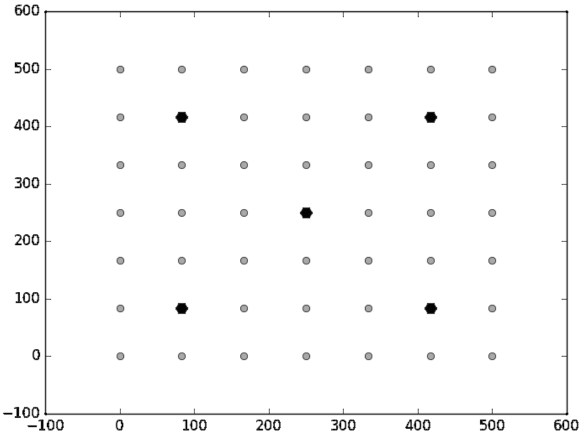

Figure 6. Deployment of five sensors with modified objective function

D. Genetic Algorithm

Genetic Algorithms (GA) are heuristic methods, which among plurality of their applications, can be used for solving optimization tasks. They are modelled on processes of natural selection, reproduction and mutation. Therefore, definitions such as gene, chromosome, population, individual are part and parcel of GAs. In case of the sensor network deployment optimization task, a single sensor is defined as a chromosome and the entire network deployment as an individual. Each chromosome consists of two genes: x-axis and y-axis coordinates. In the future work, chromosome can be extended by including genes that store information about: sensor's model, sensing range or sensor's type. However, the current definition is sufficient enough for preliminary genetic algorithm design.

The developed genetic algorithm is specified in the following steps:

1. Input data are provided – lattice, the number of epochs and sensors.
2. The initial population is created.
3. Algorithm is executed:
   a. Fitness value is calculated for entire population;
   b. Selection operator – best individuals are selected as parents;
   c. Crossover operator – new individuals are created from parents;
   d. Mutation operator – individuals not selected for crossover are mutated;

e. Population guard – incorrect individuals are mutated;
f. If reached given number of epochs, go to 4, else go to 3.
4. Return individual with the highest fitness value.

As a first step, an input is provided for the algorithm. The lattice of a given parameter has to be created first and stored in the program's memory (dynamic list is used for this). Moreover, at this step, a number of sensors and maximal number of GA epochs is defined. The GA epoch stands for one iteration of the algorithm, which includes one pass through all basic operators and fitness calculation described later. After this, initial population is created. The initial population consists of X random individuals (random individual generator function for RSA is used). When these steps are carried out, algorithm proceeds to its main execution part. In every epoch, consecutively, fitness calculation, selection, crossover and mutation operators are applied. Fitness calculation is performed by determining the quality measure and assigning it to each individual. It is done by using previously defined objective function. Fitness rating is used in the next step of the GA – Selection. There are many alternatives as far as selection methods are concerned including[17] roulette wheel or tournament selection algorithm.

For the sensor deployment optimization task, a tournament selection has been chosen due to its efficiency and simplicity. In this method, the predefined number of individuals is selected from the population. Subsequently, the one with the highest fitness value is chosen as a future parent. This is the repeated for Y times, with Y equal to the number of parents selected for reproduction. The tournament selection is easily scalable and can be manipulated by changing tournament size (the number of individuals chosen from the population). Parents selected during tournament are directly assigned to the new population. After this step, a crossover operator is applied. Parents are paired first and, then, a new individual is created from every pair. The crossover operator, used in the described algorithm, relies on selecting random crossover point in chromosomes vector. The first chromosomes are inherited from the first parent, and the rest from the second one. Parents' children are, subsequently, added to the new population. As a consequence, the new population is complemented to its

[17] S. K. Guptaa, P. Kuila, P. K. Jana, *Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks*, "Computers & Electrical Engineering", 2015, p. 1–13.

starting size by using mutation operator. Individuals that were not selected for reproduction mutate. In this GA, a simple mutation operator is used. It is performed by creating a new random individual in the place of the old one. This is achieved through the application of the same function that was used during the creation of the initial population. Due to the random nature of the mutation and crossover operators, flawed individuals might be occasionally generated. For example, as a result of reproduction, an individual solution with several sensors deployed in the same place can be produced. Such a solution has its quality equal to zero (with the usage of objective function equation (2)). What is more, if it is assigned to the new population, there is a small probability of "infecting" other individuals. In order to deal with this problem, the population guard functionality has been implemented. Every individual is checked by a special function, and in case of finding an invalid one, mutation operator is applied. As the last step of the loop, the algorithm checks if it reached a given epoch. If not, then another iteration is performed. In other case, the individual with the highest fitness value from the current population is returned as an output. The sample outputs of the algorithm (initial population size – 200, epochs – 400) are presented in the Figure 7 (27x27 lattice, 11 sensors) and 8 (27x27 lattice, 17 sensors).

