FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Maksym Huk

Algorithms for indoor wireless sensor system placement

Department of Theoretical Computer Science and Mathematical Logic

Supervisor of the bachelor thesis: RNDr. Jiří Fink, Ph.D.
Study programme: Computer Science
Study branch: General Computer Science

Prague 2018

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date .... signature of the author

Title: Algorithms for indoor wireless sensor system placement

Author: Maksym Huk

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: RNDr. Jiří Fink, Ph.D., Department of Theoretical Computer Science and Mathematical Logic

Abstract: Genetic algorithm, which imitates the process of natural selection, has shown to be efficient search technique for optimization problems that often do not have polynomial time algorithm. One of such problems is an optimal placement of the Wireless sensor network indoors in complex office spaces. In this thesis, we create a program which allows the user to set desired settings of the network and floorplan of the building in order to receive back optimal placement. It is also possible to compare results between different genetic methods. In addition, we make an overview of papers in this area.

Keywords: Wireless sensor network, genetic algorithm

I wish to thank RNDr. Jiří Fink, Ph.D. for supervising this thesis. Also, I would like to express gratitude to my parents and grandfather for their support. Without them, this work would not have been possible.

Contents

1. Introduction

Public safety inside the building is ensured by the set of measures, an important components of identification of threats and their neutralization are active disaster response systems [1]. They are required in any commercial property, especially on the factories and industrial enterprises, in banks and molls, as well as educational, medical and government institutions.

Emergency systems ensure the stability of the organization working process, performing various functions of protection and control. It is security systems that can cope with the prevention of fire-hazardous and other emergencies, life-threatening and health-threatening situations; provide real-time monitoring, carry out notification and take urgent measures in case of emergency.

Previously, security systems were scattered sensors to detect a specific emergency that triggered an alert or response system.

Modern building emergency systems are automatic control systems with a multi-level network structure, which has a common control center based on a local computer network. The emergency system contains communication lines, information receiver controllers and other peripherals designed to collect and process information from various sensors (fire and burglar alarms, air quality and humidity sensors). Automated systems also perform the functions of centralized control of various response tools (warning lights, sprinklers, engineering systems).

Nowadays Wireless sensor network(WSN) is a promising tool for exploring the physical world. Important advantages of WSN are the lack of cable infrastructure, small units, low power consumption, built-in radio interface, high enough computing power and relatively low cost. The Wireless sensor network is a distributed system of non-serviced miniature electronic devices (network nodes) that collect data about environmental parameters and transfer them to the server by node to node communication using wireless connectivity. Several nodes of such a network act as gateways (hubs) that perform communication with the global network. Management of all nodes, as well as decision-making in case of an unusual situation, is carried out by the server (information processing center).

The basis of the network is made up of sensors that can measure physical parameters of the environment, for example, temperature, pressure, smoke, and humidity. In addition, the sensor contains a microcontroller, memory, a radio transmitter, an autonomous power source and sometimes actuators.

The network based on mesh topology is characterized by high reliability, high bandwidth, and reduced power consumption. High reliability is provided by the redundancy of nodes (if one node fails, the data will be transmitted bypass, along a different path). Using multiple alternative routes increases the network bandwidth. Reducing power consumption is achieved by reducing the power of signals and data transfer over a greater number of nodes, separated by smaller distances. All this makes WSN wide application possible in many spheres of human activity for the automatization of information gathering, various technical and natural objects monitoring [2].

The sensor nodes of the network can be fixed permanently, but they can also have some mobility, so they can arbitrarily move relative to each other in some space without violating the logical connectivity of the network. In the latter case, the sensor network does not have a fixed topology, and its structure dynamically changes [3].

Moreover, the nodes must perform data processing from the sensors, decide whether to transfer information to the gateway or not and determine the subset of the information to be transmitted. Nodes can also act as hubs that provide transfer of information from territorially remote sensors to gateway nodes. Networks in which nodes perform different roles belong to heterogeneous networks. When designing the network, in addition to locating the nodes, it is necessary to assign an appropriate role to each node of the network.

It is also necessary to take into account limitations of energy consumption, computational capabilities and throughput capacity of radio channels. These aspects can limit the network's scalability, requiring to use more gateways in order to cover a larger area of observation [4].

Thus, important requirements for the WSN are the ability to scale area and number of sensors, the reliability of the system, optimal placement of stationary sensors, the definition of roles among sensors, low power consumption, self-organization, joint signal processing, ability to work on request. Moreover, wireless sensor networks are required to ensure the certain reliability of operation, especially in areas where the failure or delay in the delivery of information about a particular phenomenon can have critical consequences.

All these aspects make it difficult to design an effective architecture and often require the development of specialized solutions. Based on the formulated network requirements, it becomes necessary to solve the problems of the optimal devices placement indoors, distribution of roles between devices, ensuring the stability and reliability of the network.

Since the optimization problem of the placement of the device indoors is computational hard we implement efficient search technique to ensure that restrictions on coverage and robustness are met. For that, we use the Genetic algorithm (GA) with some genetic operators, make the comparison of these operators and propose their best combination for given problem. We also make an overview of papers in this area and compare received results with some of them.

Protocols and physical models of radio communication between the devices of active response system, as well as literature overview can be found in Chapter 2. Genetic algorithm with all used operators is described in the details in Chapter 3. Chapter 4 is dedicated to designing and implementation of the created application. Received results from the experiments are compared in Chapter 5.

The digital version of this thesis also contains the program which was created for solving a particular optimization problem using GA. Appendix A explains content and structure of digital bundle.

2. Background

This chapter gives an idea about what the Wireless sensor network is (Section 2.2) and provides the literature overview (Section 2.2).

2.1 Wireless sensor network

More particullary we discuss roles and mission of devices in the network (Subsection 2.1.1), physical constraints (Subsection 2.1.2) and radiomodels used to estimate connectivity (Subsection 2.1.3).

2.1.1 Overview

Radio frequency (RF) module usually works in 868 MHz or 2.4 GHz frequency range, supporting one of the main high-level network protocols for machine-to-machine communication (Zigbee, Thread, Z-Wawe) [5]. These standards mainly support pair, cluster tree, star and mesh topologies. The most promising for building fault-tolerant, secure and scalable network is a mesh topology (Figure 2.1) in which all components cooperate with one another for the data distribution and transmission. The automatic configuration makes this network robust and adaptive. To maximize the performance of data transmission within the network, it is proposed to compute a channel metric, based on the bandwidth, signal strength, its stability, latency or other parameters.

There are 3 main roles of devices in the mesh network:

- Leaders
- Hubs
- Sensors

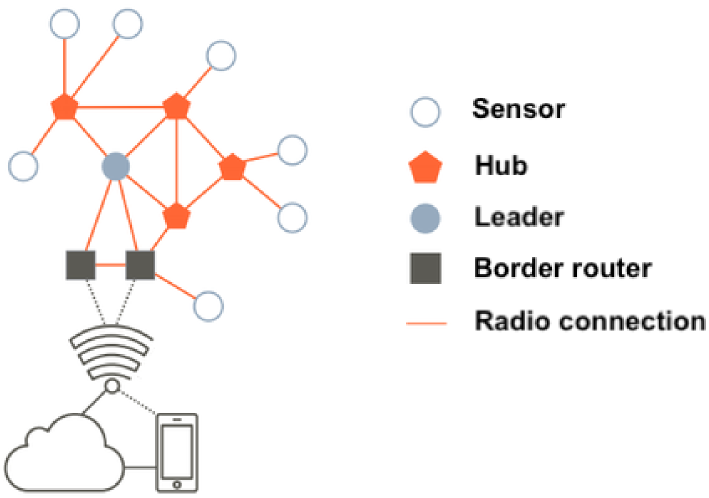


Figure 2.1: Network architecture

Leaders control the process of network formation and security, specifically allow connection of new nodes, encrypting data, assign roles within the network between other devices. There can be only one leader at the same time. Hubs expand the network range, in case leader disconnects - one of them takes his role. Sensors are equipped with environment parameter detectors, can communicate with only one chosen hub simultaneously.

Addressing is done with the protocol IPv6, and the interaction is based on top of the standard for low-power wireless personal area networks IEEE 802.15.4 [5]. For communication with a global network, one or more hubs are connected to the Internet via any available channel. To ensure the security of data transmission, encryption AES-128 standard is used as well as the system for identifying devices in the network.

2.1.2 Physical constraints

Because each device has limited sensor detection radius, it is required to place a large number of devices for full area coverage. Moreover, devices are usually battery-powered and it is hard or even impossible to charge batteries often. There are several straightforward approaches for nodes' placement and battery distribution management, which do not guarantee good results. Namely, Direct Transmission and Minimum Transmission Energy, following which may result in fast discharging of sensors, placed far from hubs (because of huge transmission power needed); or hubs, placed close to the leader (because they work almost all the time as relays for other nodes). The material of walls, their thickness, mutual position of devices are also playing a big role on battery consumption and network stability.

Mesh networks usually do not have a single point of failure. If hub disconnects, all its sensors try to reconnect to another one, while other hubs consider new ways of communication between each other. If leader losses connection new device takes its role.

Therefore, if robustness of network is one of the goals, there is a need in an algorithm for optimal mesh network nodes' placement on the floor map of the building.

2.1.3 Wireless data link signalling models

One of the possible ways how to obtain information about radio signal propagation is to build a spatial signal propagation model. Using an appropriate mathematical model of the signal propagation we can determine the distance at which probability of erroneous data packet receipt is less than some preset acceptable error probability and thus determine the reliable radio signal reception distance. The connection between the network nodes is established, and the maximum data transmission rate is ensured in the primary service area.

The data link modeling task is determining the maximum distance at which the wireless mesh nodes can be located and which ensures the connection reliability. Usually, the threshold value is 1% probability of an erroneous data packet reception which ensures 99% connection reliability.

Friis Model

A lognormal Friis Model with attenuation is used for describing the signal propagation process in the paper [6]. We can describe it using following notation: $\overline{\mathrm{PL}}(d)$ is the average attenuation of the signal at the distance of d in dBm, $\mathrm{PL}(d_0)$ is the average attenuation of the signal at the distance of $d_0 = 1m$, n is the rate of transmission loss, $X_\sigma(0, \sigma)$ is Gaussian random variable with zero mathematical expectation and standard deviation σ.

$$\overline{\mathrm{PL}}(d) = \mathrm{PL}(d_0) + n * 10 \log \frac{d}{d_0} + X_\sigma(0, \sigma) \tag{2.1}$$

The signal attenuation at the distance of $d_0 = 1m$ with the carrier signal wavelength λ can be calculated using Friis equation:

$$\mathrm{PL}(d_0) = 20 \log \frac{4\pi d_0}{\lambda} \tag{2.2}$$

It is assumed that at distances not exceeding one meter the signal propagates just like in the open space model. Let us determine the carrier signal wavelength at frequency $f = 868 * 10^6$ Hz and speed of sound $c = 300 * 10^6 \frac{m}{s}$ as $\lambda = \frac{c}{f} = \frac{300}{868} = 0.346$ m. Then the signal attenuation at the distance of d_0 will be equal to

$$\mathrm{PL}(d_0) = 20 \log \frac{4\pi}{0.346} = 31.198 \text{ dBm} \tag{2.3}$$

Let us assume that a random variable X_σ determines all dynamic changes in the environment as well as the multipath signal transmission. Parameters n and σ for this model are set based on the experimental environment data. Based on the set transmitter power P_t and receiver sensitivity P_s, as well as the calculated value of the signal attenuation, the signal-to-noise ratio γ_{dB} is calculated in dB at some distance at the receiver input.

$$\gamma_{dB} = \mathrm{P}_t - \overline{\mathrm{PL}}(d) - \mathrm{P}_s$$

This value can be used to determine the probability of erroneous receipt of one bit. The VirtualWire standard is designed for transmission and reception of short messages using broadcast wireless communication and amplitude modulation [7]. The probability of erroneous amplitude modulation reception:

$$\mathrm{P} = 0.5 e^{-(\frac{\gamma_{dB}}{4})} 0.5 Z(\sqrt{\frac{\gamma_{dB}}{2}}) \tag{2.4}$$

where $Z(\sqrt{\frac{\gamma_{dB}}{2}})$ is the additional Laplace error function.

Probability of erroneous reception of one data bit is $\mathrm{PER} = 1 - (1 - \mathrm{P})$. Based on this, one can determine the probability of erroneous reception of 8-byte long data packet (it is enough for transmission of data from sensors) as

$$\mathrm{PER} = 1 - (1 - \mathrm{P})^{8*8} \tag{2.5}$$

The Figure 2.2 illustrates the erroneous reception of the VirtualWire data packet with amplitude modulation as a function of signal-to-noise ratio obtained using relationships (2.4) and (2.5). Using the graph in the Figure 2.2, one can determine the signal-to-noise value at which the probability of erroneous data

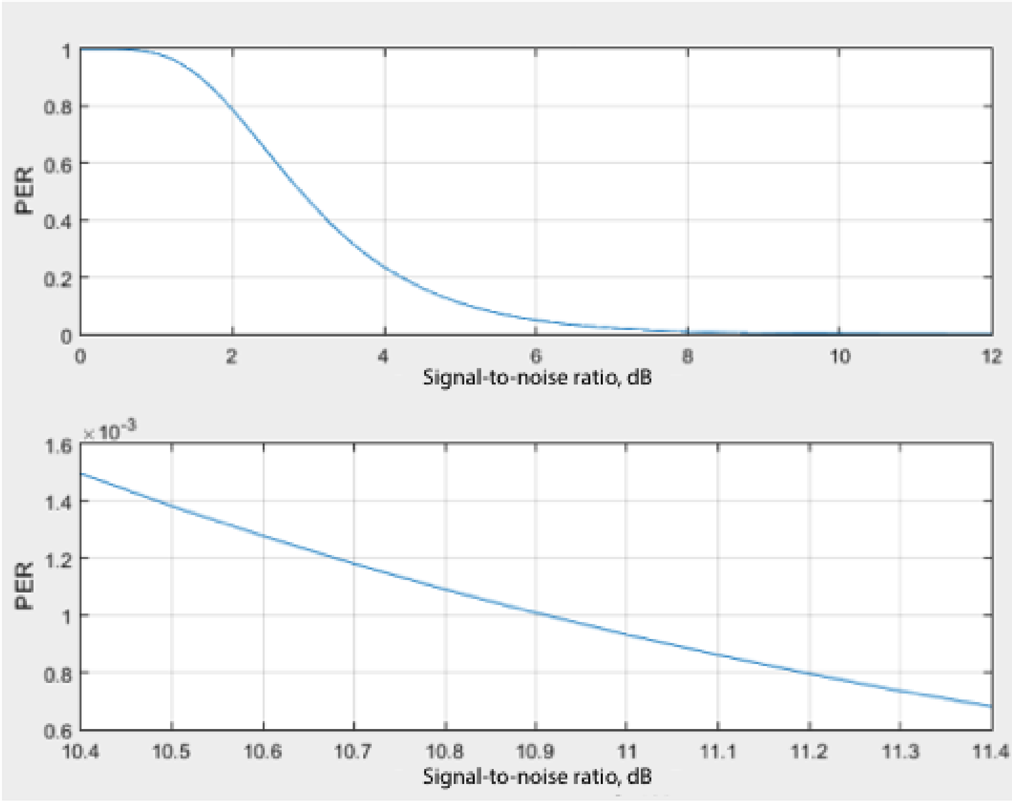


Figure 2.2: Probability of erroneous data packet reception

packet reception is 1%. This value is 10.9 dB. Thus, it can be concluded that at signal-to-noise ratios exceeding 10.9 dB the connection between two nodes of wireless smart sensor network ensures 99% reliability.

