A Report on

Hybrid multi-objective evolutionary algorithms based on decomposition for wireless sensor network coverage optimization

By

Bharat Gupta 2018A3PS0446G

M Jai Sidharth 2018A8PS0660G

Sridhar Dhamija 2018A80707G

Prepared in partial fulfillment of the course Mobile Telecommunication Networks EEE F431

For

Prof. Dr. Nitin Sharma

Acknowledgements

We would like to show our gratitude to Prof. Dr. Nitin Sharma, for giving us this opportunity to enhance technical, interpersonal and communication skills through practice through this simulation project. It has helped us gain proficiency to solve real life problems, applying theoretical concepts and contemporary tools.

Table of contents

1 Introduction

A Wireless Sensor Network (WSN) is a group of spatially dispersed sensor nodes, which are interconnected by using wireless communication [1]. As seen in Figure 1.1, a sensor node, also called mote, is an electronic device which consists of a processor along with a storage unit, a transceiver module, a single sensor or multiple sensors, along with an analog-to-digital converter (ADC), and a power source, which normally is a battery. It may optionally include a positioning unit and/or a mobilization unit.

A sensor node uses its sensor(s) in order to measure the fluctuation of current conditions in its adjacent environment. These measurements are converted, via the ADC unit, into relative electric signals which are processed via the node's processor. Via its transceiver, the node can wirelessly transmit the data produced by its processor to other nodes or/and to a selected sink point, referred to as the Base Station.

As illustrated in Figure 1.2, the Base Station, by using the data transmitted to itself, is able to both perform supervisory control over the WSN it belongs to and transmit the related information to human users or/and other networks .

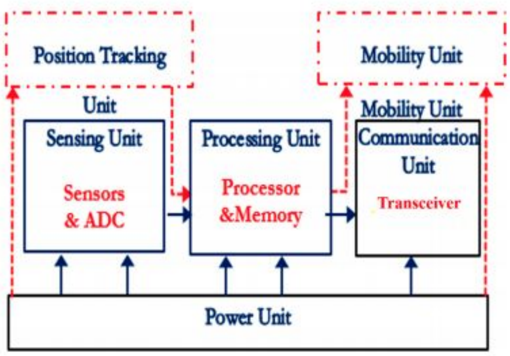


Figure 1.1. Sensor node architecture in WSN

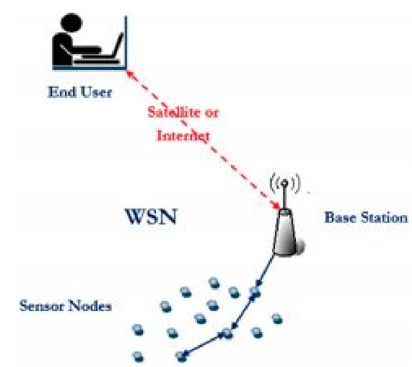


Figure 1.2. Typical WSN arrangement

The energy of sensor nodes, the network communication bandwidth and the computing ability are generally limited resources, and thus the coverage sustainability in WSNs cannot always be guaranteed. We balance the energy consumption to prolong network lifetime, while maintaining low congestion which is an important issue and can be modelled as a multi-objective optimization function (MOP) [2,3,4].

The goal to solve MOPs with two or more conflicting optimization objectives is to calculate an approximation of the Pareto Front. MOPs should be provided with multiple non-dominated solutions concerning different objectives, which are difficult to be optimized if converted into a single combined objective.

Several optimization methods have been proposed in the past, Kulkarni et. al. [5] used computational intelligence and evolutionary algorithms to solve MOPs of coverage control in WSN in complex and dynamic environments. Ozturk et al. [6] obtained better dynamic deployments for WSN by using an artificial bee colony algorithm. Kulkarni et al. [7] applied particle swarm optimization (PSO) to address issues such as optimal deployment, node localization, clustering, and data aggregation in WSN. It is shown that PSO is a simple, effective and efficient algorithm. Özdemir et al. [8] modelled the WSN coverage control problem as a MOP problem with two objectives: the coverage rate and the network lifetime. The multi-objective problem is then converted into a series of single objective subproblems, each solved by a genetic algorithm. Experimental results showed that the proposed MOEA/D algorithm outperformed an improved non-dominated sorting genetic algorithm (NSGA-II)[9]. Shen et al. [10] proposed a MOEA/D-PSO algorithm by considering two optimization objectives including coverage rate and network lifetime, and applied a particle swarm optimization algorithm in MOEA/D. Since the balance of energy consumption has a great impact on the entire network, the energy equilibrium [11] is thus added as another objective in this simulation assignment. In a novel PSO algorithm based on the jumping PSO (JPSO) algorithm developed by Xu et al. [12], a path replacement operator has been used in particle moves to improve the positions of the particles with regard to the structure of the routing tree. The experimental results demonstrated the superior performance of the proposed JPSO algorithm over a number of other state-of-the-art approaches.