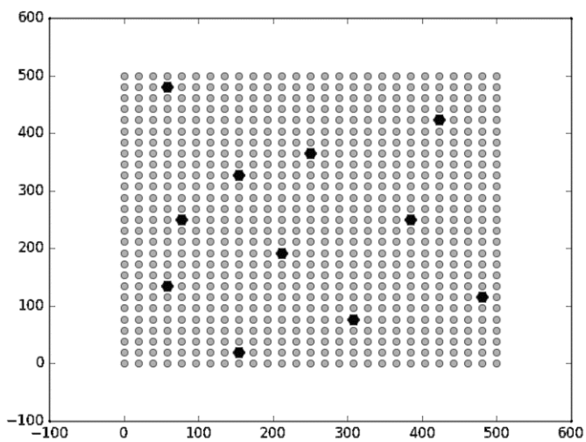

Figure 7. Sample deployment of 11 sensors, using genetic algorithm

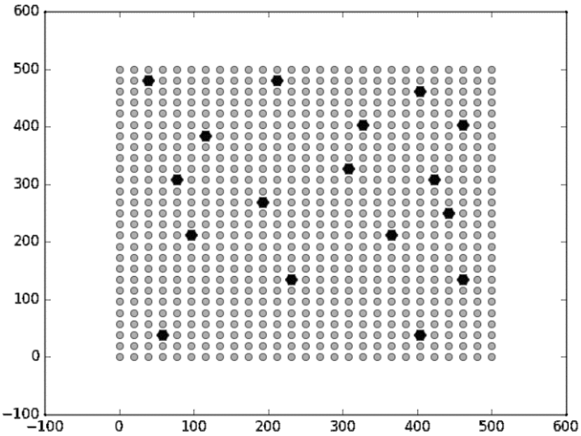

Figure 8. Sample deployment of 17 sensors, using genetic algorithm

6. Results

First of all, this chapter discusses the method for computing overall quality of the deployment. Afterwards, the results from the different simulations are presented. Following factors were taken into consideration during the research: initial population size, the number of sensors and epochs. What is more, the random and exhaustive search algorithms results are shown for sample deployment scenarios.

A. Validation method

After an optimization solution has been found, its absolute quality function has to be specified in order to compare it with different algorithm solutions. The following test algorithm was created:

1. Set of test points is defined.
2. Effective detection range for each sensor is determined specified.
3. For each sensor, it is verified whether there is any test point within range. If yes, the point is removed from the list.
4. Detection rate is calculated as percentage of detected points.

First of all, a set of test points needs to be established. For the current research, the lattice of equally distributed points will be used. The lattice has different constant than the one used for deployment. The set of test points is stored in the dynamic list in order to efficiently manipulate it.

In the next step, detection range should be defined for each sensor. At the current research, the usage of homogenous sensors with the common sensing range is assumed. After this preliminary work, each sensor from the tested deployment is checked if it can detect any point from the defined set. Detected points are removed from the list. Subsequently, the detection rate is calculated. It is defined as follows:

Sc is the size of test points set after detection check and Sp is the preliminary size of test points set. The sample validation of test result is presented in the Figure 9. Black lines indicate which sensor detected a given point and grey cross markers show which points were not detected by the network. Detection range of 150 (arbitrary unit) was assumed in this test.

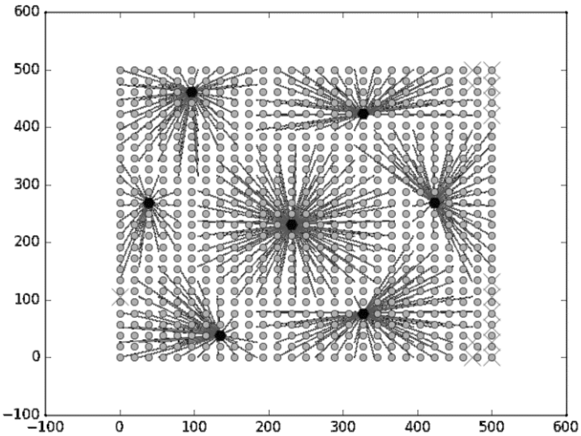

Figure 9. Sample of deployment and validation test visualization

This test algorithm can be extended by providing additional calculations. For example, additional statistics for test points can be evaluated in order to check how many sensors detected a given point.

B. Genetic algorithm – number of epochs impact on output deployment

In this research, focus will be put on determining the impact of number of epochs on the output deployment. Simulation tests for 5, 10, 20, 1000 epochs were performed. Initial population consists of 50 individuals (13 chromosomes each), effective sensing range is set to 100 and validation test points set is based on 27x27 lattice. The results are shown in Figure 10, Figure 11, Figure 12, and Figure 13.