The logarithmic model of signal attenuation at $n = 4$ (the value corresponding to the flat Earth model), in accordance with relationships (2.1), (2.2) and (2.3), can be described as:

$$\overline{\mathrm{PL}}(d) = 31.198 + 40log(d) + X_\sigma(0, \sigma) \tag{2.6}$$

The reliable radio signal reception distance can be determined by setting definite receiver sensitivity and transmitter power values. The selected model can be used for designing a wireless data link in premises or space where the multipath transmission effect occurs.

Keenan-Motley Model

Since Friis model is describing the signal propagation on the line-of-sight, without any obstacles between the transmitter and receiver and our primary goal is to describe placement of the devices indoors, we have to introduce second radio model.

This radio model should count with walls between the devices and adjust reliable radio signal reception distance according to it. For this purpose we have chosen Keenan-Motley Model [8], empirical model which includes the free space path loss (2.1) with additional loss due to the walls and floors. Because we are optimizing only one-level floorplan, the path loss for this model is going to be simplified to

$$\mathrm{PL}_{KM}(d) = \overline{\mathrm{PL}}(d) + n_w L_w, \tag{2.7}$$

where $\overline{\mathrm{PL}}(d)$ - free space path loss, d - distance between the devices, n_w - number of walls between them and L_w - Wall attenuation factor (WAF).

Signal attenuation is computed similarly to (2.6) but using (2.7) as path loss.

The main advantage of this model is that it is well-studied. This means that we can find tables with WAF for a lot of different materials, wall thicknesses and radio frequencies.

For our particular case, 3 was chosen as WAF for the modern office building walls. Unfortunately, it is quite time-consuming to compute this radio model for big floorplans and dense grids so the Friis model was picked as a primary radio model.

2.2 Literature overview

A lot of papers has been dedicated to design and analyze topology of WSN.

In the literature, the case of modeling networks with fixed position is widely represented e.g. [9, 10]. In this case, algorithms of graph theory are used to solve the problem. The solution of problems in this formulation is sought by constructing minimum spanning trees, Steiner trees, Hamiltonian paths in the graph. However, this approach can be used only if the weights of the links connecting the devices are known.

Another approach to design a network topology are optimization problems. As a criterion of optimality, we can use the cost of equipment or the stability of the signal in the coverage area which, however, does not make it possible to ensure compliance with other criteria, such as e.g. network robustness.

When designing the optimal network topology, the problem can be formulated as a multicriteria optimization problem. The criteria convolution method or main criterion method are used for the problem linearization [11]. The brute-force algorithm presented in this paper considers all possibilities of the solution and chooses from them the one which corresponds to the extremum of the criterial function.

The network topology design problem can also be solved using a facility location problem, or a covering problem which are described in the following papers.

In the placement problem [9], the initial data are the sets of sensors and hubs, the maximum possible radius of the wireless connection between the sensor and the hub, distance function between the sensor and the hub. It is necessary to find such placement of hubs that they form a set of minimum power, while for each sensor the distance to the hub should be at most the radius of the wireless connection.

The coverage problem is formulated as a problem of defining a system of subsets of devices with assigned weights such that their union coincides with a given set. It is required to find a coverage of the minimum total weight.

Given the limitations on the number of sensors connected to the hub, the coverage problem can be reduced to the linear programming problem and then solved by the greedy algorithm e.g. [9]. The introduced modification of the greedy algorithm is aimed to reduce the power of the solution sets on the algorithm iterations and to obtain the optimal network topology in polynomial time.

Without taking into account the extinction of the signal in a room, the range of the sensor can be described by a circle. Thus, the coverage problem reduces to the task of constructing circular covers.

The problems of constructing and analyzing circular coverages are discussed in a number of studies [12, 13, 14]. Mentioned articles are dedicated to the creation of accurate mathematical models for solving coverage problems. In the paper [15] a mathematical model of the problem based on Voronoi diagram is constructed. This approach is applicable only to cover polygons by circles of the same radius and requires some additional variables (vertices of Voronoi diagram).

Based on the proposed coverage criterion [16], a mathematical model of the covering problem of an arbitrary polygon with circles of different radiuses is developed. Formalization of the conditions of covering by circles of different radii requires the input of additional variables, which led to a significant dimension increase of the problem.

In the paper [17] an approach based on the construction of the Dirichlet-Voronoi covering on a given initial system of centers with successive improvement is developed. The results of finding the optimal covering by circles of the minimal radius are presented only for a hyperbolic plane and the sphere. The implementation of the algorithm essentially depends on the shape of the area to be covered.

Paper [18] considers a problem of the covering of the set of points as the problem of quasi-differentiable optimization in R^n and an algorithm for solving it in R^2 was presented. There are also restrictions on the form of the covering set.

For a continuous problem of an optimal cover of a compact set with balls stated in paper [19], an algorithm based on the theory of optimal partitioning of sets and the application of the Shore algorithm is proposed and justified. When using the proposed approach, the results of calculations (the minimum radius of the spherical coverage) depend on the parameters of the algorithm, namely the step size of the spatial grid and the value of the step of numerical differentiation for the components of the generalized gradient calculation. The heuristic of finding the optimal solution of the problem decreases with increasing dimension.

It should be noted that in all papers mentioned above, only minimization of the device transmit power is considered as the objective function. Therefore, the application of these approaches does not allow to take into account additional requirements when designing the network structure regarding the distribution of roles between devices, ensuring the stability and reliability of the network.

Usage of precise mathematical models for the covering problem requires significant restrictions on the condition of the problem, which makes it impossible to use in practice. In addition, the use of classical optimization methods is possible only for calculating the topology of relatively small networks due to significant computational difficulties.

Therefore, the most promising direction is approximate (heuristic) optimization method. The merit of such an approach is the possibility to solve large-dimensional problems with relatively small computational costs.

Various heuristic techniques and algorithms substantially reduce the solution space [20].

One of the heuristics for solving the problem of a circular covering is to place the sensors in deterministic points. To split the plane the basic placement template in the form of the grid is used. If the template is completely covered by sensor zones located at its vertices, then the entire region can be covered by a set of such polygons. There are more advanced methods of coverage, based on the placement of sensors in deterministic nodes, using Voronoi diagram [21] or Delaunay triangulation [22].

Another basic heuristic approach to solve circular covering problem is based on the random placement of sensors in the area [23]. In this case, during the stochastic construction of the covering, two subproblems arise: how to determine that the cover is constructed and how to remove the redundant sensors. The basic methods for solving these subtasks are well studied [12].

One of the most effective approaches is the coverage criterion, based on the analysis of the belonging of circle and boundary intersection points [24]. In case each of these points belongs to the area of at least one another sensor, then the coverage is constructed.

Various heuristic optimization methods are often used after random placement of the sensors, namely the method of the ant colony [25], genetic algorithms [26], the simulation of annealing [27], etc. The main aim is to improve the quality of the solution obtained. These methods allow to significantly improve the result of preliminary design and to construct a quasi-optimal covering, satisfying additional conditions.

Recently, many researchers are studying and developing technologies to ensure such optimal placement of sensors that maximizes the coverage area and ensures the connectivity of network nodes.

Paper [28] presents an approach that combines computational geometry and methods of the graph theory (Voronoi diagram and search algorithms on a graph), which allows one to construct a covering in polynomial time. The coverage of the sensor area is characterized by Maximal Breach Path and Maximal Support Path, these parameters can be used for future deployment or reconfiguration of an existing network in order to improve the overall quality of the coverage.

Geometric analysis of the relationship between coverage quality and connectivity of network nodes is presented in the paper [29]. However, the approach proposed does not allow to construct an initial configuration that ensures the optimal coverage of the domain and robust sensor connectivity.

Some researchers use methods of evolutionary computation and optimization to construct a covering. The paper [30] suggests an approach based on a genetic algorithm to the deployment of a mobile sensor network in real time. During the initial planning of the sensor network, their optimal placement is calculated by a genetic algorithm. Then an optimization algorithm is used to distribute the roles of the sensor nodes taking into account their topology [31].

An example of successful application of modified genetic algorithms for WSN design is given in the paper [32]. The proposed algorithm finds the optimal nodes arrangement in the network, which ensures the minimization of consumed energy for transmitting messages. Various approaches to the basic formation, including the cluster's creation in order to reduce the computational complexity of the algorithm, are investigated.

Thus, from the analysis of literature sources, it follows that analytical models and methods of solution have a limited field of application, certain shortcomings which do not allow to solve practical problems. The overwhelming majority of papers related to the problems of circular coverage are studying heuristic methods for their solution. Modern computational technologies, namely the genetic algorithm and the method of ant colony, are used to solve problems of optimal sensor network design. However, the quality functionals formulated in the literature do not take into account all the requirements that are imposed in practice.

In this paper, we propose designing an algorithm that realizes the optimal coverage of a given area and automatic role assignment (sensor, hub) for each node. The resulting configuration should ensure connectivity of the network nodes and robustness of the network itself.

3. Genetic algorithm

Coherence and efficiency of the biological organisms elements suggest the possibility of using the principles of biological evolution in order to optimize systems that are practically important for humans.

In 1975 the fundamental book ”Adaptation in Natural and Artificial Systems” of John Holland [33] was published, where he proposed the genetic algorithm(GA). The algorithm was based on the principles of natural selection by Charles Darwin.

The genetic algorithms are referred to the field of soft computing. The term ”soft computing”, according to Lotfi Zadeh in his work ”Soft computing and fuzzy logic” [34], implies a set of inaccurate, approximate methods of solving problems that often do not have a solution in polynomial time. Such problems have arisen in biology, medicine, humanitarian sciences, and management. The methods of soft computing well complement each other and are often used together. The area of soft computing includes such methods as fuzzy logic, neural networks, probabilistic reasoning, Bayesian network-based trust models, evolutionary algorithms.

The genetic algorithm is a method that reflects a natural evolution of problem-solving techniques, and primarily optimization problems.

Genetic algorithms are search procedures, based on mechanisms of natural selection and inheritance. They use the evolutionary principle of survival of the fittest individuals. They differ from traditional optimization techniques by several basic elements. In particular, GA have a number of distinctive properties:

- parameter coding – genetic algorithms process not values of the parameters of the task itself, but their encoded form;
- operations on population - genetic algorithms search for the solution based not on a single point (the initial approximation) but on some population;
- minimum information about function – they use only the target information, but neither derivatives nor the additional information;
- randomization of information - genetic algorithms apply probabilistic, rather than deterministic, rules of choice.

Bayesian networks of trust are the model of probabilistic and cause-effect relations between variables in statistical information modeling.

The scope of genetic algorithms is basically the optimization of multi-parameter functions. Application of genetic algorithms is very extensive. They are used for software development in artificial intelligence systems, optimization, artificial neural networks and in other fields. It should be noted that genetic algorithms help to solve the tasks for which neural networks were previously used. In this case, the genetic algorithms are simply in the role of the method independent on neural networks, designed to solve the same problem. For instance, the traveling salesman problem, originally solved by a Hopfield network. Genetic algorithms are often used together with neural networks. They can support a neural network or both methods can interact within the same hybrid system, designed to solve a specific problem. Genetic algorithms are also used in conjunction with fuzzy systems.

However, as a general rule genetic algorithms are not the panacea for optimization problems. With high probability, genetic algorithms will show at least not better results comparing to the specially-developed methods for solving specific tasks. A great advantage of evolutionary computation is the unified approach they provide to solve a wide variety of problems.

Genetic algorithms show excellent results in solving complex search problems, most of which are NP-complete, such as the traveling salesman problem or Boolean satisfiability problem [35].

3.1 Basic concepts of genetic algorithms

For describing GA, definitions in a simplified form are borrowed from genetics and the basic concepts of linear algebra are used.

Vector is an ordered set of elements, called components of a vector. A Boolean vector is a vector whose components take values from the two-element set, usually 0 or 1.

The population is a finite set of individuals. The individual is a set of chromosomes coded in them set of problem parameters, so-called points in the search space.

The chromosome is a vector of genes. The chromosome can be represented as a Boolean vector obtained by binary or Gray code. The chromosome is usually denoted as A. A gene is an atomic element of a genotype, in particular, chromosome. It carries the hereditary information. It is denoted by X.

Genotype is a set of chromosomes of a given individual. Consequently, individuals of the population can be either genotype or a single chromosome. A phenotype is the set of values corresponding to a given genotype, it is decoded structure or a set of problem parameters. An allele is the meaning of a particular gene. Locus is the position indicating the placement of the gene in the chromosome.

A very important concept in genetic algorithms is the fitness function, also called the evaluation function. This function plays an important role since it allows to determine for the specific individuals in population the level of their fitness and to choose from them the most adapted according to the evolutionary principle of the survival of the fittest. In optimization problems, the fitness function is usually optimized (mainly it is maximized) and is called the objective function. For the tasks of minimization, the objective function is transformed, and the problem is reduced to maximization.

On every iteration of the genetic algorithm, the fitness of each individual of a given population is evaluated using the fitness function, and based on this the following population of individuals is created constituting the set of solutions of the problem.

3.2 Classical genetic algorithm

Unlike evolution in real nature, GA simulates only the essential for development processes in population.

The most adapted individuals have the opportunity to reproduce offspring with other individuals of the population, which leads to the new individuals that combine certain characteristics inherited from their parents. Less adapted individuals are less likely to reproduce offspring, so the properties they possessed will gradually disappear from a population during evolution.

Thus, the whole new population of permissible solutions is produced by selecting the best representatives of the previous generation, crossing them and getting a set of new individuals. This new generation will contain a higher ratio of characteristics that the fittest members of the previous generation had. As a result, good characteristics spread throughout the entire population.

Crossing the fittest individuals leads to the fact that the most promising areas of the search space are explored. In the end, the population will converge to the optimal solution of the problem.

There are many ways to implement the idea of biological evolution within the framework of the genetic algorithm. The scheme of the genetic algorithm is shown in Figure 3.1.

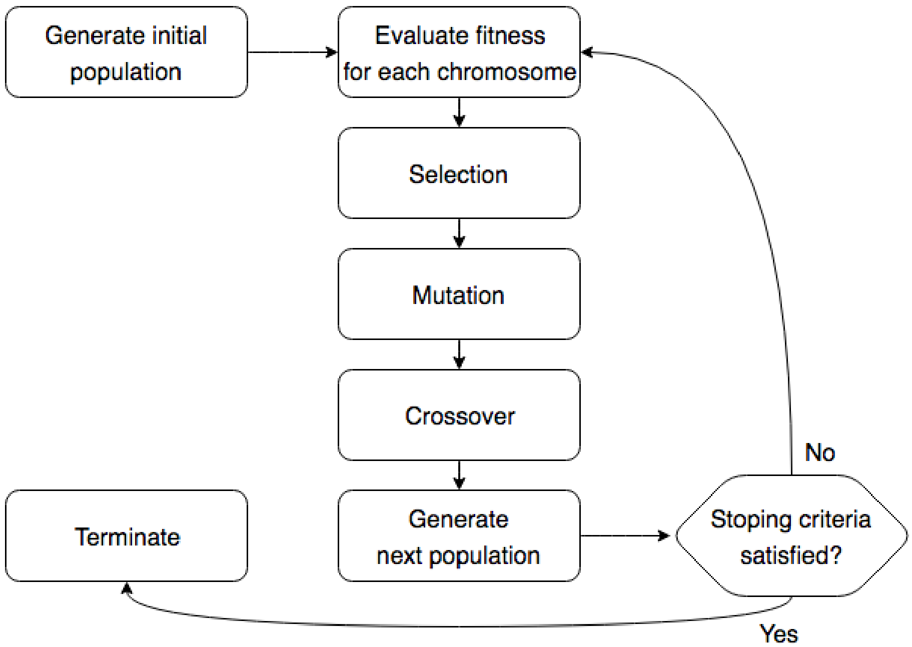


Figure 3.1: Genetic algorithm block diagram

Genetic operators are necessary in order to apply the principles of heredity and variability to the population. Despite some distinctive features, which will be discussed below, all operators do have such a property as the probability. That is the described operators do not necessarily apply to all crossed individuals, which introduces an additional element of uncertainty into the process of finding a solution. In this case, the uncertainty does not imply a negative factor but is a degree of freedom for the genetic algorithm.