In this report we will take up 3 algorithms NSGA II, NSGA III [13] and MOPJSO and compare their results based on the the objectives:

1) Number of sensor nodes required in a region of interest of fixed area.
2) Energy Consumption for the whole setup.
3) Congestion rate.

2 Optimization objectives

Objective - 1

Assume the target field Dis a two-dimensional square, the targeting point ST= { $t_1, t_2 ...., t_M$ }, t_j = (x_j, y_j) are randomly distributed in D, M is the number of grid points, j ∈ [1,. . ., M], x_j and y_j are the coordinates of each grid point. A set of sensor nodes S is randomly deployed over D where S = { $s_1, s_2, ..., s_n$ }, si= (x_i, y_i), n is the number of sensor nodes, i ∈ [1, ..., n], x_i and y_i are the coordinates of each sensor node, and r_i is the maximum ideal sensing radius of sensor node s_i .

The coverage rate must be maximised to cover all parts of the region of interest. The problem of maximising the coverage rate can be interpreted as maximising the percentage of grid points that are within the sensing region of at least one sensing node.

$$U_{s_t} = \begin{cases} 1, if \ \exists \ t_i : d(s_i, t_j) \leq r_s \\ 0, otherwise \end{cases}$$

Objective - 2

The second optimization objective is to determine the optimized value of the communication radius, r_c . The ultimate goal of each sensor in the network is to relay the sensed information to the sink. This task is accomplished by transmitting the information to the neighbouring nodes situated at a distance less than r_c . With increase in this distance the amount of attenuation in the received signal increases and therefore larger power will be required resulting in more energy consumption. Hence, it is of critical importance to have an optimized value of r_c in the network. The maximum feasible value of r_c for the network is $r_{c,max}$.

Algorithm to find $r_{c,min}$:

1. while(length(searchedNodes<=total no. of sensors)):
2. Choose a node toBeSearchedNode randomly among the set chosenNodes.
3. Add toBeSearchedNode to the list of searchedNodes.
4. Find all nodes within $r_{c,max}$ distance of the toBeSearchedNode and store it in closeNodes.
5. Delete occurrences of nodes that are present in both closeNodes and searchedNodes from closeNodes.
6. Append the closeNodes to choseNnodes and append corresponding distances values to the chosenNodeDists list.
7. Pick the toBeSearchedNode randomly from the list of nodes that are present in chosenNodes but not in searchedNodes.
8. end
9. Divide each element of chosenNodeDists by corresponding distances from the sink.
10. The objective is to minimize the $r_{c,min}$ =mean(chosenNodeDists)+std(chosenNodeDists).

Objective - 3

The third objective is to optimise the number of sensing nodes (n) in a given area. It has been set to be a maximum of 100 nodes in the simulation.

3 Nondominated Sorting Genetic Algorithm III

The basic framework of the NSGA-III algorithm remains similar to the NSGA-II algorithm with significant changes in its mechanism.