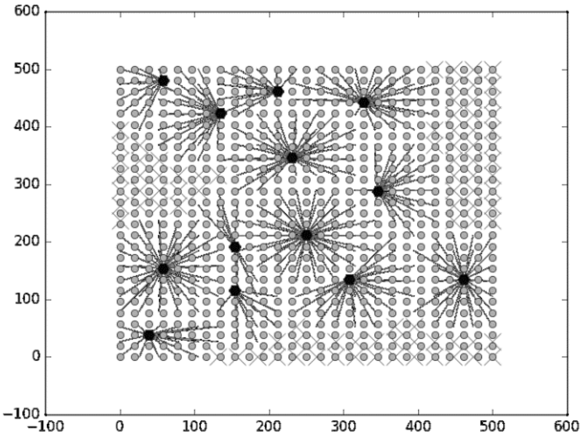

Figure 10. Deployment solution computed by genetic algorithm in 5 epochs

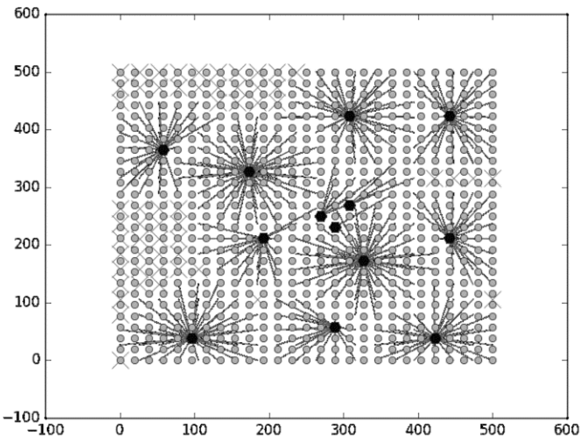

Figure 11. Deployment solution computed by genetic algorithm in 10 epochs

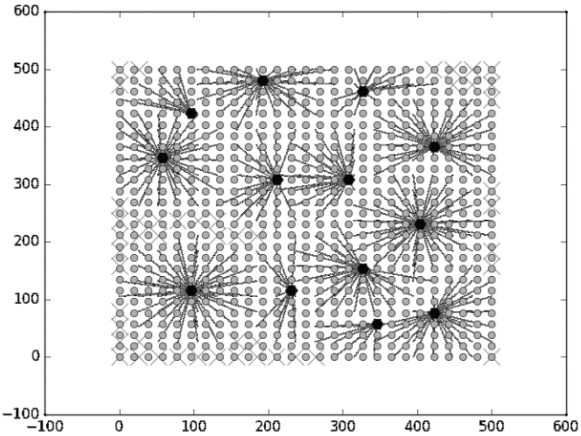

Figure 12. Deployment solution computed by genetic algorithm in 20 epochs

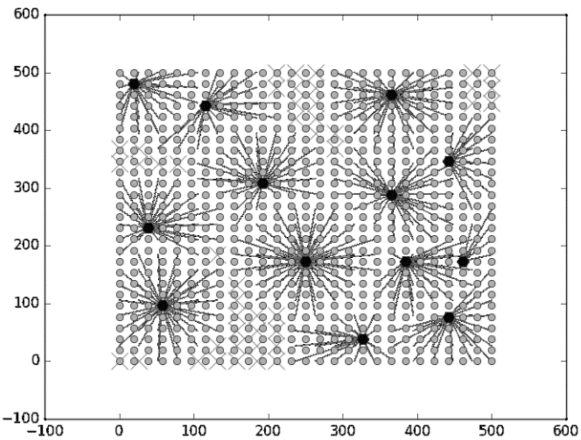

Figure 13. Deployment solution computed by genetic algorithm in 1000 epochs

The research results are presented in the Table 1. It features detection rate, execution time and the number of epochs are presented. Tests with RSA have shown that detection rate of at least 95% can be achieved.

Table 1. Results from the genetic algorithm for different number of epochs

Id	Epochs	Time [s]	Detection [%]	Figure
1	5	0.33	78	10
2	10	0.51	86	11
3	20	0.99	88	12
4	1000	45.50	89	13

The first test was performed for a limited number of epochs. As can be seen in the Figure 10, the deployment turned out to be, in a significant degree, focused in the center. Many test points near the boundaries were omitted so that the achieved detection rate was under 80%. In a small number of epochs, the algorithm did not produced a satisfying result. As it was shown by the second test, the increasing number of epochs, improved the result. However, further increase did not provide better detection rate of the solution. Even 1000 epochs, did not created optimal solution. The main problem of the applied genetic algorithm is a significant impact of random methods. The initial population, selection, crossover and mutation operators all have a randomness factor. This results in high probability of falling into local minima. As for execution time, 20 epochs were completed in less than one minute. This is dependent mainly on the population size, however a suboptimal solution can be rapidly found without performing long computations. This was not enabled by exhaustive search algorithm. In the next subsection, a research on the impact of the initial population size will be presented.