Below we will have a closer look at the steps of the algorithm.

3.2.1 Initial population

From biology, we know that any organism can be represented by its phenotype, which actually determines how the object looks like in the real world, and the genotype, that contains all information about the object at the level of chromosome set. Moreover, each gene reflects somehow in the phenotype. Thus, we need to present each feature of the object in a form suitable for the genetic algorithm.

All further functions perform at the level of the genotype, being agnostic to the information about the internal structure of the object, which determines the wide application of the genetic algorithm in a variety of problems.

The genetic algorithm uses bit strings for representing the object's genotype. In this case, each attribute of the object in the phenotype corresponds to one gene in the genotype of the object. A gene is a bit string, often of a fixed length

For coding genotype from phenotype, it is possible to use the bit values of characteristics as simplest method. Then it will be quite easy to create a gene of a certain length, sufficient to represent all possible values of features. Unfortunately, this method of coding is not perfect. The main drawback is that adjacent chromosomes by phenotype can differ in many bits of the gene, which greatly increases the size of the search space. One possible solution is to use the Gray code.

In order to determine the phenotype of the object, we only need to know what gene values correspond to what features of the object in phenotype. This operation is called decoding.

Thus, in order to create the initial population, it is first necessary to determine the methods of coding the individuals. These methods are later used to decode optimized chromosome back to the individual.

3.2.2 Crossover

The crossover operator is an operator which exchanges the chromosomes parts. It simulates the process of individuals interbreeding.

Single-point

Consider two parent chromosomes A and B and randomly choose point inside the chromosome, which is dividing both of them into two parts. We call this point a crossover point or breakpoint. The described process is shown in Figure 3.2.

This type of crossing-over is called single-point crossover because the parent chromosomes are cut only at one random point.

Two-point

In two-point and multi-point crossovers, chromosomes are considered as cycles, which are formed by connecting the ends of linear chromosomes. To replace a segment of one cycle with a segment of another cycle, you need to select two break points. In this representation, the single-point crossover can be considered as the two-point crossover, but with one cut point fixed at the beginning of the chromosome. Therefore, a two-point crossover solves the same task as it is shown in the Figure 3.3.

Uniform

Uniform crossover is taking two parent chromosomes A and B and then creating two new chromosomes from them gene by gene. The first descendant is taking the gene from chromosome A with probability p and from chromosome B with probability $1 - p$. The second descendant is taking the gene from the opposite chromosome than the first one. Uniform crossover is shown in the Figure 3.4

Particular case when the mixing ratio equals 0.5 is called half-uniform crossover.

Three parent

In this method, we are combining three initial chromosomes A, B, and C into one descendant. The rule is the following. Genes from the chromosome A and B at the same position are compared. If they are the same this gene goes to the descendant, otherwise, the gene from the chromosome C is taken.

The new chromosome is substituting chromosome A in the new population.

3.2.3 Mutation

The probability of a mutation is much smaller than the probability of crossing-over and rarely exceeds 1%. Similarly, the probability can be a function of the characteristics of the problem being solved. For instance, the probability of gene mutation can be put inversely proportional to the length of the chromosome or the size of the population.

Depending on the type of the function being optimized, the strategy of selecting the mutation probability varies. For example, a mutation with a fixed probability leads to good results for unimodal functions. For multimodal, a self-adapting estimation is used.

Uniform

As well as crossing-over, mutations can be carried out not only at one random point. You can choose to change several genes in one chromosome, and their number can be again random. It is also possible to invert a group of contiguous points at once.

Insertion

The insertion operator is a change of the genes order in the chromosome or its fragment. This operator is used quite rarely, but its main purpose is to try to find the order of genes, which has better evolutionary potential. Inversion also greatly expands the scope of the search. The GA does not only try to find good sets of gene values, it also simultaneously tries to find a good ordering of the genes.

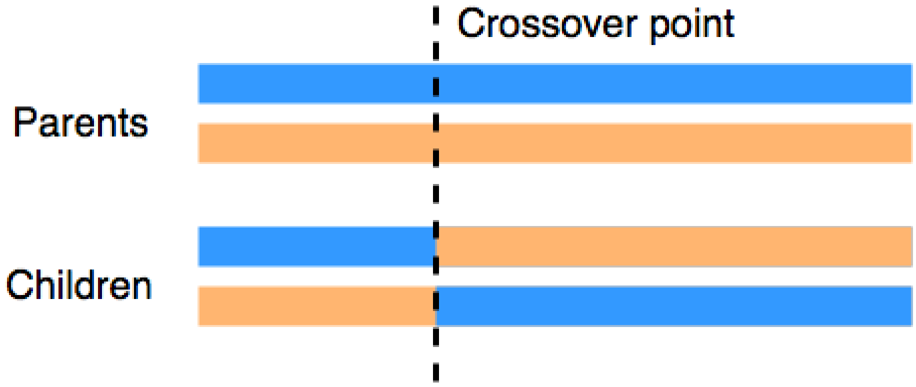


Figure 3.2: Single-point crossover

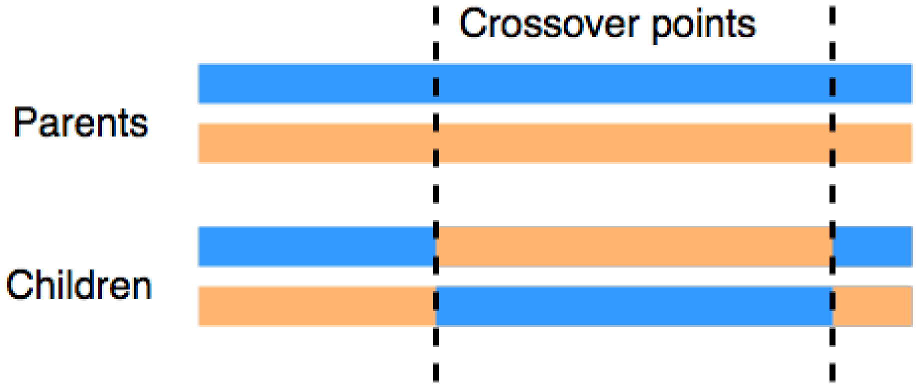


Figure 3.3: Two-point crossover

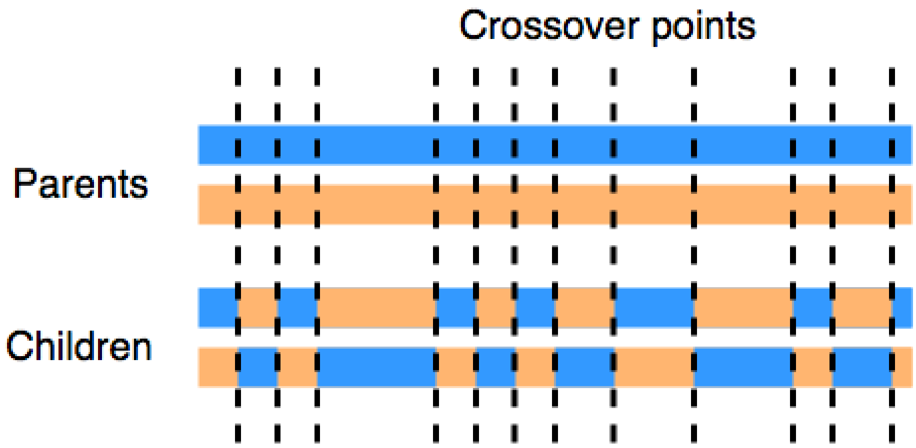


Figure 3.4: Uniform crossover

Flip Bit

This operator of the mutation is taking the particular chromosome and negates the bits with some probability p, independently on other bits in the solution. It is suggested to have this probability as $p = \frac{1}{n}$ where n is the number of bits in the chromosome.

3.2.4 Selection methods

After the new population of individuals is generated, the fitness function is calculated and terminating condition is checked. In case the stopping condition is not fulfilled, for the further development of the search process, specialized operators of the genetic algorithm are applied. One of them is the selection operator.

The selection operator is one of the most important operators helping to maintain the genetic diversity of the population. Diversity is responsible for the behavior of the population whether it is going to converge on some local maxima or be optimized further.

The selection operator is the one which selects the individuals with some given feature (for instance good value of the objective function) to produce an offspring. So only those individuals whose fitness is greater or equal to some threshold value can become parents. The threshold can be for example the average fitness of the population.

Roulette wheel

Roulette method is one of the selection methods which is used in the classic genetic algorithm in the following manner. Each chromosome is associated with a segment of the roulette wheel whose value is set proportional to the value of the fitness function of the given chromosome, as it is shown on the Figure 3.5. Therefore, the greater the value of fitness function, the larger the sector on the roulette wheel. Hence it follows that the larger the sector on the roulette wheel, the higher the chance that this particular chromosome will be chosen.

The weak side of this method is that individuals with very small fitness functions are too quickly excluded from the population, which can lead to convergence of the genetic algorithm. Therefore, alternative selection algorithms have been created and used.

Tournament

In the Tournament selection, a group of $t \geq 2$ individuals is selected randomly from a population consisting of N individuals. The individual with the highest fitness in the group is selected, the rest are discarded. This operation is repeated k times. Then, the selected individuals are used for crossing-over. The size of group t is often equal to 2, in such cases we speak of paired (binary) tournaments. The number t is called the tournament size.

The advantage of the tournament selection is that it does not require additional computations or ordering of the individuals in the population. The process is shown on the Figure 3.6.

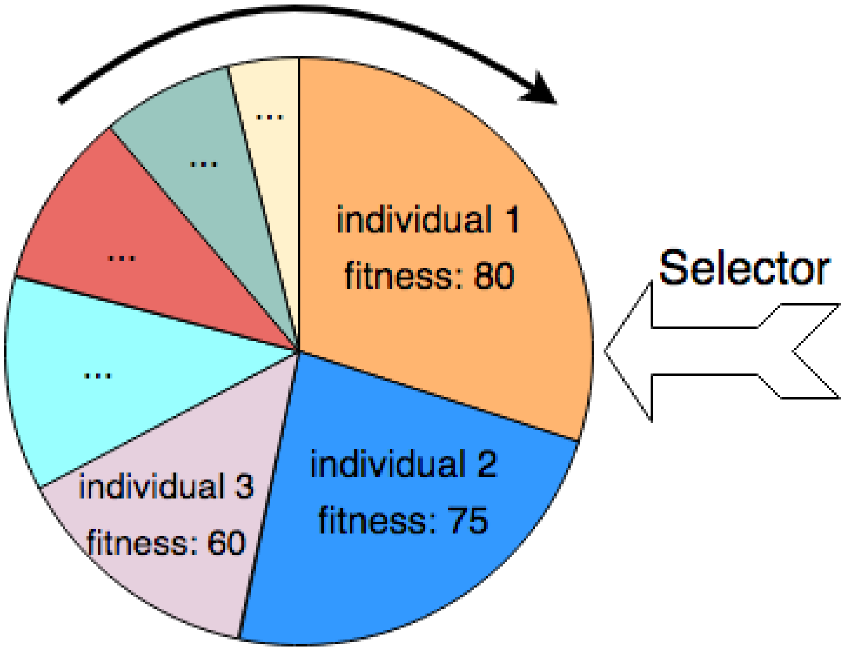


Figure 3.5: Roulette selection method

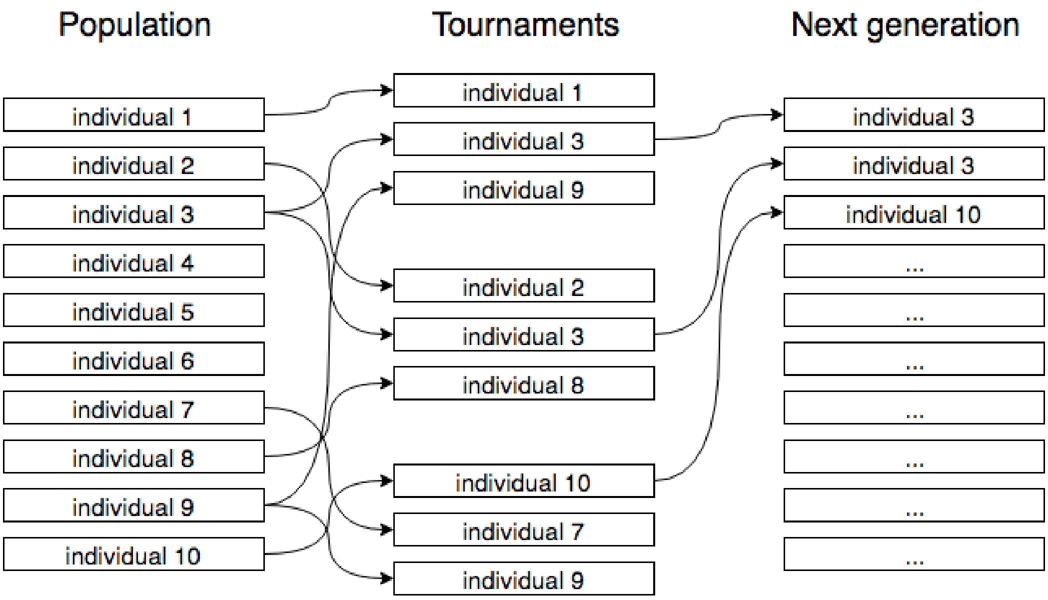


Figure 3.6: Tournament selection method

Truncation

In the Truncation selection individuals are ranked based on their rank descending, so the fittest individual is the first to stand. The number of individuals for crossing-over is selected in accordance with the threshold $T \in (0;1]$. The threshold determines which proportion of individuals starting with the very first will participate in the selection. All the individuals that fall under the threshold are reproduced $\frac{1}{T}$ times.

Due to the fact that the sorted population is used in this strategy, the execution time can be large for big populations and depends on the sorting algorithm.

3.2.5 Generation of the next population

This step is, in its own way, one of the types of breeding. Here individuals are selected from two populations (parents and descendants) into a new population that will work on the next step of the algorithm.

In order to have different optimization strategies, it is necessary to provide diversity in the formation schemes of a new generation. Here are the main schemes for the formation of a new population.

Elitism strategy is the method based on building a new population only from the best individuals of the reproductive group that unites parents and their descendants. This method is good from the point of view that it excludes random walk through the search space, as the best specimens (found at this stage of the search or earlier) transit to the next generation. Unfortunately, it leads to converging on the local maximum instead of the global one in most cases.

Exclusion selection is based on the bi-criteria principle. An individual from the reproduction group is included in the population of a new generation based on the fitness of this individual and whether in the next generation there is no individual with a similar chromosomal set. Between all the individuals with the same genotypes, the one with the higher fitness is preferred. Thus, two goals are achieved: firstly, the best solutions possessing different chromosome sets are not lost, and secondly, the population is consistently maintained with sufficient genetic diversity.