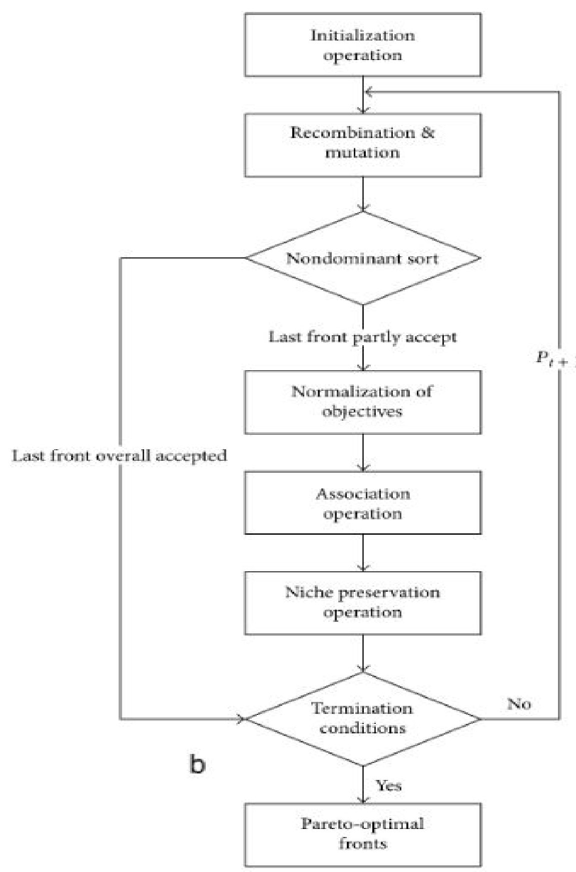
It starts with a random population of size N and a set of widely-distributed prespecified M-dimensional reference points H on a unit hyper-plane having a normal vector of ones covering the entire R^M_+ region. The hyper-plane is placed in a manner so that it intersects each objective axis at one.

The following operations are performed for t generation. First, the whole population P_t is classified into different non-domination levels, in the same way it is done in NSGA-II, following the principle of non-dominated sorting. An offspring population Q_t is created from P_t using usual recombination and mutation operators. Since only one population member is expected to be found for each reference point, there is no need for any selection operation in NSGA-III, as any selection operator will allow a competition to be set among different reference points. A combined population $R_t = P_t \cup Q_t$ is then formed. Thereafter, points starting from the first non-dominated front are selected for P_{t+1} one at a time until all solutions from a complete front cannot be included.

Algorithm 1 Generation t of NSGA-III procedure

Input: H structured reference points Z^s or supplied aspiration points Z^a, pare
Output: P_{t+1}
$S_t = \emptyset, i = 1$
Q_t = Recombination+Mutation(P_t)
$R_t = P_t \cup Q_t$
$(F_1, F_2, \ldots)$ = Non-dominated-sort(R_t)
repeat
  $S_t = S_t \cup F_i$ and $i = i + 1$
until $|S_t| \geq N$
Last front to be included: $F_l = F_i$
if $|S_t| = N$ **then**
  $P_{t+1} = S_t$, break
else
  $P_{t+1} = \cup_{j=1}^{l-1} F_j$
  Points to be chosen from F_l: $K = N - |P_{t+1}|$
  Normalize objectives and create reference set Z^r: `Normalize` $(\mathbf{f}^n, S_t, Z^r$
  Associate each member s of S_t with a reference point: $[\pi(s), d(s)]$ = `As:`
  % $\pi(s)$: closest reference point, d: distance between s and $\pi(s)$
  Compute niche count of reference point $j \in Z^r$: $\rho_j = \sum_{s \in S_t/F_l} ((\pi(s) = j$
  Choose K members one at a time from F_l to `Niching` $(K, \rho_j, \pi, d, Z^r, F_l, P_{t+1})$
end if

a



b

Figure 3.1 Algorithm(a) and flowchart(b) depicting NSGA-III working.

Front that cannot be associated to select (denoted as) F_l , P_{t+1} and F_l perform *niching and normalized mechanisms.* Each member associated with a specific reference point based on the shortest perpendicular distance of each population member with a reference line created by joining the origin with a supplied reference point. Finally, the niching mechanism chooses the F_l member that is linked with minimum reference points in P_{t+1} .

The process is then expected to find one population member corresponding to each supplied reference point close to the Pareto-optimal front, based on crossover, mutation and recombination operators that are used to develop uniform solutions. The use of a well-spread reference point ensures a well-distributed set of trade-off points at the end.
The NSGA-III algorithm has been demonstrated to work well from three to 15- objective DTLZ and other problems.

4 Simulation results and comparison between algorithms

In this report, all algorithms were implemented using MATLAB. To evaluate the performance of NSGA-III, simulation results are compared to NSGA-II and MOJPSO using the same machine and the same parameters. The experimental parameters are shown in Table 1. For each NASGA-III simulations, 20 different configurations have been randomly generated. For comparison between NSGA-II, NSGA-III and MOPJSO, 10 random configurations were generated. The comparison parameters are their optimisation performance, Non-Dominating Solution (NDS), Hypervolume estimation, Average set of coverage.