C. Population size impact

The initial population size has a great impact on the execution time and output solution. In this research, 27x27 lattice, sensing range of 100, 20 epochs are assumed. The output from the simulations is presented in the Figure 14, Figure 15, Figure 16 and Figure 17. The initial population size was set consecutively to 10, 20, 50, 200 individuals. The deployment of 13 sensors is assumed.

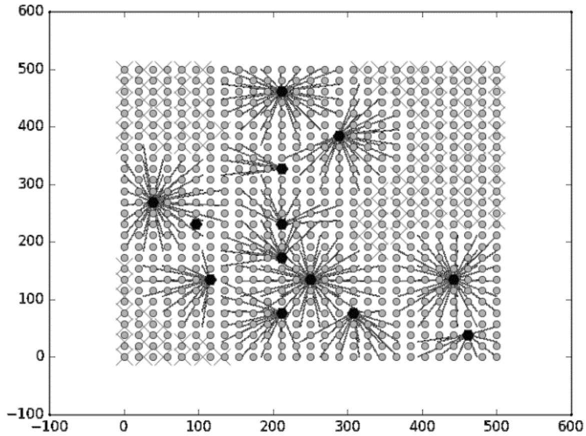

Figure 14. Deployment solution computed by genetic algorithm with population size 10

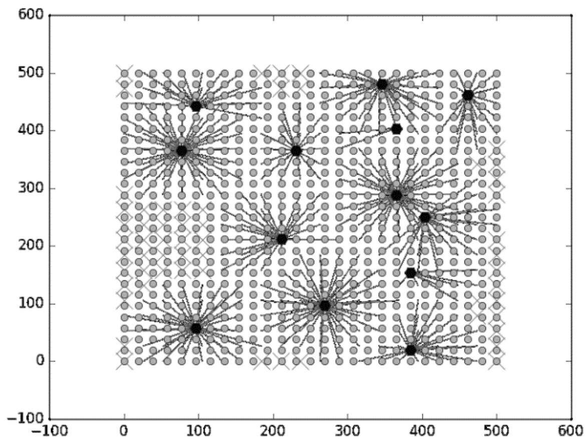

Figure 15. Deployment solution computed by genetic algorithm with population size 20

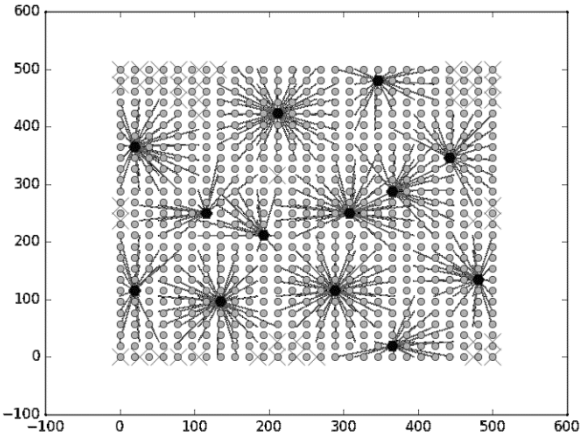

Figure 16. Deployment solution computed by genetic algorithm with population size 50

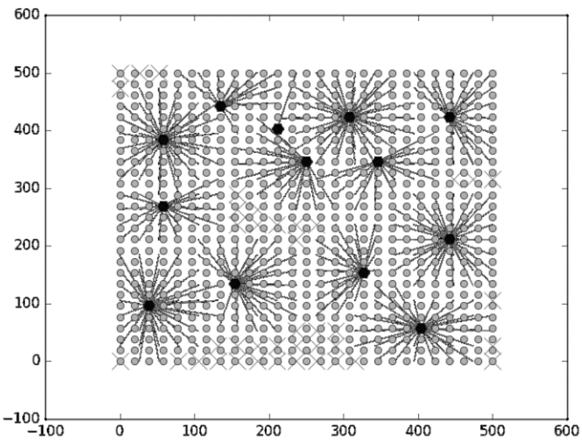

Figure 17. Deployment solution computed by genetic algorithm with population size 200

The numerical results are summarized in the Table 2. Detection rates, execution time and population size are presented.

In conclusion, using a larger population provides better results. On the other hand, the execution of the algorithm takes a longer period of time. In case of small initial population, the results were not satisfying.

Table 2. Results from the genetic algorithm for different population size

Id	Population	Time [s]	Detection [%]	Figure
1	10	0.16	69	14
2	20	0.33	86	15
3	50	0.95	89	16
4	200	4.76	90	17

D. Determining the minimal number of sensors for blanket coverage

Determining the minimal number of sensors, that is required in order to cover the entire area of interest is a common problem of the deployment strategy. Usually, the network cost has to be reduced as much as possible. By applying previously described test and designed genetic algorithm for sensors deployment, the optimal number of sensors can be computed. For this research, the initial population of 50 individuals, 40 epochs and 27x27 lattice is assumed. A sample deployment for different number of sensors is presented in the Figure 18, Figure 19 and Figure 20.