Only descendants selection - the method is based on building a new population only from the descendant population.

Random selection - when the individuals, that create a new population are randomly selected from a reproduction group that unites parents and their descendants.

3.2.6 Stop condition

The definition of a stop condition for the GA depends on its specific application. In optimization problems, where the maximum (or minimum) value of the fitness function is known, the algorithm can stop after reaching the expected optimal value, possibly with the specified accuracy.

The algorithm termination can also occur when its execution does not lead to any improvement of the fitness value already achieved.

The algorithm can be terminated after a certain execution time or after a specified number of iterations have been performed.

3.2.7 Best specimen

If the termination condition of the algorithm is satisfied, then the required solution of the problem should be present. The best solution is considered to be the individual with the highest fitness function value.

After the stopping condition is fulfilled, the final step is to select the fittest chromosome from the population and decode it back to the phenotype.

4. Project design

This chapter describes the technical details of the Genetic algorithm implementation for the WSN optimal placement. We describe the mathematical model of the stated problem in Section 4.1, details of introduction of particular problem to the GA in Section 4.2, technical implementation in Section 4.3 and interface of the program in Section 4.4.

4.1 Model of the optimal network topology design problem

Consider the problem of mesh network nodes placement as the coverage problem.

Let us denote Ω as a region of a building from $\mathbb{R}^2$, which should be covered with devices

X_k - a set of coordinates of sensors' centers

X_p - a set of coordinates of hubs' centers

N_1, N_2 - number of hubs and sensors

τ_p - a set of coordinates of centers of hubs, visible from $\overline{x_p}$

$\tau_{pd}, d = 1, 2$ - a set of coordinates of centers of hubs, visible from $\overline{x_k}$

$rf(\overline{x_1}, \overline{x_2})$ - function which returns maximum distance of stable radio connection between devices $\overline{x_1}$ and $\overline{x_2}$ based on radio propagation model.

Call the circle of radius r with a center in point $\overline{x}$ as set from $\mathbb{R}^2$, s.t. $B(\overline{x}, r) = \{x \in \mathbb{R}^2 | c(x, \overline{x}) \leq r\}$, where $c(x, \overline{x})$ is the Eucledian distance between points x and $\overline{x}$.

A set of centers $\{X_k, X_p\}$ forms coverage of region Ω with circles of radius $r_1^{(p)}$ and $r_2^{(k)}$, where $r_1^{(p)}$ - radius of p-th hub coverage, $r_2^{(k)}$ - radius of k-th sensor coverage, $k = 1, ..., N_2$, $p = 1, ..., N_1$.

It is necessary to place centers of circles X_k, X_p, such that measure on a set of all covered points from Ω by these circles is maximized

$$\mu(\Omega \cap [(\bigcup_{p=1}^{N_1} B(x_p, r_1^{(p)})) \bigcup (\bigcup_{k=1}^{N_2} B(x_k, r_2^{(k)}))]) \Rightarrow \max$$

Let us distinguish following subsets in set Ω which are created after coverage of Ω with circles.

- $\Omega^{(1)} = \Omega \setminus [(\bigcup_{p=1}^{N_1} B(x_p, r_1^{(p)})) \bigcup (\bigcup_{k=1}^{N_2} B(x_k, r_2^{(k)}))]$ - subset of points in Ω, which are not covered with circles.
- $\Omega^{(2)} = \Omega \cap (\bigcup_{i=1}^{N_1+N_2} \bigcup_{j=1, i\neq j}^{N_1+N_2} (B(x_i, r_i) \cap B(x_j, r_j)))$ - set of points which belong to set Ω and intersection of two or more circles.
- $\Omega^{(3)} = [(\bigcup_{p=1}^{N_1} B(x_p, r_1^{(p)})) \bigcup (\bigcup_{k=1}^{N_2} B(x_k, r_2^{(k)}))] \setminus \Omega$ - set of points which belong to at least one coverage circle but do not belong to region Ω.

To have optimal coverage of region Ω with sensors it is necessary to minimize the number of points, which belong to

$$\Omega^* = \bigcup_{i=1}^{3} \Omega^{(i)}: \ \mu(\Omega^*) \Rightarrow \min$$

For robustness of network let's formulate following additional conditions:

- Each sensor should see at least two hubs. Using following notation $\overline{x_k}$ - coordinate of sensor, τ_{pd} - coordinate of hub, visible from $\overline{x_k}$, $d = 1, 2$, this condition can be represented as

$$\sum_{k=1}^{N_2} \sum_{d=1}^{2} c(\overline{x_k}, \tau_{pd}) - rf(\overline{x_k}, \tau_{pd})) \geq 0 \tag{4.1}$$

- Each hub should see at least one another hub. For the $\overline{x_p}$ - coordinate of hub, τ_p - coordinate of hub, visible from hub $\overline{x_p}$, it is described as

$$\sum_{p=1}^{N_1} c(\overline{x_p}, \tau_p) - rf(\overline{x_p}, \tau_p) \geq 0 \tag{4.2}$$

As a target function, we will consider following

$$J(\overline{x}) = \sum_{i=1}^{3} \sum_{p=1}^{N_1} \sum_{k=1}^{N_2} [\int_{\Omega_p^{(i)}} f(x, \overline{x_p}, r_1^p) dx + \int_{\Omega_k^{(i)}} f(x, \overline{x_k}, r_2^k) dx], \text{ where}$$

$\Omega_p^{(i)} = \Omega^{(i)} \cap \Omega_p, i = 1, 2;$
$\Omega_p^{(3)} = \{x \in \bigcup_{p=1}^{N_1} B(\overline{x_p}, r_1^p), \exists p | c(x, \overline{x_p}) - r_1^p \leq 0\};$
$\Omega_k^{(i)} = \Omega^{(i)} \cap \Omega_k, i = 1, 2;$
$\Omega_k^{(3)} = \{x \in \bigcup_{k=1}^{N_2} B(\overline{x_k}, r_2^k), \exists p | c(x, \overline{x_k}) - r_2^k \leq 0\}.$
Then, $\Omega^{(3)} = \bigcup_{p=1}^{N_1} \bigcup_{k=1}^{N_2} \Omega_p^{(3)} \Omega_k^{(3)}$

Function $f(x, \overline{x}, r)$ should be chosen in such manner that it has values in boundary points of subsets $\Omega_p^{(i)}$, which are at the same time boundary points of circles $B(\overline{x}, r)$ should be equal to zero. $f(x, \overline{x}, r)|_{x \in \Omega_B^*} = 0$

If take $f(x, \overline{x}, r) = |c(x, \overline{x}) - r|$, then in boundary points of circles $B(\overline{x}, r)$ value of $c(x, \overline{x}) - r$ would be equal to zero or different from zero in point x depending on whether x is covered or not covered with circle.

So the target function will be

$$J(\overline{x}) = \sum_{i=1}^{3} \sum_{p=1}^{N_1} \sum_{k=1}^{N_2} [\int_{\Omega_p^{(i)}} |c(x, \overline{x_p}) - r_1^p| dx + \int_{\Omega_k^{(i)}} |c(x, \overline{x_k}) - r_2^k| dx] \tag{4.3}$$

Additional conditions for robustness of the network (4.1) (4.2) can be added to function (4.3) using Penalty method in the following way

$$\begin{aligned} J(\overline{x_p}, \overline{x_k}, \alpha) = J(\overline{x_p}, \overline{x_k}) + \alpha(\sum_{p=1}^{N_1} \frac{1}{c(\overline{x_p}, \tau_p) - rf(\overline{x_p}, \tau_p)} + \\ + \sum_{k=1}^{N_2} \sum_{d=1}^{2} \frac{1}{c(\overline{x_k}, \tau_{pd}) - rf(\overline{x_k}, \tau_{pd})}) \end{aligned} \tag{4.4}$$

Thus, the problem of finding the minimum of a functional (4.3) with constraints (4.1), (4.2) is reduced to the problem of unconditional optimization of the functional (4.4). The value of α should be chosen rather small, so that in the minimum point vicinity the impact of the constraints is not noticable. Then the minimum point of the function without restrictions $J(\overline{x})$ will coincide with the one for the function $J(\overline{x_p}, \overline{x_k}, \alpha)$ restricted by constraints (4.1), (4.2).

4.2 Application of GA

In this section we describe methods which adapt our particular problem for the Genetic algorithm. Those methods are used for grid generation (Section 4.2.1), chromosome encoding (Section 4.2.2) and population initialization (Section 4.2.3).

4.2.1 Floorplan sampling

Consider a two-dimensional floorplan as a region Ω where the WSN deployment is required. This region should be sampled in order to solve the problem of device placement. The easiest way is to cover the region with the uniform grid where each node can be either empty or filled with one device of any type. The grid is pretty dense to cover all points of interest and give wide possibilities of device placement. This type of grid is shown in Figure 4.1

However, the fact that this method is later used to place sensors in real environment imposes some restrictions. Usually, it is not allowed to place temperature/humidity sensors on the ceiling since measured values there can be different from real up to 10%. The best place for the environmental sensor is on the wall at the human's head height. According to this, the second method of the region sampling was introduced. It is placing the nodes only on the walls with the defined step which means that later optimal solution will consist of the sensors which can be stuck to the wall. The result of this placement is shown in Figure 4.2

4.2.2 Encoding of individual

After the grid is created we receive nodes X_n, $n = 1, ..., N$ and each of them represents the exact position on the map. This grid is stable therefore allowing to decode the phenotype of the chromosome. Then the binarily coded chromosome consisting of N genes is created. Each gene G_n is represented by two bits and encodes the status of the node X_n as follows in Table 4.1

Node status	Bit 1	Bit 2
Empty node	0	0
Node contains sensor	0	1
Node contains sensor	1	0
Node contains hub	1	1

Table 4.1: Gene description

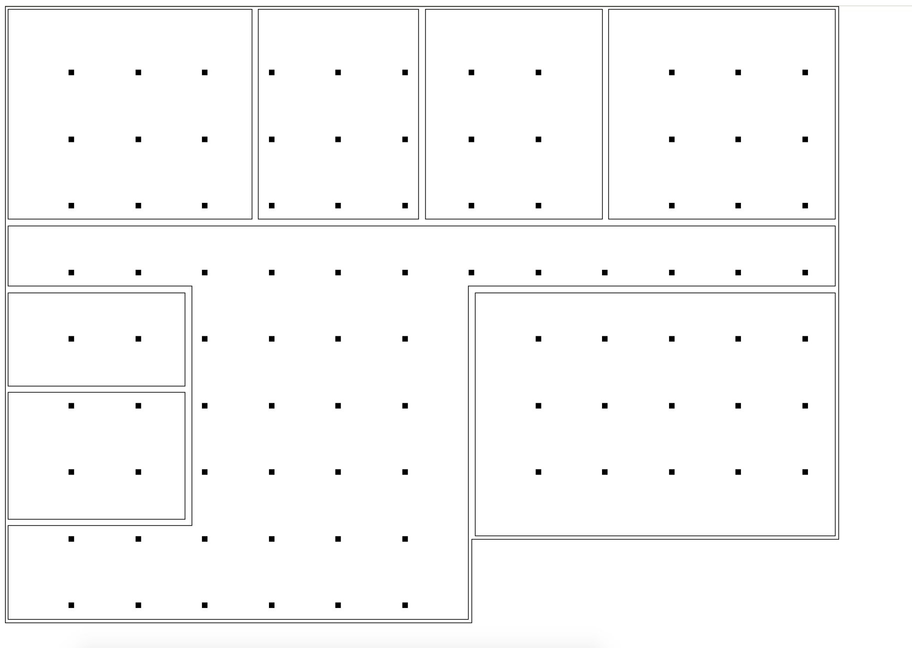


Figure 4.1: Uniform grid

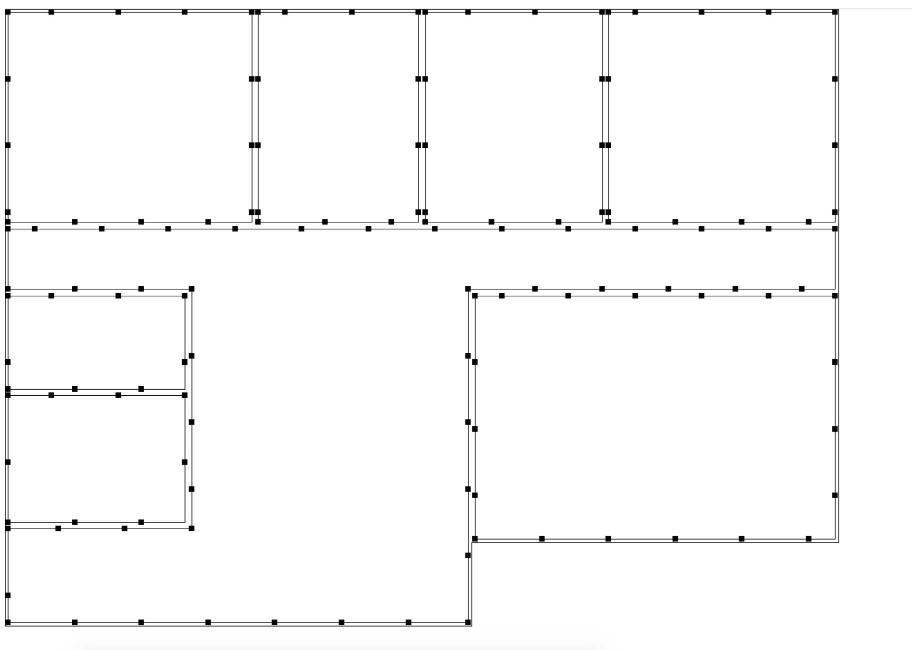


Figure 4.2: Wall grid

4.2.3 Method of initial population generation

At the initial stage of the genetic algorithm, a certain population of chromosomes is initialized and randomly filled with values. The number of chromosomes in population is not fixed and is usually chosen quite significant in order to diversify the population.

Even though chromosomes are filled with random values, there is a possibility to improve the initial population by picking correct probabilities of bits to be either 0 or 1. Creating "good" initial population consisting of a set of local optima can lead to a noticeable reduction of steps for the global optimum achievement, which is not guaranteed when the initial population is formed randomly [36].

That will allow us to speed up the program execution and find an optimal solution in the lower amount of steps. In our case, the user is able to choose the probability of having 1 in the first and second bit of every gene from the interface. But the program is suggesting the most appropriate probabilities based on the floor area, grid density, hub and sensor coverage.

From the grid density, we know the total number of nodes N on the particular floor plan. Total area and area of single device coverage give us target number of hubs and sensors. From this, we can calculate target probability to have the device in one node as a ratio of the target number of devices to the total number of nodes.

4.3 Implementation

In this section we discuss technical details of the implementation.

4.3.1 Floorplan representation

The user of this tool has an ability to run the optimization of the WSN placement on top of his own floorplan. To upload it to the program the following format is used.

A file consists of two lines. On the first line we have the JSON object (Listing 4.1) which represents the outline of the floorplan. It is the array of walls where each wall is the line between two points with coordinates (x_1, y_1) and (x_2, y_2). The second line describes the internal structure of the floor which is the array of rooms, where each room consists of walls as described before with Wall attenuation factor parameter (Listing 4.2).