Parameters	Range/Value
Region of Interest (RoI) [m m]	1000
Sink node position [x,y]	500,500
Sensing Range (R_s) [m]	$1 \leq R_s \leq 200$
Connectivity Range (R_c) [m]	$1 \leq R_c \leq 150$
Number of sensors (n)	$ceil(RoI^2/\pi.R_s^2) < n < 100$
Number of grid points	100

Table 1

4.1 NSGA-III Simulation Results

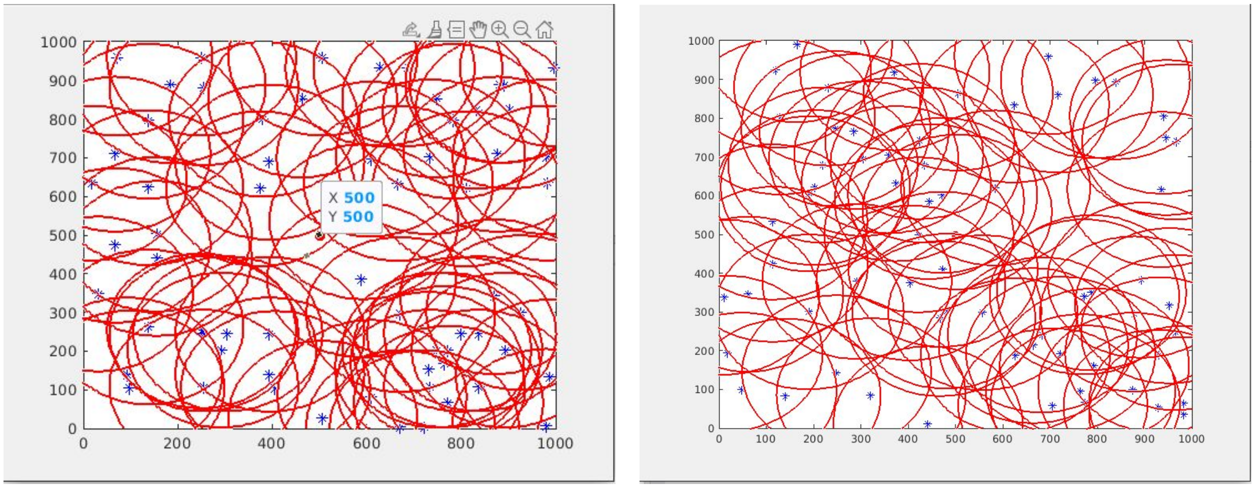


Figure 4.1. Initial setup of WSN region

Two different random initial setup of WSN networks for the simulation is shown in Figure 4.1. The region is an area of 1000x1000 m^2 with the sensor node placed at 500,500 (x,y). And the red circles show the individual sensing radius of each of the nodes which vary between 1m to 200m for each. Figure 4.2 shows the optimization of NSGA-III between the three objective functions.

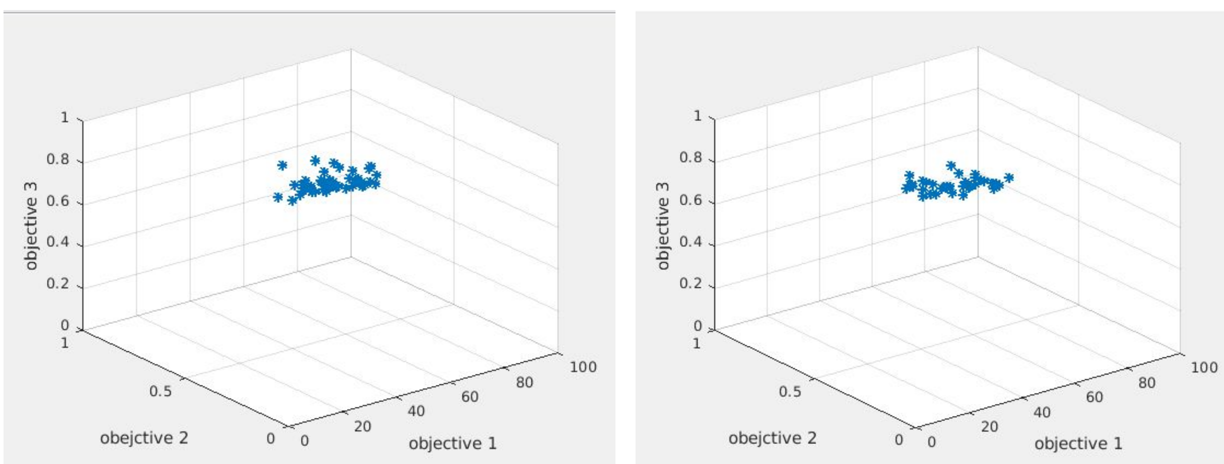


Figure 4.2. Optimization performed by NSGA-III against number of nodes (objective 1), Energy consumption (objective 2) and congestion rate (objective 3)