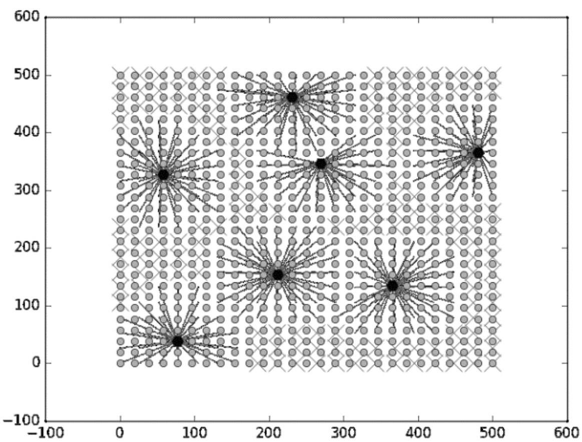

Figure 18. Deployment solution for 7 sensors network computed by genetic algorithm

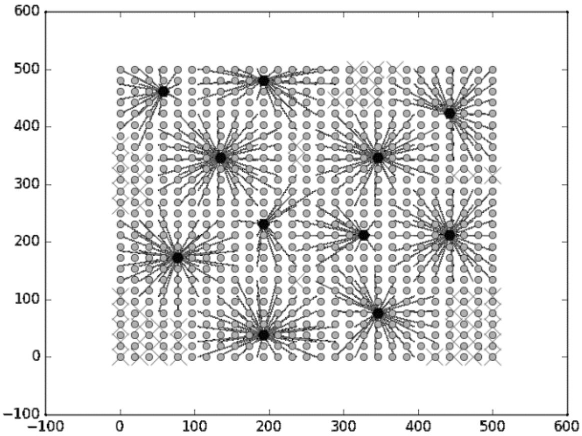

Figure 19. Deployment solution for 11 sensors network computed by genetic algorithm

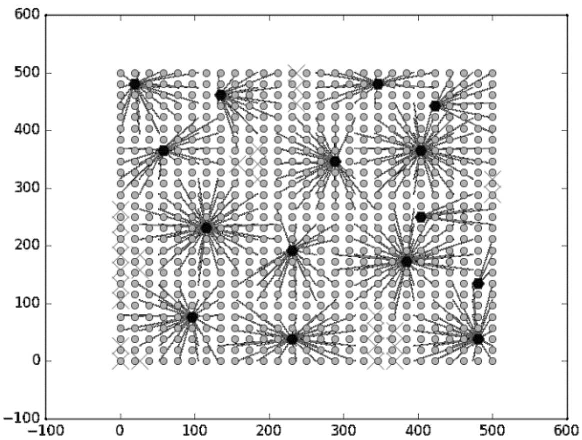

Figure 20. Deployment solution for 15 sensors network computed by genetic algorithm

A summary of the simulations is presented in the Table 3. Detection rates, execution time and number of sensors are shown.

The detection rate of at least 90% is assumed to be satisfying for the deployment solution. The more sensors are used, the better detection rate the network achieved. As a conclusion, it can be said that number of sensors has little impact on execution time. At the current stage, there is no evaluation of redundancy, however it will be done as part of the future work. The genetic algorithm in current form was successfully used in determining the minimal number of sensors needed for blanket coverage of the given area.

Table 3. Detection rate for different number of sensors deployed by genetic algorithm

Id	Sensors	Time [s]	Detection [%]	Fig
1	6	0.92	0.53	-
2	7	1.01	0.65	18
3	8	1.14	0.73	-
4	9	1.25	0.75	-
5	10	1.39	0.80	-
6	11	1.63	0.87	19
7	12	1.74	0.88	-
8	13	1.83	0.90	-
9	14	2.09	0.91	-
10	15	2.27	0.93	20
11	16	2.51	0.94	-

E. Exhaustive search time efficiency evaluation

As a comparison to genetic algorithm results, the time efficiency of the ESA will be described in this section. Due to the long computations, only 7x7 and 9x9 lattices are used in this research. Moreover, only 3 and 5 sensors networks are tested. The results are summarized in Table 4. C stands for number of combinations in thousands and TPC for time per combination in milliseconds. A sample result (3 sensors, 9x9 lattice) is shown in the Figure 21.

Table 4. Results from the exhaustive search algorithm tests

Id	Sensors	Lattice	Time [s]	C	TPC
1	3	7	1.65	18	0.09
2	3	9	8.14	85	0.09
3	5	7	292	1906	0.15
4	5	9	3804	25621	0.15

As can be seen in the table, even a simple task of 5 sensors deployment in 9x9 lattice requires an immense amount of computations. The genetic algorithm, on the other hand, rapidly provides a suboptimal solution, which can be used as an online deployment method when considering network of mobile'sensors.