Outline and rooms should be simple polygons.

```
[
    {
      "x1": 0,
      "y1": 0,
      "x2": 1,
      "y2": 1
    },
    ...
]
```

Listing 4.1: Outline JSON format

```
[
    {"room":
        [
            {"x1":0,"y1":0,"x2":2,"y2":0,"waf":3},
            {"x1":2,"y1":0,"x2":2,"y2":2,"waf":3},
            {"x1":2,"y1":2,"x2":0,"y2":2,"waf":3},
            {"x1":0,"y1":2,"x2":0,"y2":0,"waf":3}
        ]
    },
    ...
]
```

Listing 4.2: Floorplan JSON format

4.3.2 Radio connection matrix

After the floorplan is imported and sampled we are creating the radio connection matrix. That is the square Boolean matrix of order N where N is a number of the grid nodes. At row i, column j we store the status of the radio communication between devices i and j. It is 0 if they can not exchange the message between each other and 1 otherwise.

This matrix computation is quite time-consuming especially if we have the dense grid. But later we don't need to recompute it again since the grid is not changing. And because radio connectivity is the important part of the fitness evaluation for each individual in the population it is more efficient to precompute it once rather than compute it for every new generation.

Radio models for this computation were described in Section 2.1.3. For the Friis model computations are pretty straightforward, we just need to know the distance between two points. But for the Keenan-Motley model it is also required to count the number of walls between two devices. For that we are finding the intersection of two segments geometrically. Those two segments are namely the wall and line-of-sight between devices. We are iterating walls collection and in such manner it is possible to calculate total WAF.

4.3.3 Compilation

The program is written in C# 7.0 using Microsoft Visual Studio 2017.

In this project we use Json.NET [37] for deserializing JSON objects and ClipperLib [38] for deflating polygons in order to display floorplan in a nice way. Both these libraries have to be linked as DLL files in order to compile.

4.4 User interface

In this section we describe user interface of the created program. The main screen (Figure 4.3) consists of 3 zones and 3 buttons. Those zones are the output of GA parameters and results, tabs for user input, and graphical representation of the result. Buttons are for the generating floorplan, first population, and running the genetic algorithm with this input until the stopping criteria is satisfied.

As the output user see total floorplan area, performed number of genetic steps, target and currently achieved number of hubs/sensors, best fitness in the current generation and among all generations. Fitness is split into 3 fields, those are for the high, medium and low priority constraints.

There are 9 tabs for the user input, mainly they are dedicated to change method or parameters of some genetic operator. On the Floorplan tab (Figure 4.3) user can choose the floor plan and it's scale, type and step for the grid.

On the Population tab (Figure 4.4 (a)) user can change the probability of having hub/sensor in the initial population using sliders. This probability is calculated real-time and displayed on the label below. It is also possible to change the number of chromosomes in first population.

Mutation tab (Figure 4.4 (b)) allows to choose the mutation method from the list of Uniform, Insertion and Flip Bit, change number of mutations in one chromosome and make it either fixed or random. And for the Uniform mutation, it is possible to change the probability of having hub/sensor the same way as on the Population tab.

Crossover tab (Figure 4.5 (a)) gives an ability to choose the method from Single-point, Two-point, Three parent and Uniform list. Also, user can choose the number of crossovers in population, either fixed or random in range.

On the Fitness function tab (Figure 4.5 (b)) user is able to choose parameters for the hub and sensor coverage as well as penalization parameter for high, medium and low priority constraints.

Selection tab (Figure 4.6 (a)) gives an ability to choose selection method from Roulette wheel, Tournament and Truncation list. For the Roulette wheel, there is a checkbox to put several bad chromosomes to the new generation.

On the Stopping criteria tab (Figure 4.6 (b)) user can choose criteria from the Number of steps, Time, Stagnation list and it's parameter.

Radio tab (Figure 4.7 (a)) is dedicated for changing radio devices parameters such as transmit power, receive sensitivity, signal to noise ratio and type of radio model (Friis or Keenan-Motley model).

On the Settings tab (Figure 4.7 (b)) user can change visual representation of results such as color and size of hubs/sensors icon and connectivity between devices.

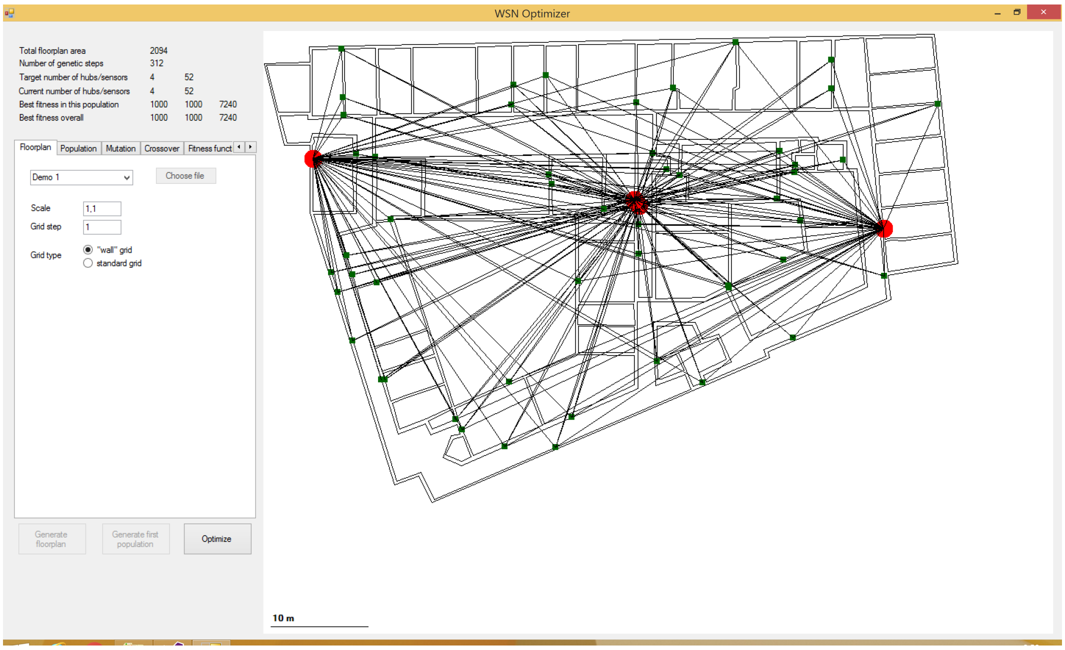
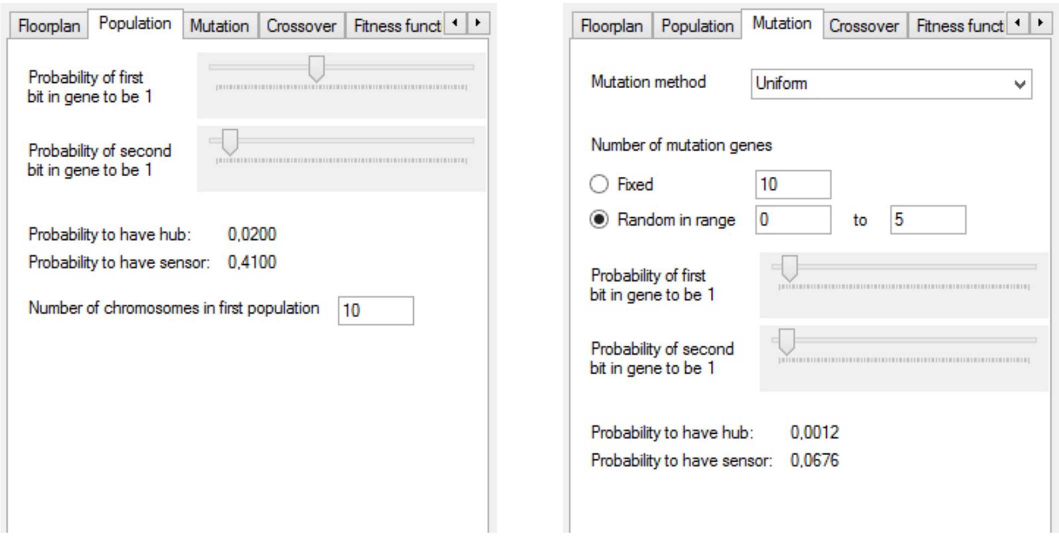


Figure 4.3: Main screen



a) b)

Figure 4.4: Population (a) and mutation (b) settings

Figure 4.5: Crossover (a) and fitness function (b) settings

Figure 4.6: Selection (a) and stoping criteria (b) settings

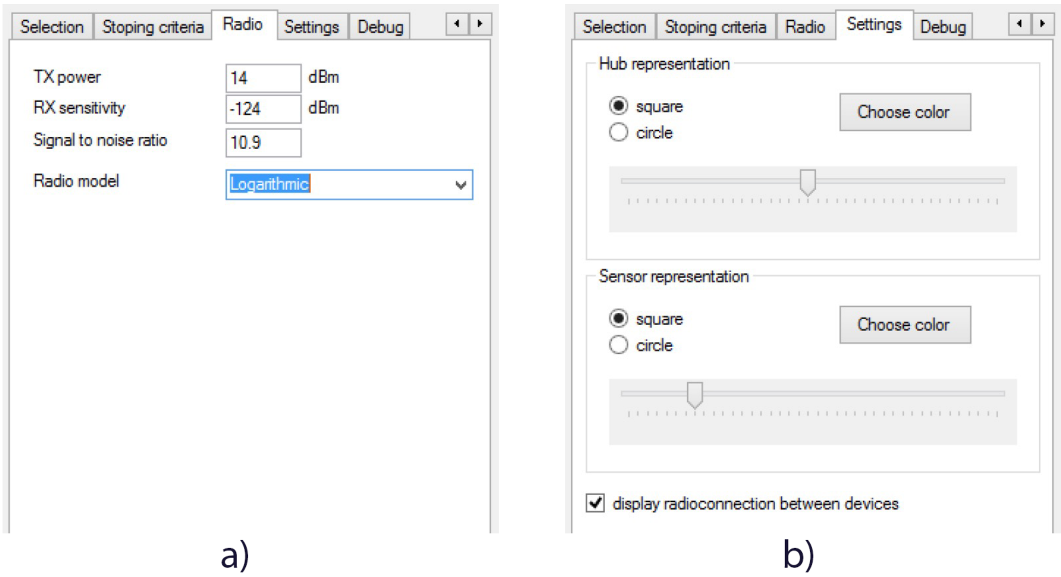


Figure 4.7: Radio (a) and display (b) settings

5. Experiments

This chapter is dedicated to comparing results between different algorithms for indoor WSN placement. Primarily we would compare various genetic method approaches between each other to achieve the best result in given problem (Section 5.1). Also received the result from genetic algorithm will be compared with other papers in Section 5.2

5.1 Genetic methods

As the basic configuration, we are taking floorplan from the file demo1.json with scale 1.1 and grid step 1. Hub and sensor cover 450 m^2, and 40 m^2 respectively. Target numbers of hubs and sensors are computed as the ceiling of the ratio of the total area to the coverage area. Coefficients for the hub optimal number, sensor optimal number, and robustness of the network are 1, 0.75 and 0.0001 respectively. GA is performing 1000 steps and outputting fitness values of the best chromosome in each generation.

For the method comparison, we introduce mistake on a number of hubs and sensors in the population. That is the ratio of the difference between current and target devices number to target for the best individual in the population. Zero mistake means that we have exactly target number of devices, negative means lower amount than needed and positive means more than needed.

We also introduce degeneracy as the difference between best and worst chromosome in the population or as a ratio of different genes at the same position to the total number of genes.

5.1.1 Population size

For the initial population, we place hub with 1.44% probability and sensor with 21.12% while the target numbers are 0.38% and 4.9% respectively. Numbers for the initial population are taken empirically, they should be a bit higher than the target. We will test populations of size 50, 100, 200 and 500. Bigger population of e.g. 1000 or 2500 chromosomes becomes hard to compute in the reasonable time.

As seen in Figure 5.1 for sensors mistake, population of 50 individuals is not enough. Populations of 100 and 200 are better but still not reaching optimum, while the population of 500 chromosomes reaches optimum on both the number of hubs and sensors in approximately 300 generations.

5.1.2 Crossover method

Then we performed a comparison of 4 crossover methods, namely Single-point, Two-point, Three parent, and Uniform. A number of crossovers in each population is equal to 10%, this number is also received empirically. For the uniform crossover, the coefficient is equal to 0.5 so it is a half-uniform method.

Only Uniform crossover has reached optimum value on number of sensors (Figure 5.3) under 1000 generations. When using other methods our population degenerated earlier than reaching the optimal result.

5.1.3 Selection method

Roulette wheel as a selection method is not useful in our particular case (Figure 5.4). It is degenerating much quicker than other two methods and not reaching the optimal result. Roulette wheel method with bad chromosomes gives a bit better result but not much.

Comparing Truncation and Tournament methods it turns out that Truncation is faster to find the optimal solution in a number of sensors but it also degenerates (Figure 5.5) quicker than Tournament so the second one gives better result on the robustness of the network.

5.1.4 Mutation method

The probability of chromosome mutation in population is 1%. And when mutation operator is executed on chromosome it can mutate from 0 to 3 genes in a chromosome.

Comparing three methods we can immediately notice that Flip Bit is not useful in our case. That is because of the design of this method, which switches the bits. In the initial population, we have around 90% of empty points which are coded as 00. Flip Bit will change them to the hub (11) with high probability. That is why the number of hubs grows dramatically (Figure 5.6). Also, this operator not changing the number of sensors which are coded as 01 or 10 (Figure 5.7).

If talking about Uniform and Insertion mutations, on the first look Insertion mutation looks more promising. They are working with almost same speed to the optimal point on the number of hubs and sensors. But because insertion operator just changes genes order without introducing or deleting random devices, it then sticks to the optimal number and only improves robustness (Figure 5.8).

However, when taking a look on the decoded result it is noticeable that result from Insertion method (Figure 5.10) is worse than the one from the Uniform method (Figure 5.9) simply because Insertion is grouping devices together to achieve better robustness. And that is making the overall floor coverage worse.

5.1.5 Proposed set of methods

As a result of this comparison we propose the following settings for the genetic algorithm to achieve good results in the problem of WSN placement indoors.