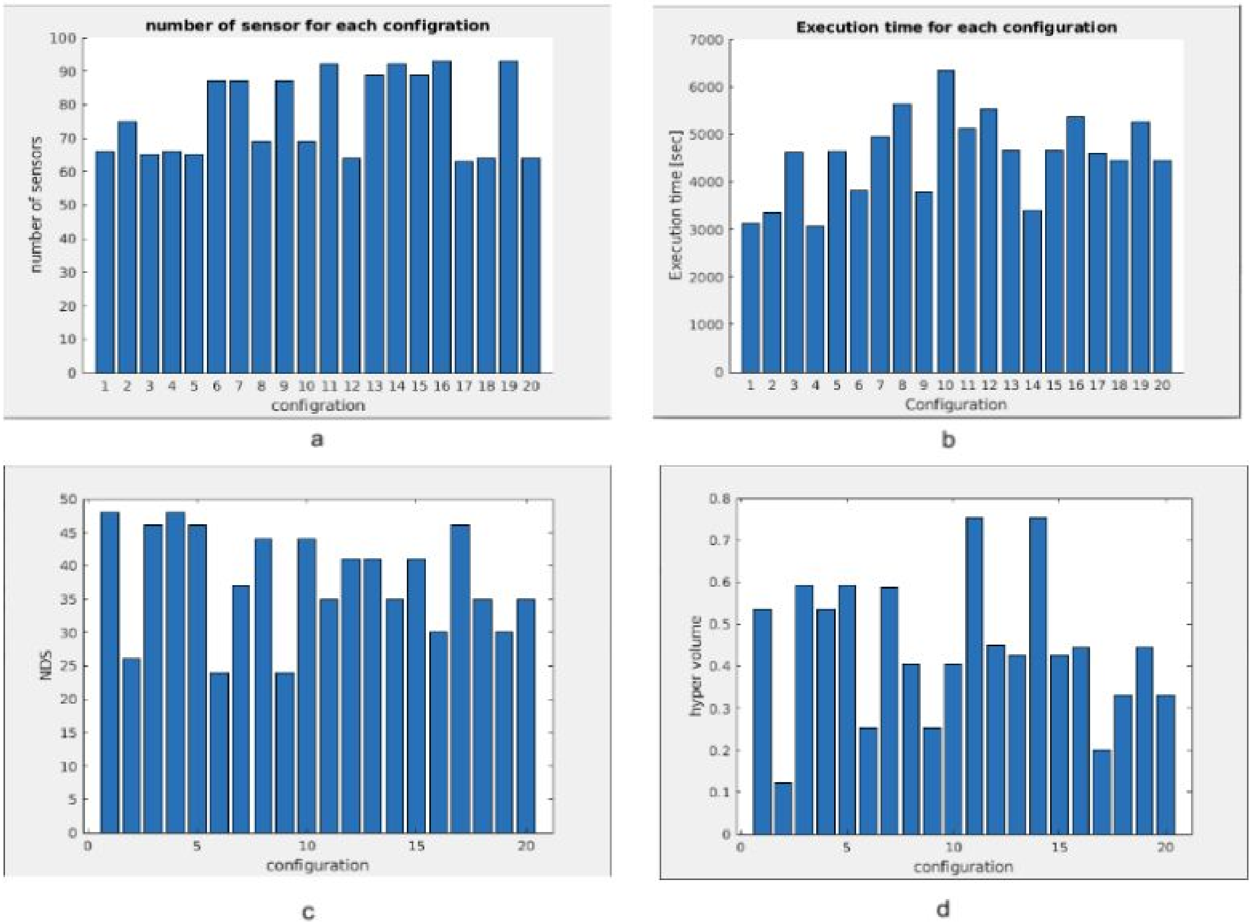


Figure 4.3 Various statistics of the simulation

Figure 4.3 shows various simulation statistics for the 20 different randomly generated configurations. Taking into account the number of nodes from plot 5.3 (a) we can see the optimization time is longer for more number of nodes, which is expected. The Non-Dominating Solution (NDS) in plot 5.3 (b) is greater for lesser number of nodes (in the range of 60-70 nodes) and the plot 5.3 (d) shows the hypervolume indicator, which is a set measure used in evolutionary multiobjective optimization to evaluate the performance of search algorithms and to guide the search. The configurations with lower number of nodes (in the range 60-70) tend to have higher value of the hypervolume indicator. The higher the hypervolume value the better.

4.2 Comparison of NSGA-II, NSGA-III and MOPJSO

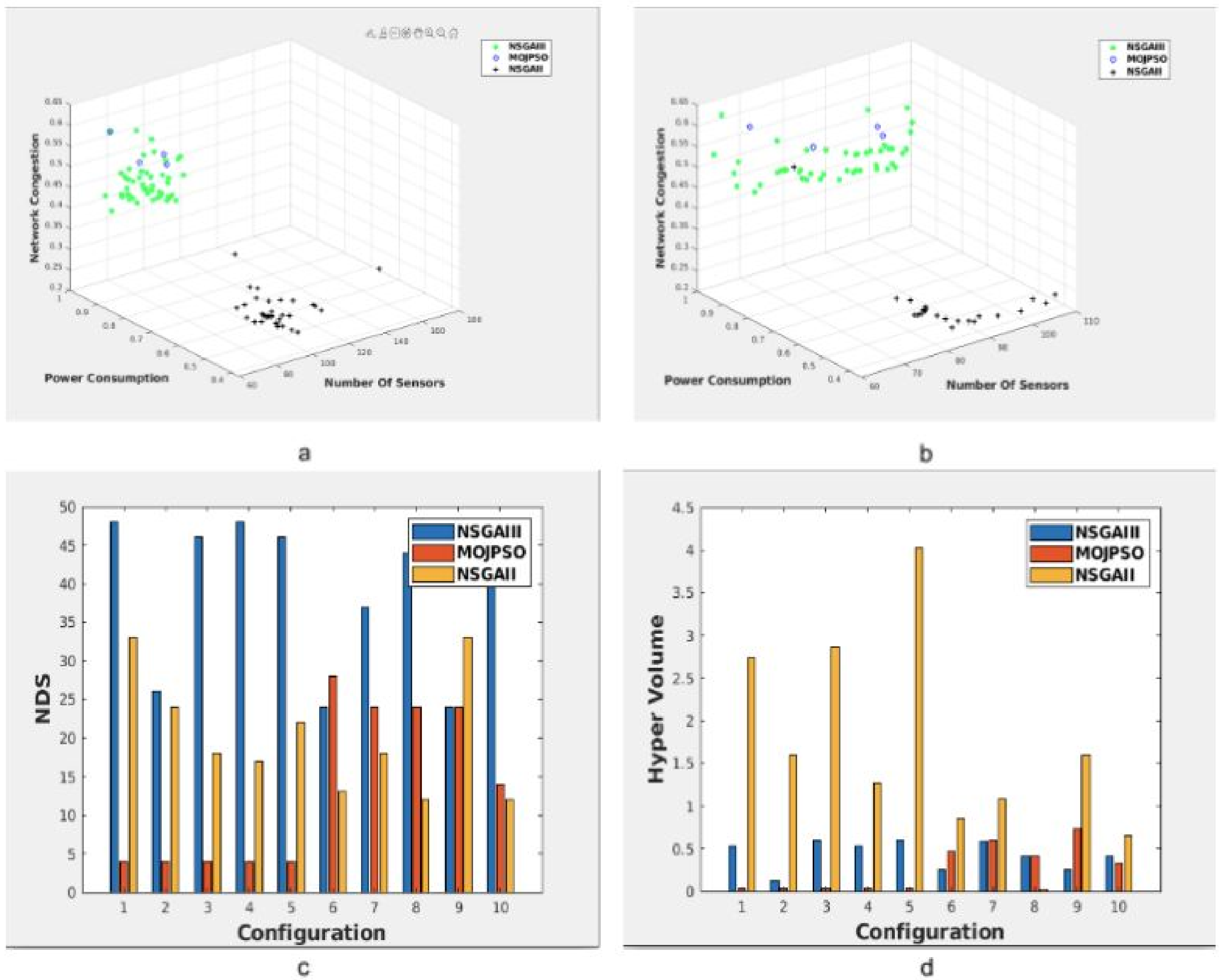
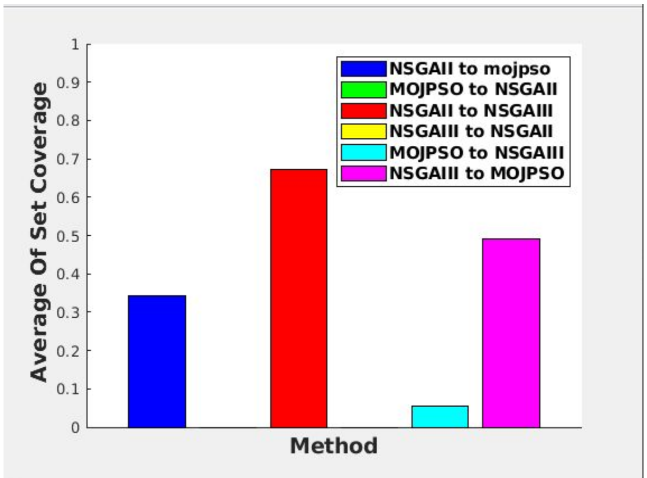