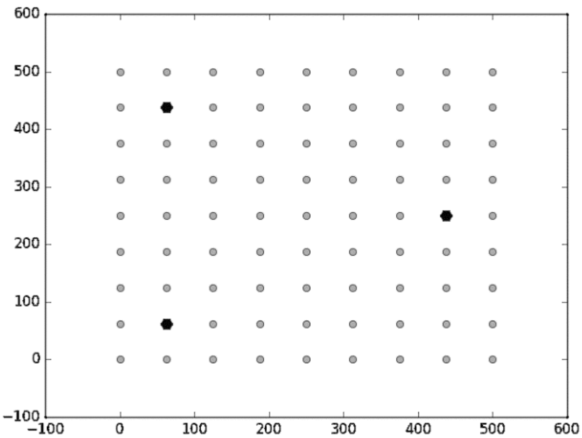

Figure 21. Deployment solution for 3 sensors computed by exhaustive search algorithm

F. Random search algorithm results

In addition to previous research, several solutions from the random search algorithm will be presented in this section. The deployment of 13 sensors in 27x27 lattice is assumed. The Table 5 presents the solution for 10, 1000, 2000 and 10000 iterations of the algorithm.

Table 5. Results from the random search algorithm tests

Id	Iterations	Time [s]	Detection [%]	Figure
1	10	0.018	0.80	22
2	1000	1.667	0.89	23
3	10000	15.79	0.83	24
4	100000	164.37	0.94	25

In a conclusion, increasing the number of iterations of random search does not ensure increasing detection rate. In general, RSA is faster than exhaustive search but the genetic algorithm provides better result in fewer iterations. Moreover, GA is more stable. In the random search, there is no control over the optimization process. Following Figure 22, Figure 23, Figure 24 and Figure 25 show sample random search deployments.

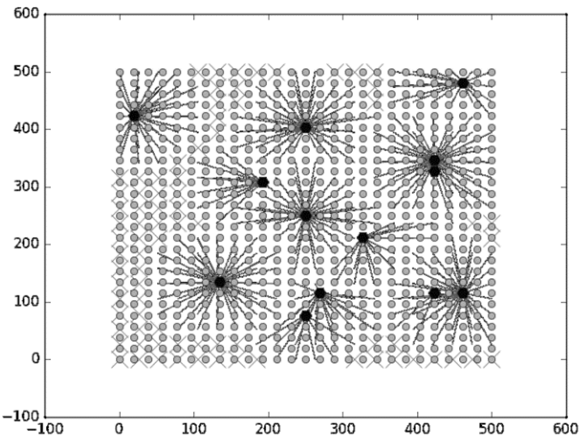

Figure 22. Deployment solution for 13 sensors computed by random search algorithm in 10 iterations

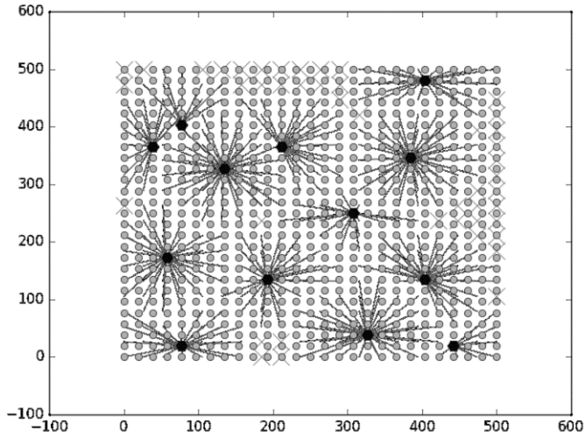

Figure 23. Deployment solution for 13 sensors computed by random search algorithm in 1000 iterations

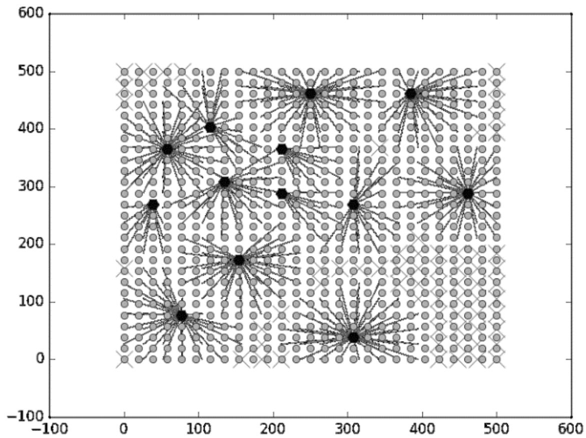

Figure 24. Deployment solution for 13 sensors computed by random search algorithm in 10000 iterations