- Initial population: 500 individuals, initial hubs/sensors probability of placement higher than target
- Mutation: uniform with 1% probability and 0-3 genes to mutate. Hubs/sensors probability of placement lower than target
- Crossover: uniform with 10% probability
- Selection: tournament

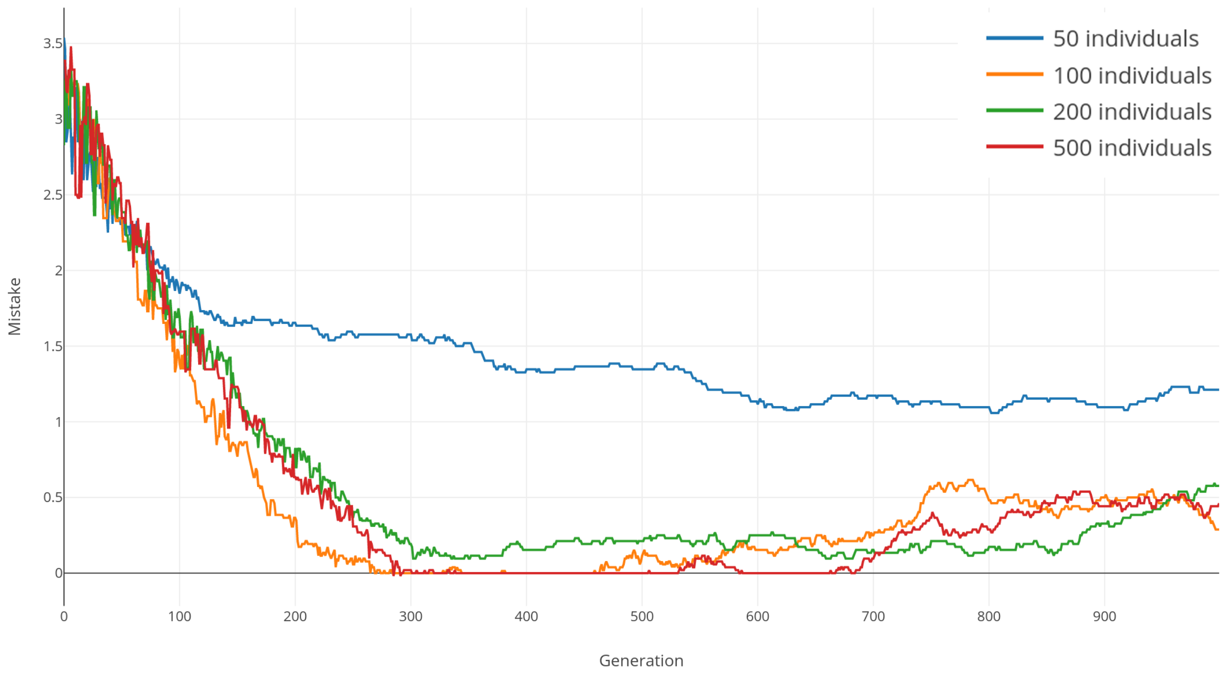


Figure 5.1: Sensors mistake depending on population size

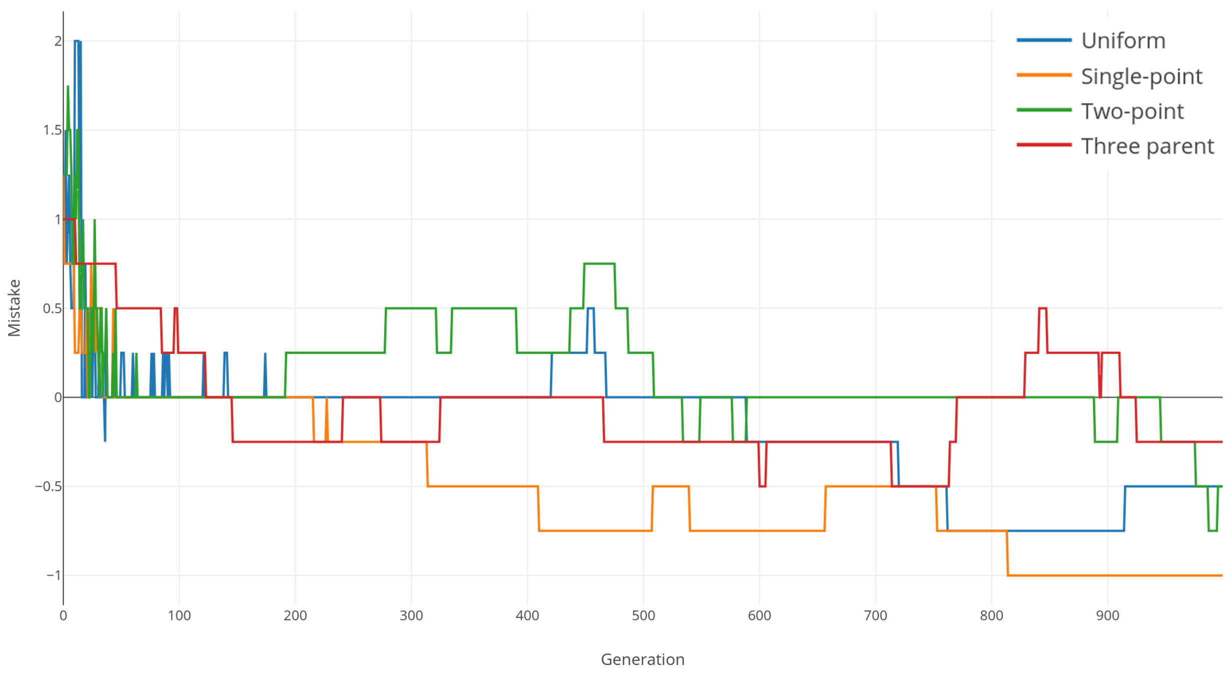


Figure 5.2: Hubs mistake depending on crossover method

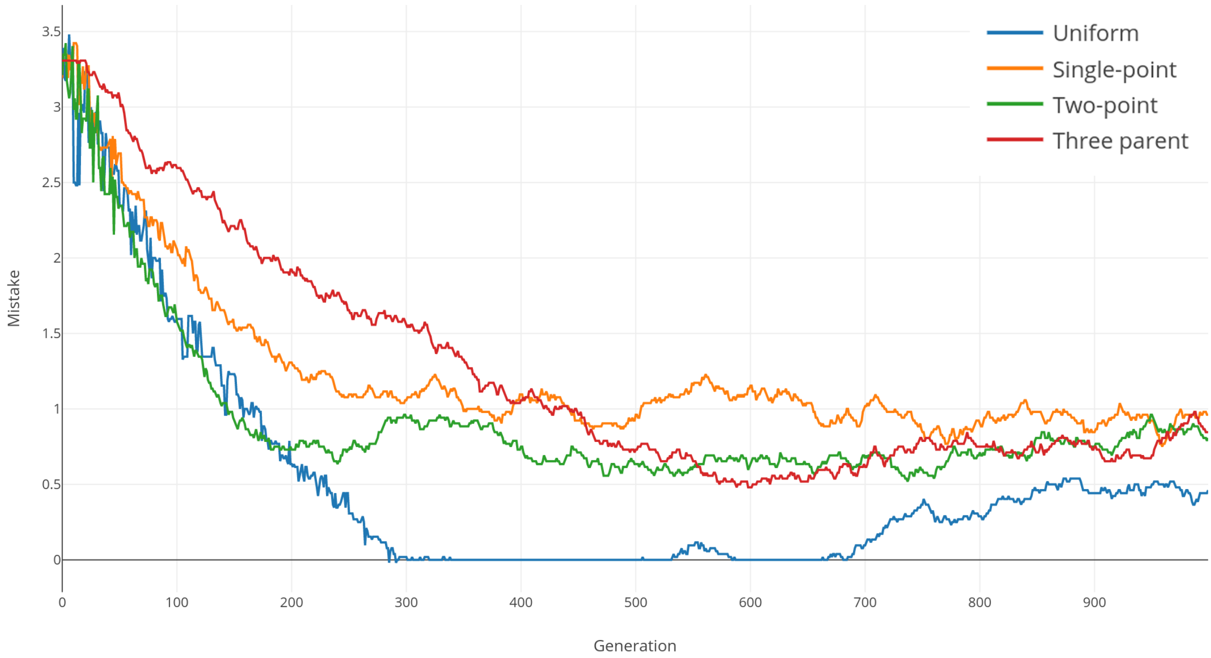


Figure 5.3: Sensors mistake depending on crossover method

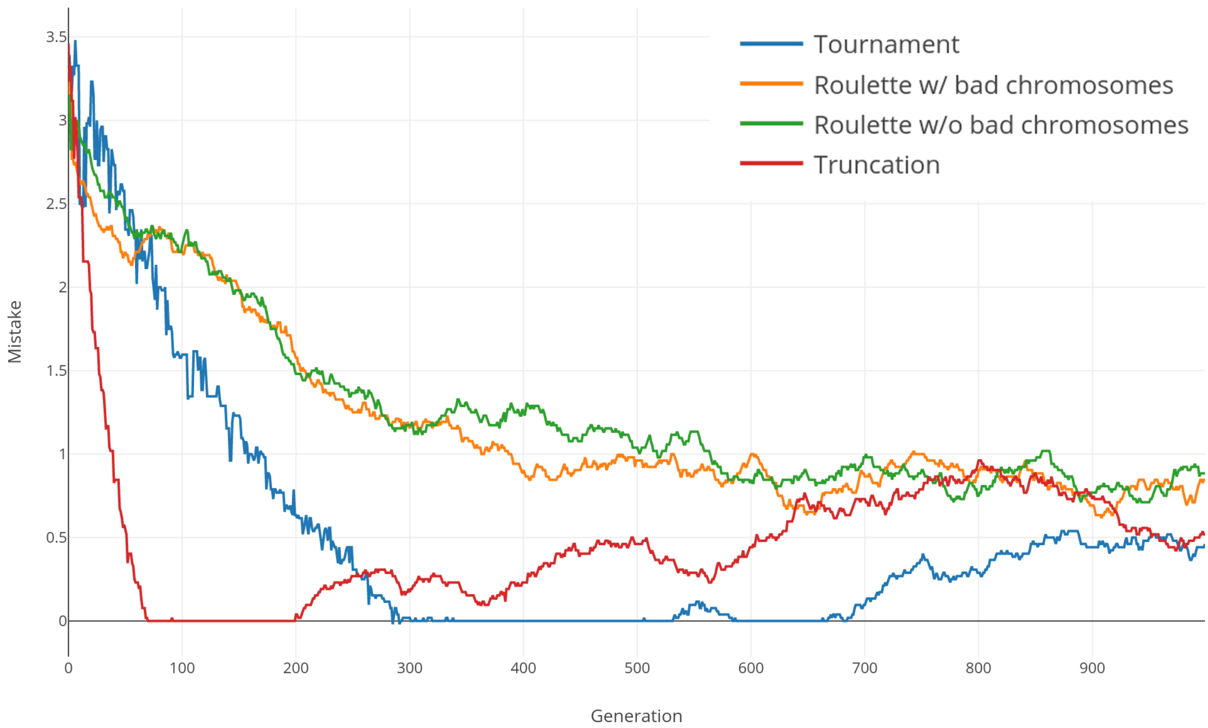


Figure 5.4: Sensors mistake depending on selection method

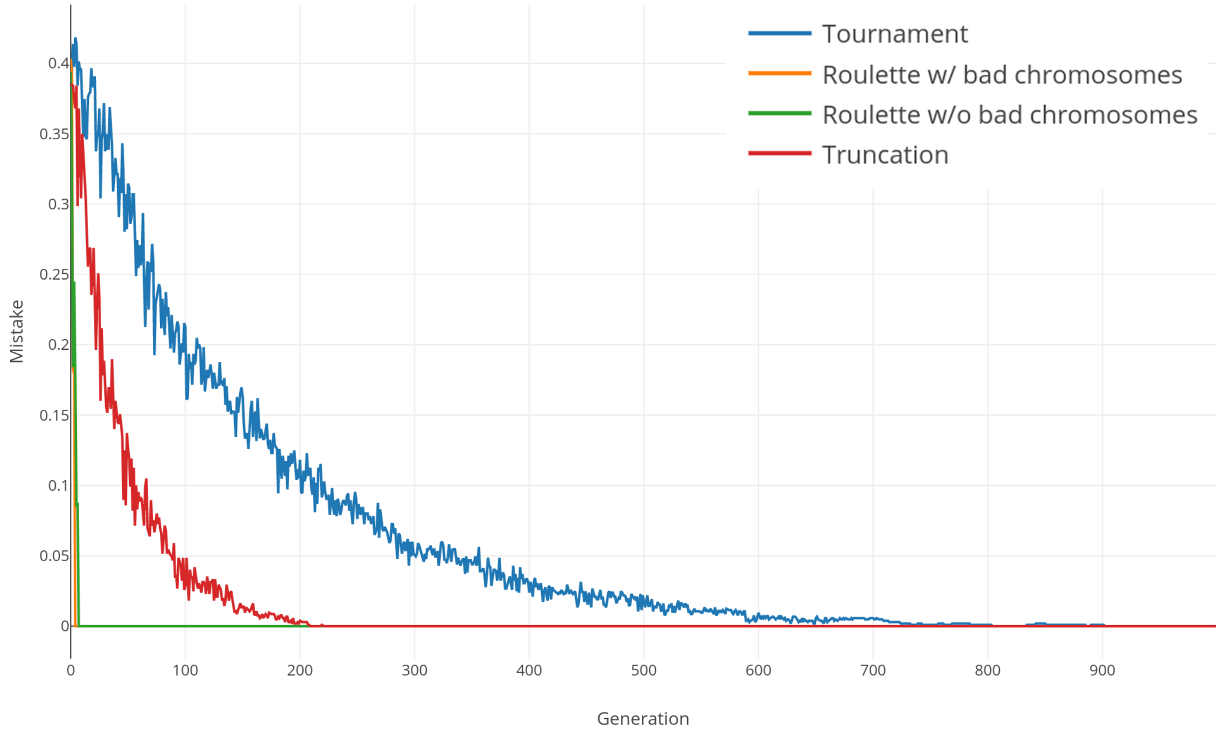


Figure 5.5: Degeneracy depending on selection method

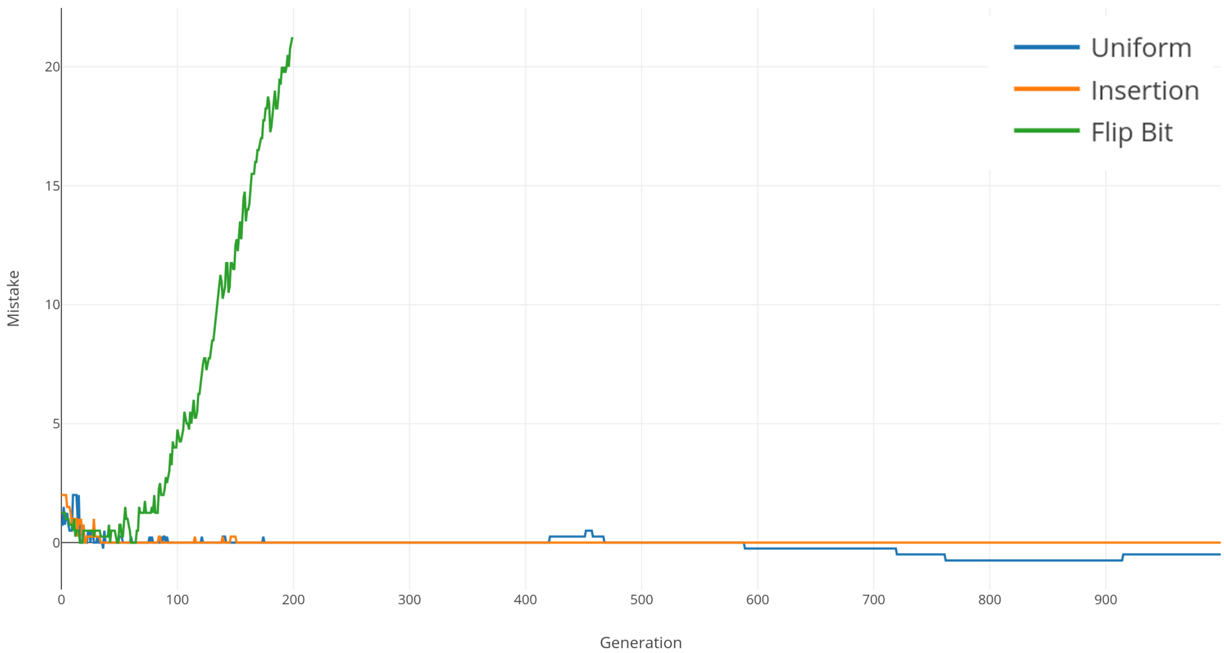


Figure 5.6: Hubs mistake depending on mutation method

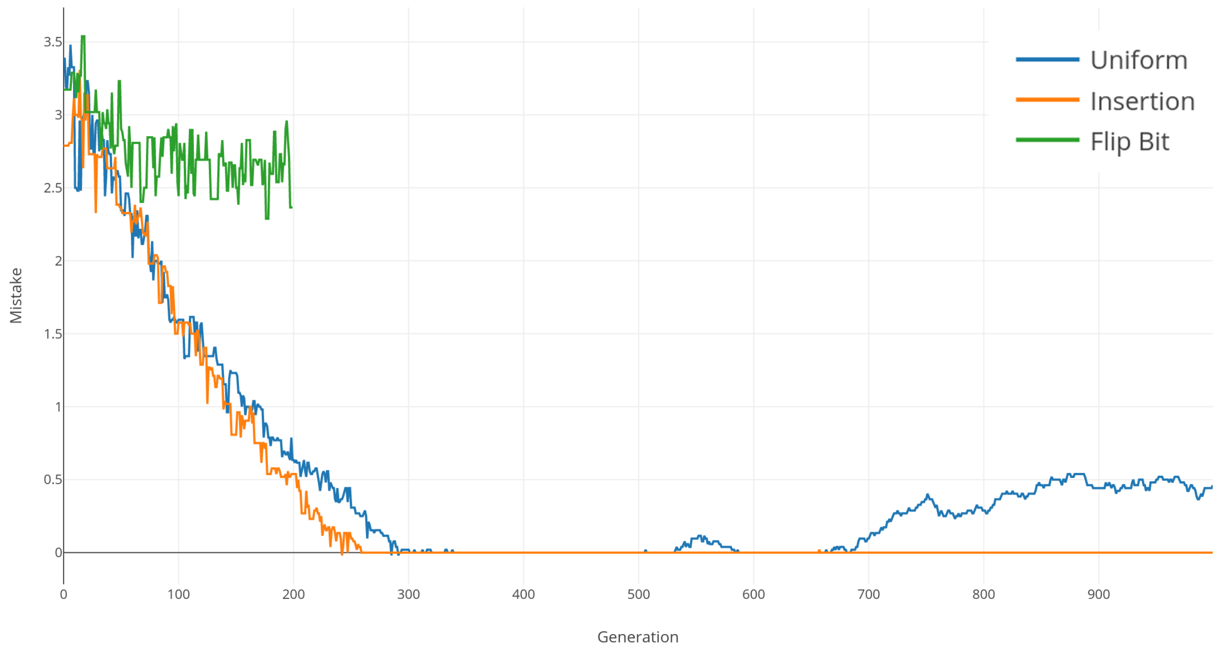


Figure 5.7: Sensors mistake depending on mutation method

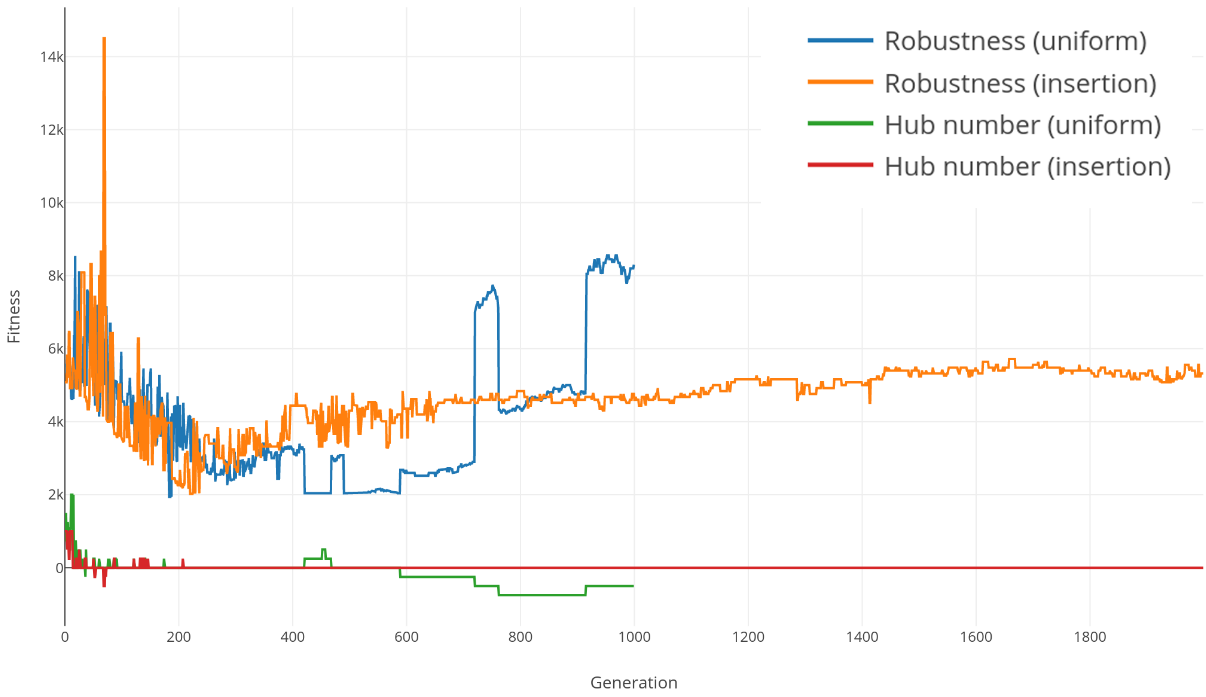


Figure 5.8: Comparison between uniform and insertion selection

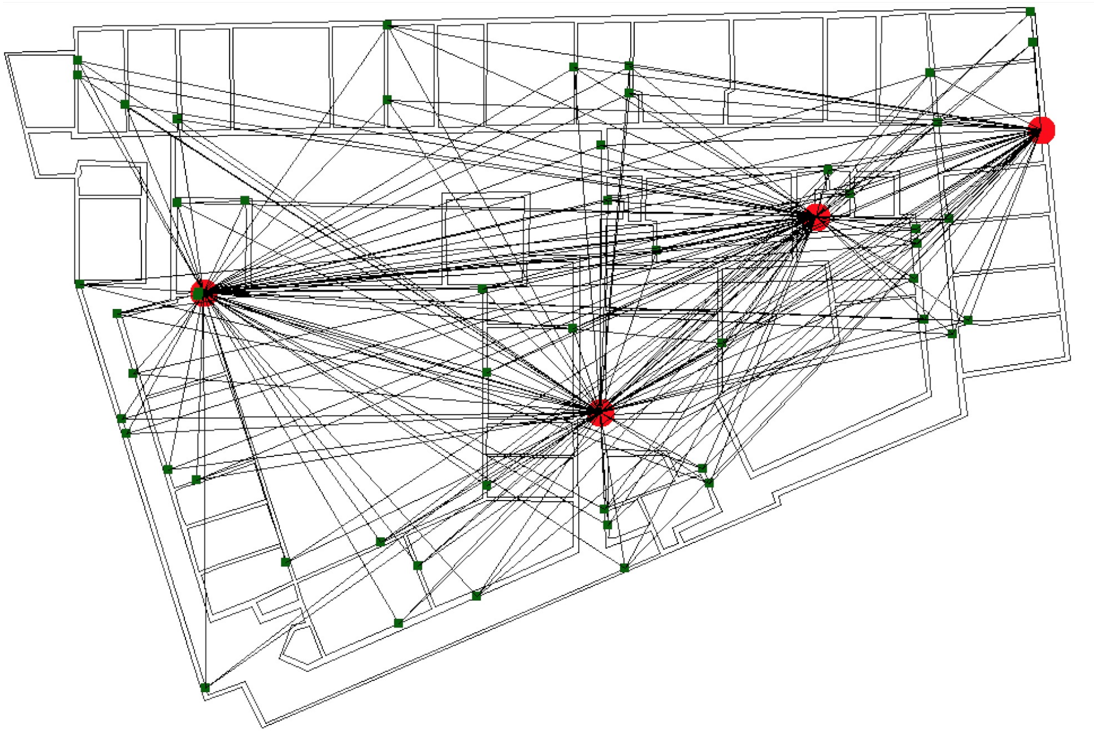


Figure 5.9: Solution received with uniform mutation method

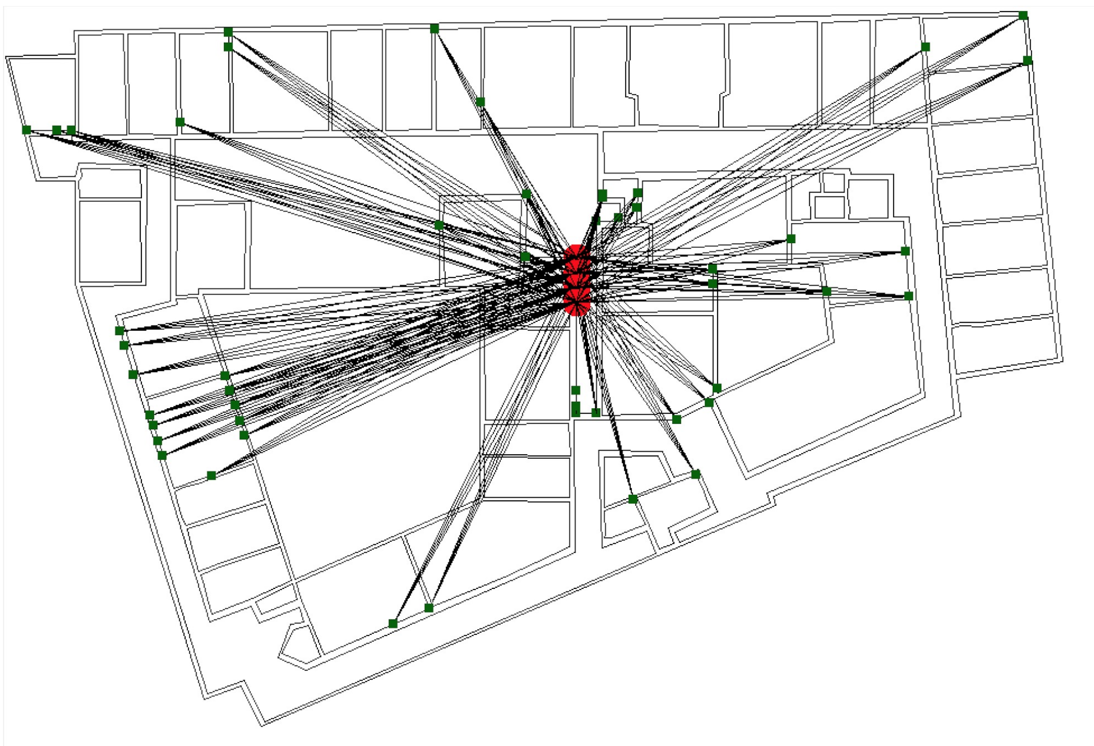


Figure 5.10: Solution received with insertion mutation method

- Fitness coefficients: 1 for high priority, 0.75 for medium priority, 0.0001 for low priority
- Device coverage: 450 m^2 for hub and 40 m^2 for sensor
- Stoping criteria: stagnation on 20 generations

This setup allows to receive an optimal result, it stops at around 300 generations with the outcome as shown in Figure 5.9 and the way to this result in Figure 5.11. The degeneracy graph in Figure 5.12 over 1000 generations shows that population is not degenerated before finding the optimal result.

5.1.6 Test case of different size