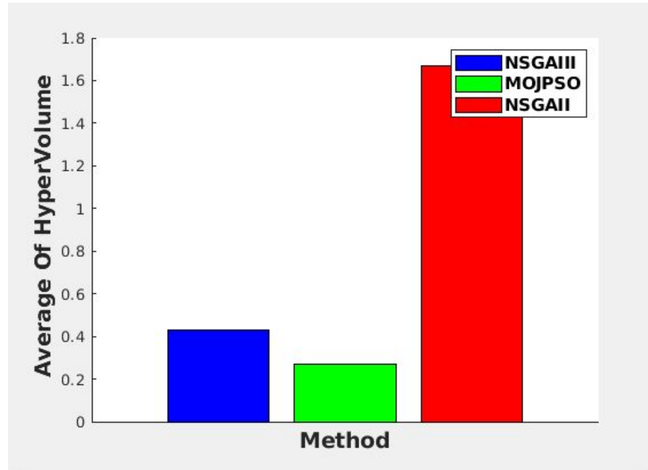
Figure 4.4 Comparative plots between NSGA-II, NSGA-III and MOPJSO

For comparison of the three algorithms NSGA-II, NSGA-III and MOPJSO they were run against the same parameters given in Table 1. Each of the WSN optimization algorithms 10 random configurations were made. Figure 4.4 (a) and (b) shows the 3-D plot of the optimization between the 3 algorithms in two different configurations.

From Figure 4.4 (c) we can see that NSGA-III has the highest Non-Dominating Solution (NDS) of the three algorithms with NSGA-II having the second highest average and MOJPSO having the lowest. In general in minimizing multi-objective optimization problems we aim to maximise the hypervolume indicator. Interestingly NSGA-II is found to have the highest hypervolume indicator value of the three algorithms through all the configurations, as seen from Figure 4.4 (d).



a



b

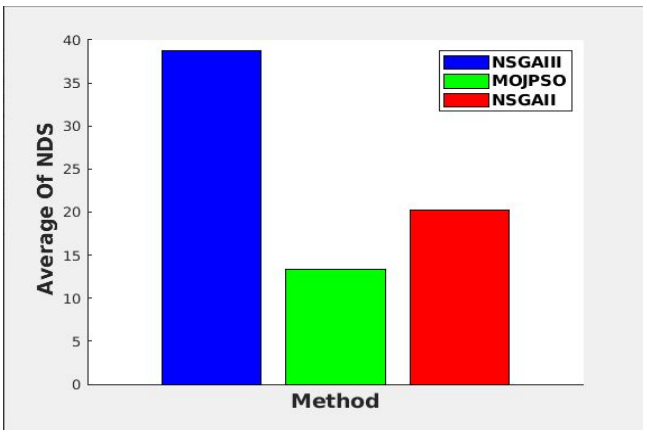


Figure 4.5 Average of the comparative measurement between NSGA-II, NSGA-III and MOPJSO

Figure 4.5 (a) shows the average of set coverage between the three algorithms. Set coverage C(A,B) is a quality indicator factor which shows the percentage of solutions from algorithm B that are weakly dominated by A. From the graph we can see that NSGA-II provides the best coverage amongst the three algorithms which is also seen from the hypervolume graph in Figure 4.5 (b). However the highest average NDS value is of NSGA-III as seen in Figure 4.5 (c).