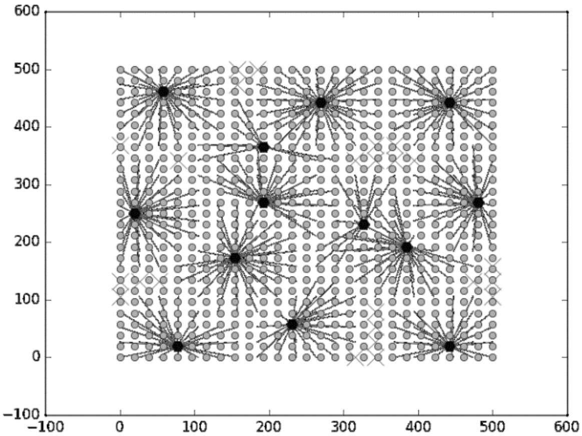

Figure 25. Deployment solution for 13 sensors computed by random search algorithm in 100000 iterations

7. Conclusions and Future Work

A. Conclusions

As a part of the research described in this paper, the algorithms for sensors deployment was designed and implemented. As a preliminary work, exhaustive search and random algorithms were implemented and objective function for blanket coverage was defined. Subsequently, after testing objective function, genetic algorithm was designed. Basic crossover and mutation operators were implemented. Moreover, tournament selection was chosen. After specifying validation method, the genetic algorithm was tested in different cases. It occurred that we can manipulate the algorithm's efficiency by changing several factors. Population size, the number of epochs, all have an impact on the output and execution time. The main problem of this method is that it usually finds only suboptimal solution and stops in the local minima. However, GA in its preliminary form provided satisfactory results. It occurred to be time-efficient and stable. Moreover, it can be used as online deployment method for mobile sensors.

B. Future work

During the research, only basic forms of the mutation and crossover operators were used. In conclusion, they produced sufficient results and were very simple in the implementation; the program code can be kept efficient and readable. However, there are many different variants of them that can be used in genetic algorithms. Future research will include testing new operators variants. Moreover, a mix of the variants will be tested. For example, in the future algorithm, there will be two different mutation operators, one that is currently used, and different that changes the mutated individual in a random way instead of generating new one.

In addition, as a part of the future work, different selection operators will be tested (e.g. roulette wheel selection). This operators are one of the most critical part of the genetic algorithm. Without it, they are just random methods of optimization. By using right selection method, the research is focused on promising areas of search space. What is more, algorithm vulnerability for falling into local minima can be reduced by this operator[18].

Moreover, an additional research will be done in order to check combinations of the designed algorithms, specifically random search and genetic algorithm. For example, RSA will be used in order to create initial population with good fitness rating.

What is more, research showed that current genetic algorithm for sensor network deployment usually finds the suboptimal solution. As part of the future work, special algorithm that improves output of the GA will be designed and tested.

Finally, as noted above in chapter IV, there are many deployment strategies. There will be research made on designing objective function that specifies different deployment strategies, especially barrier coverage. Genetic algorithm in the current form will be also tested in that case in order to find out if it is suitable for different scenarios.

8. Implementation Notification

All algorithms have been implemented in Python programming language. For matrix operations and lattice generation *numpy* package was used. *Numpy* is a package for numerical computations. It is commonly used in

[18] K. Jebari, M. Madiafi, *Selection Methods for Genetic Algorithms*, Int. J. Emerg. Sci., 2013, vol. 3, no. Issue: 4, p. 333-344.

research applications due to its simplicity and efficiency. Each figure was generated by using *matplotlib* package, which has implemented methods and functions for drawing plots and figures. All researched algorithms will be incrementally collected into modular-designed framework for sensor network deployment and simulation.

References:

1. B. K. a. C. C. J. K. Jae-Joon Lee, *Impact of heterogeneous deployment on lifetime sensing coverage in sensor networks*, "Sensor and Ad Hoc Communications and Networks", 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on, 2004.
2. Beutel J., Römer K., Ringwald M., Woehrle M., *Deployment Techniques for Sensor Networks*, "Signals and Communication Technology", October 2009.
3. Cardei M., Andwu J., *Coverage in wireless sensor networks*, "Handbook of Sensor Networks", 2004, p. Chapter 19.
4. Dhillon S. S., Chakrabarty K., *Sensor placement for effective coverage and surveillance in distributed sensor networks*, "IEEE Wireless Communications and Networking Conference (WCNC)", 2003.
5. Ebrahimian N., Sheramin G. Y., Navin A. H., Foruzandeh Z., *A Novel Approach for Efficient k-Coverage in Wireless Sensor Networks by Using Genetic Algorithm*, "Computational Intelligence and Communication Networks (CICN)", 2010 International Conference on, 2010.
6. Erdelj M., Razafindralambo T., Simplot-Ryl D., *Covering Points of Interest with Mobile Sensors*, "IEEE Transactions on Parallel and Distributed Systems, Institute of Electrical", 2013, vol. 24, no. Issue: 1.
7. Gage D. W., *Command Control For Many-Robot Systems*, "Proceedings of AUVS-92", vol. 10, no. Issue: 4, 22-24 June 1992.
8. Guptaa S. K., Kuila P., Jana P. K., *Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks*, "Computers & Electrical Engineering, November 2015.
9. Izadi D., Abawajy J. H., Ghanavati S., Herawan T., *A Data Fusion Method in Wireless Sensor Networks*, "Sensors" (Basel, Switzerland), 2015, vol. 15, no. Issue: 2.
10. J. S. S. Z. L. L. a. L. S. Y. Chen, "Data Fusion in Wireless Sensor Networks," Electronic Commerce and Security, 2009. ISECS '09. Second International Symposium on, 2009.