To demonstrate that the proposed set of methods works well not depending on floor area or plan we created small and medium-size demo files. The newly created test cases were four and two times smaller respectively than the original one. The medium size demo file will be used later in Section 5.2.

For the small size demo result is pretty straight-forward, we have achieved the target number of hubs and sensors, each sensor has both of those hubs in range since the distances are small. The received result is shown in Figure 5.13.

5.2 Comparison with another algorithm

For the comparison with other existing algorithms used for designing the WSN topology we have chosen paper [39]. Our works do have a lot in common:

- Devices of two types, namely sensors and hubs (gateways)
- Real building structure with walls
- Uniform grid sampling
- Radio model which counts the Wall attenuation factor

For the initial grid generation algorithm based on Self Growing Neural Gas Algorithm with the radio ray tracing propagation Motif Model was used. Then for the optimization part, they have used distributed agent-based algorithm, originally developed for the WiFi network optimization.

To compare results we have to estimate the floorplan area. Luckily, the proposed floorplan is for existing building of Environmental Research Institute based in Cork City. Using online tool we were able to estimate it's area. We are also mimicking their radio parameters to achieve the same connection range between devices.

Our settings for this experiment are:

- File Demo3.json
- Scale 3.1, area 1127 m^2
- Hub coverage 250 m^2

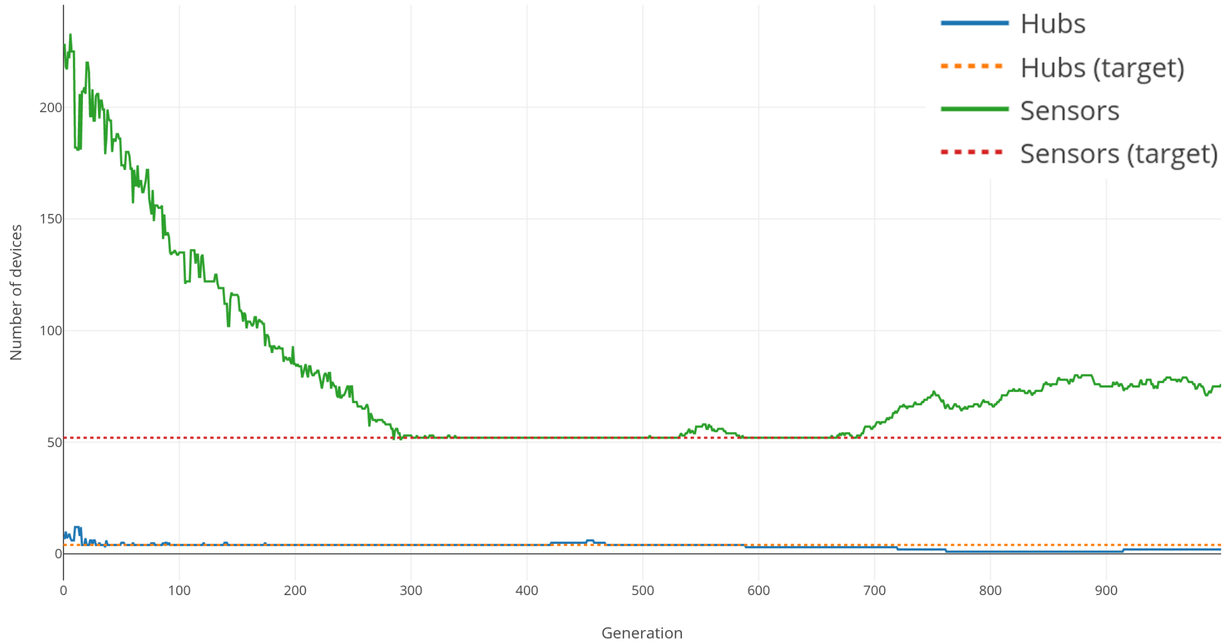


Figure 5.11: Number of devices over generations

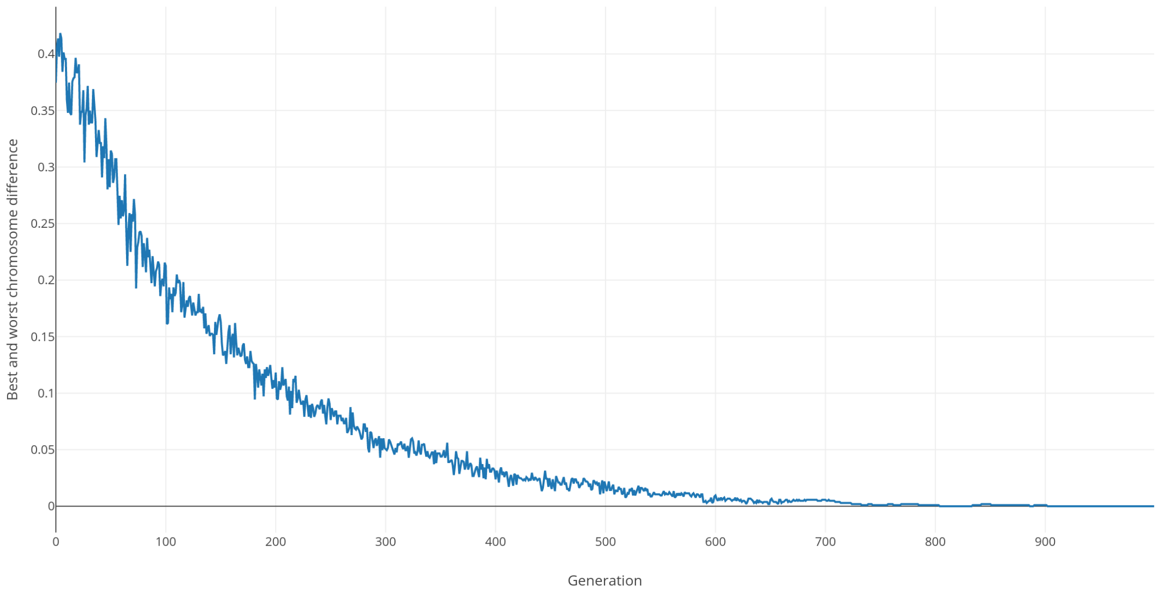


Figure 5.12: Difference between best and worst chromosome

- RX sensitivity -102 dB

Results are compared on the number of criterions in Table 5.1. Results of external and our algorithms are also visualized in Figure 5.14 and Figure 5.15 respectively.