5 Conclusion

Wireless Sensor Networks (WSNs) can be defined as a self-configured wireless network to monitor physical and environmental conditions, such as temperature, seismic vibrations, pressure, motion or pollutants amongst other various use cases. A sink or a base station acts like an interface between users and the network and is used to retrieve information from the sensor networks. To make the WSNs more efficient, prolong their life and reduce costs and increase Quality of Service (QoS) several Multi-Objective Optimization Problems (MOPs) algorithms have been developed and researched upon.
In this simulation assignment we got in an in depth understanding of how NSGA-III algorithm can be used to tackle three objectives related to WSNs which were - Number of sensor nodes required in a set region of interest, the Energy consumption of the whole configuration and the Congestion that is achieved by the WSN. NSGA-III algorithm was then further compared to two other algorithms namely NSGA-II and MOPJSO. The comparisons have been discussed in section 5 of this report.

Much scope of improvement for WSNs lies in the network security - Confidentiality, Data Integrity, Accountability - and target detection, time critical applications etc. WSN use cases are also being explored in the area of neural networks. Neural Networks consist of multiple layers, each layer separated by a non-linear activation function. Each layer consists of multiple 'neurons'. A visual representation of a Neural Network would be as follows:

These networks are initialized arbitrarily, and trained on a data-set using algorithms like back propagation. Once a neural network is trained on a data-set it can be deployed on real time inputs and the outcome can be evaluated.

Each neuron can be imagined as a node in a sensor network. The data collected in real time, can also be processed in-network and the WSN can be used to make smarter decisions.

References

[1] Kandris, D.; Nakas, C.; Vomvas, D.; Koulouras, G. Applications of Wireless Sensor Networks: An Up-to-Date Survey. Appl. Syst. Innov. 2020, 3, 14.
[2] Hybrid multi-objective evolutionary algorithms based on decomposition for wireless sensor network coverage optimization
[3] Xi.-xi. Xiang, H.-G. Huang, Y.-d. Li, Hybrid sensor networks coverage-enhancing approach based on particle swarm optimization, Appl. Res. Comp. 27 (6) (2010) 2273–2275
[4] J. Jia, J. Chen, G. Chang, Y. Wen, J. Song, Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius, Comput. Math. Appl. 12 (2009) 1767–1775.
[5] R.V. Kulkarni, A. Förster, G.K. Venayagamoorthy, Computational intelligence in wireless sensor networks: a survey, IEEE Commun. Surv. 13 (1) (2011) 68–96.
[6] C. Ozturk, D. Karaboga, B. Gorkemli, Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm, Sensors 11 (11) (2011) 6056–6065.
[7] R.V. Kulkarni, G.K. Venayagamoorthy, Particle swarm optimization in wireless-Sensor networks: a brief survey, IEEE Trans. Syst. Man Cybern. Part C 41 (2) (2011) 262–267.
[8] S. Özdemir, B.A.A. Attea, Ö.A. Khalil, Multi-Objective Evolutionary Algorithm Based on Decomposition for Energy Efficient Coverage in Wireless Sensor Networks, Wireless Pers. Commun. 71 (1) (2013) 195–215.
[9] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197
[10] X. Shen, J. Li, Q. Zhang, WSN coverage hierarchical optimization method based on the improved MOEA/D, Metall. Min. Ind. 7 (6) (2015) 348–354.
[11] T. Liang, H. Zhou, J. Xie, K. Wang, Multi-objective coverage control strategy for wireless sensor networks, Chin. J. Sens. Actuators 23 (7) (2010) 994–999.
[12] R. Qu, Y. Xu, J.P. Castro, D. Landa-Silva, Particle swarm optimization for the Steiner tree in graph and delay-constrained multicast routing problems, J. Heuristics 19 (2) (2013) 317–342.
[13] H. Jain and K. Deb, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach," in *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602-622, Aug. 2014, doi: 10.1109/TEVC.2013.2281534.