11. Jebari K., Madiafi M., *Selection Methods for Genetic Algorithms*, Int. J. Emerg. Sci., December 2013, vol. 3, no. Issue: 4.
12. Kim H., Han S.-w., *An Efficient Sensor Deployment Scheme for Large-Scale Wireless Sensor Networks*, “IEEE Communications Letters”, 07 January 2015, Volume: 19, Issue: 1.
13. Kumar S., Lai T. H., Arora A., *Barrier Coverage With Wireless Sensors*, MobiCom’05, 2005.
14. Lazos L., Poovendran R., *Coverage In Heterogeneous Sensor Networks*, “Network Security Laboratory (NSL)”, Department of Electrical Engineering, 2006.
15. Liu B., Towsley D., *A study of the coverage of large-scale sensor networks*, “IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS)”, 2004.
16. McCall J., *Genetic Algorithms for Modelling And Optimisation*, “Journal of Computational and Applied Mathematics”, December 2005, vol. 184, no. Issue: 1.
17. Savkin A. V., Javed F., Matveev A. S., *Optimal Distributed Blanket Coverage Self-Deployment of Mobile Wireless Sensor Networks*, “IEEE Communications Letters”, June 2012, vol. 16, no. Issue: 6.
18. Seo J.-H., Kim Y.-H., Ryou H.-B., Jo M., *Optimal Sensor Deployment for Wireless Surveillance Sensor Networks by a Hybrid Steady-State Genetic Algorithm*, “IEICE Transactions on Communications E91-B(11)”, November 2008.
19. Sharmaa V., Patelb R., Bhadauriaa H., Prasadc D., *Deployment schemes in wireless sensor network to achieve blanket coverage in large-scale open area: A review*, “Egyptian Informatics Journal”, March 2016, vol. 17, no. Issue 1.
20. Vershinin Y. A., *A Data Fusion Algorithm for Multisensor Systems*, “Information Fusion”, 2002. Proceedings of the Fifth International Conference on, vol. 1, July 2002.
21. Wang B., *Coverage problems in sensor networks: A survey*, “ACM Computing Surveys (CSUR)”, October 2011, vol. 43, no. Issue 4, Article No. 32.
22. Wang Z., *Barrier Coverage in Wireless Sensor Networks*, PhD diss., University of Tennessee, 2014.
23. Xu Y., Yao X., *A Ga Approach To The Optimal Placement Of Sensors In Wireless Sensor Networks With Obstacles And Preferences*, CCNC 2006.

2006 3rd IEEE Consumer Communications and Networking Conference, 8-10 Jan. 2006, vol. 1.
24. Yoon Y., Kim Y.-H., *An Efficient Genetic Algorithm for Maximum Coverage Deployment in Wireless Sensor Networks*, “IEEE Transactions on Cybernetics”, 1483, 11 September 2013, vol. 43, no. Issue: 5.
25. Zhao C., Yu Z., Chen P., *Optimal Deployment of Nodes Based on Genetic Algorithm in Heterogeneous Sensor Networks*, “IEEE International Conference on Wireless Communications”, 2007.

Author

Łukasz Szklarski – studied at Poznań University of Economics at the Faculty of Informatics and Electronic Economy. He received a Master’s degree in the field of IT. Since 2007 he has been working at ITTI Sp. z o.o. in Poznań. He is an experienced project manager with extensive expertise in the field of sensor technologies for surveillance and biometric identification, IT systems, and decision support systems. Moreover, he is a member of PMI association with ITIL Expert Certificate and PRINCE2 Certificate. Łukasz has also profound knowledge of the a range of modern security and military technologies , threat assessment and decision/perception support. In the recent years, he has participated in the 7th FP projects (e.g. TALOS , ASPIS, SICMA, ARENA, DESTRIERO, SECTOR, FASTPASS) as well as projects for European Defence Agency (EDA) e.g. CARDINAL, SIMS, UGELAS, CENSIT. Additionally, he specializes in economic forecasting and has in-depth understanding of quality systems: CRM, ERP, PLM and range of modern military technologies (projects for EDA).

Cite this article as:

Ł. Szklarski, *Sensor Network Deployment Optimization for Improved Area Coverage Using a Genetic Algorithm,* “Security Dimensions. International and National Studies”, 2016, no 19, p. 150–181.