Criterion	**Paper [39]**	**Our algorithm**
Number of sensors	29	28
Number of hubs	3	4
Number of rooms covered with sensor	8	18
Target number of rooms covered with sensor	19	30
Percentage of rooms covered with sensor	42%	60%
Number of sensors seeing at least 2 hubs	0	21
Percentage of sensors having robust connection	0%	75%

Table 5.1: Result comparison

The approach developed in Paper [39] is not resistant to changes in the network topology, since each sensor is connected to only one router. So if one of the routers fails, information from all sensors associated with it will be lost. In our thesis, a mechanism that reduces the consequences of changing the local communication topology is proposed.

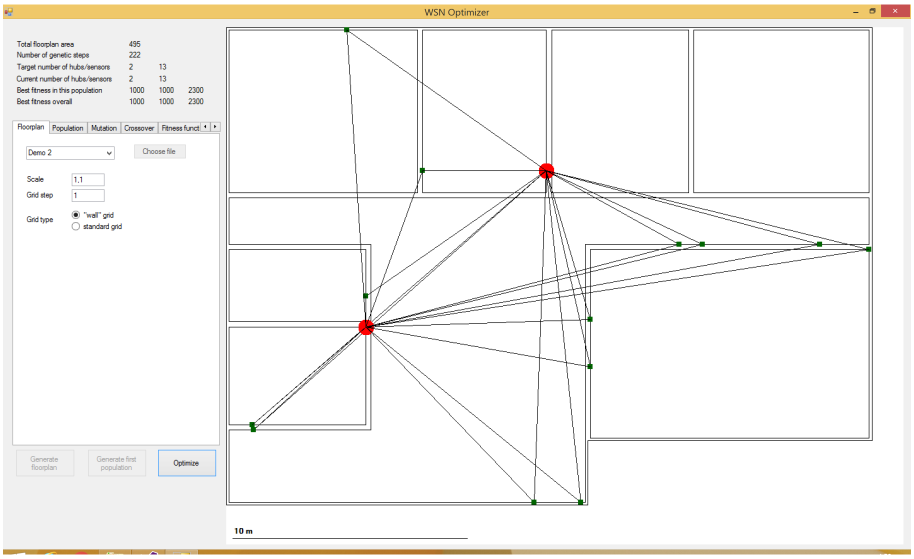


Figure 5.13: Small size test case

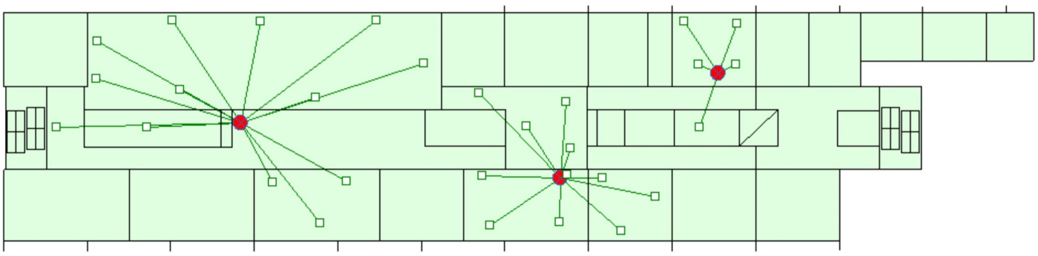


Figure 5.14: Result proposed in paper [39]

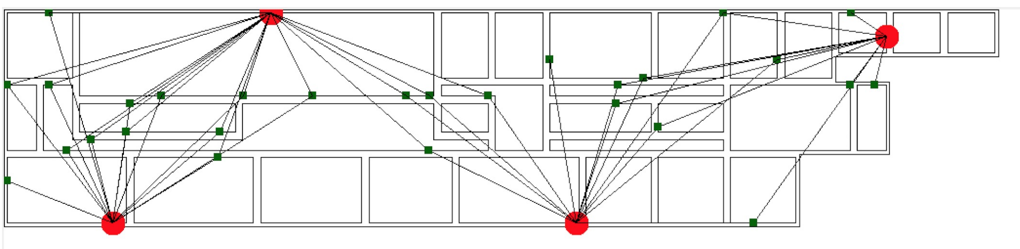


Figure 5.15: Result proposed in our thesis

6. Conclusion

This thesis has managed to satisfy all set goals. It has formulated and implemented the optimization problem of designing the Wireless sensor network topology using Genetic algorithm.

The implementation itself consists of the framework for working with real floorplans, user-friendly interface, multiple genetic operators (selection, crossover, mutation) and stopping criteria. The fitness function was defined to properly cover the area with devices and satisfy all the constraints on their number and robustness of the network.

The computational experiments with adjusting input parameters and used methods were conducted. Those experiments helped us to establish:

- lower and upper limits on the size of population
- probability parameters for generating the initial population
- combination of genetic operators which leads to converging to optimal result in our particular problem
  - 1% uniform mutation method gives us the uniform distribution of devices around the area
  - 10% half-uniform crossover method gives us a good diversity of population to achieve optimality before degeneracy
  - tournament selection is optimal from both speed and diversity points of view
- with increasing number of iterations of GA the fitness value of best chromosome in population change significantly up to a certain point after which the population degeneracy is starting and it is no longer necessary to iterate further. Therefore, the result stagnation was chosen as stopping criteria

To demonstrate the usage of those parameters and methods not dependent on chosen floor size or plan, 3 demo files have been generated and tested. Results are presented and compared to each other. One of the demo files was created in order to compare it with existing paper in this area and results of this comparison are described.

Lastly, analysis of modern scientific works devoted to the creation of models, methods and algorithms for the optimal design of the Wireless sensor network topology has been done. Based on the analysis, evolutionary modeling and genetic search methods are looking promising for the further research.

Bibliography

[1] Chun-Yen Lin, Edward T-H Chu, Lun-Wei Ku, and Jane WS Liu. Active disaster response system for a smart building. *Sensors*, 14(9):17451–17470, 2014.

[2] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications magazine*, 40(8):102–114, 2002.

[3] Sarjoun S. Doumit and Dharma P. Agrawal. Self-organizing and energy-efficient network of sensors. *MILCOM 2002. Proceedings*, 2:1245–1250, 2002.

[4] Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad hoc networks*, 2(1):1–22, 2004.

[5] Himadri Nath Saha, Abhilasha Mandal, and Abhirup Sinha. Recent trends in the internet of things. In *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual*, pages 1–4. IEEE, 2017.

[6] Bernard Sklar. *Digital communications*, volume 2. Prentice Hall Upper Saddle River, 2001.

[7] Mike McCauley. Virtualwire. `http://www.airspayce.com/mikem/arduino/VirtualWire.pdf`. 2013 (accessed May 17, 2018).

[8] Alaleh Mashkouri Najafi. Indoor path loss modeling and measurements at 2.44 ghz, 2012.

[9] M. P. Musienko, V. A. Diduk, and S. V. Kutsenko. Building M2M networks on the ZigBee technology. *Vísnik Cherkas'kogo derzhavnogo tekhnologíchnogo uníversitetu*, pages 147–149, 2009.

[10] Myung-Hee Son, Bheom-Soon Joo, Byung-Chul Kim, and Jae-Yong Lee. Physical topology discovery for metro ethernet networks. *ETRI journal*, 27(4):355–366, 2005.

[11] А. М. Тайк, С. А. Лупин, and Ю. Ф. Вагапов. Применение алгоритма перебора для оптимизации топологии беспроводных сетей. *International Journal of Open Information Technologies*, 4(9), 2016.

[12] Bang Wang. Coverage problems in sensor networks: A survey. *ACM Computing Surveys (CSUR)*, 43(4):32, 2011.

[13] Jyoti Yadav and Sandeep Mann. Coverage in wireless sensor networks: A survey. *Int. J. Electron. Comput. Sci. Eng*, 2:465–471, 2013.

[14] Anju Sangwan and Rishi Pal Singh. Survey on coverage problems in wireless sensor networks. *Wireless Personal Communications*, 80(4):1475–1500, 2015.

[15] Nor Azlina Bt Ab Aziz, Ammar W. Mohemmed, and Mohammad Yusoff Alias. A wireless sensor network coverage optimization algorithm based on particle swarm optimization and voronoi diagram. In *Networking, Sensing and Control, 2009. ICNSC'09. International Conference on*, pages 602–607. IEEE, 2009.

[16] V. Komyak, A. Pankratov, V. Patsuk, and A. Prikhodko. The problem of covering the fields by the circles in the task of optimization of observation points for ground videomonitoring systems of forest fires. *ECONTECH-MOD: An International Quarterly Journal on Economics of Technology and Modelling Processes*, 5, 2016.

[17] Károly Bezdek. Circle packings into convex domains of the euclidean and hyperbolic plane and the sphere. *Geometriae Dedicata*, 21(3):249–255, 1986.

[18] H. Jandl and K. Wieder. A continuous set covering problem as a quasidifferentiable optimization problem. *Optimization*, 19(6):781–802, 1988.

[19] Е. М. Киселева, Л. И. Лозовская, and В. И. Тимошенко. Решение непрерывных задач оптимального покрытия шарами с использованием теории оптимального разбиения множеств. Кибернетика и системный анализ, 3:98–117, 2009.

[20] А. В. Еремеев, Л. А. Заозерская, and А. А. Колоколов. Задача о покрытии множества: сложность, алгоритмы, экспериментальные исследования. Дискретный анализ и исследование операций, 7(2):22–46, 2000.

[21] Anthony So and Yinyu Ye. On solving coverage problems in a wireless sensor network using voronoi diagrams. *Internet and Network Economics*, pages 584–593, 2005.

[22] Hassan Chizari, Majid Hosseini, Timothy Poston, Shukor Abd Razak, and Abdul Hanan Abdullah. Delaunay triangulation as a new coverage measurement method in wireless sensor network. *Sensors*, 11(3):3163–3176, 2011.

[23] Loukas Lazos and Radha Poovendran. Stochastic coverage in heterogeneous sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(3):325–358, 2006.

[24] Peter Hall. *Introduction to the theory of coverage processes*. John Wiley & Sons Incorporated, 1988.

[25] Xuxun Liu and Desi He. Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks. *Journal of Network and Computer Applications*, 39:310–318, 2014.

[26] Xunbo Li and Zhenlin Wang. Cellular genetic algorithms for optimizing the area covering of wireless sensor networks. *Chinese Journal of Electronics*, 20(2):352–356, 2011.

[27] Marta Lanza, Angel L. Gutierrez, Jesus R. Perez, Javier Morgade, Marta Domingo, Luis Valle, Pablo Angueira, and Jose Basterrechea. Coverage optimization and power reduction in sfn using simulated annealing. *IEEE Transactions on Broadcasting*, 60(3):474–485, 2014.

[28] Seapahn Megerian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava. Worst and best-case coverage in sensor networks. *IEEE transactions on mobile computing*, 4(1):84–92, 2005.

[29] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 28–39, 2003.

[30] Yulai Suen. A genetic-algorithm based mobile sensor network deployment algorithm. *Technical report*, 2004.

[31] Jianli Zhao, Yingyou Wen, Ruiqiang Shang, and Guangxing Wang. Optimizing sensor node distribution with genetic algorithm in wireless sensor network. *Advances in Neural Networks-ISNN 2004*, pages 242–247, 2004.

[32] Amol P. Bhondekar, Renu Vig, Madan Lal Singla, C. Ghanshyam, and Pawan Kapur. Genetic algorithm based node placement methodology for wireless sensor networks. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 18–20, 2009.

[33] John H. Holland. Adaptation in natural and artificial systems. an introductory analysis with application to biology, control, and artificial intelligence. *Ann Arbor, MI: University of Michigan Press*, pages 439–444, 1975.

[34] Lotfi A. Zadeh. Soft computing and fuzzy logic. *IEEE software*, 11(6):48–56, 1994.

[35] Kenneth A. De Jong and William M. Spears. Using genetic algorithms to solve np-complete problems. In *ICGA*, pages 124–132, 1989.

[36] Erfan Khaji and Amin Satlikh Mohammadi. A heuristic method to generate better initial population for evolutionary methods. *arXiv preprint arXiv:1406.4518*, 2014.

[37] James Newton-King. Json.net. `https://www.newtonsoft.com/json`. 2018 (accessed May 17, 2018).

[38] Angus Johnson. Clipperlib. `http://www.angusj.com/delphi/clipper/documentation/Docs/Units/ClipperLib/_Body.htm`. 2014 (accessed May 17, 2018).

[39] Antony Guinard, Alan McGibney, and Dirk Pesch. A wireless sensor network design tool to support building energy management. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 25–30. ACM, 2009.

List of Figures

List of Tables

List of Abbreviations

GA Genetic algorithm

RF Radio frequency

WAF Wall attenuation factor

WSN Wireless sensor network

Appendices

A Digital attachment

Content and structure of the digital bundle attached to this thesis:

thesis.pdf Document containing electronic version of this thesis

WSNO Source files for the project

\WSNO

Form1.cs	Main form interface handling
genetic.cs	Genetic algorithm implementation
floorplan.cs	Floorplan parsing drawing
sensors.cs	Sensor grid creation drawing
radio.cs	Radio models connectivity matrix
geometry.cs	Helpful geometry computations
randomize.cs	Better randomization

\WSNO\bin\Debug

WSNO.exe	Executable file
Newtonsoft.Json.dll	Linked library
clipper_library.dll	Linked library
Demo1.json	Floorplan demo file
Demo2.json	Floorplan demo file
Demo3.json	Floorplan demo file