Critical Water Infrastructure Sensor Placement Optimization

Lan Jiaqi

November 7, 2018

Critical Water Infrastructure Sensor Placement Optimization

Student:

Lan Jiaqi
Student number: s1985825
E-mail : j.lan.2@umail.leidenuniv.nl
Leiden Institute of Advanced Computer Science

Supervisors:

Dr. Michael Emmerich
E-mail: m.t.m.emmerich@liacs.leidenuniv.nl
Leiden Institute of Advanced Computer Science

Dr. Kaifeng Yang
E-mail: k.yang@liacs.leidenuniv.nl
Leiden Institute of Advanced Computer Science

November 2018

Contents

Acknowledgements

During the six months project, Dr. M. Emmerich kindly offers his help throughout the process and provides insights and structured guidance to help me finish the report on time. The inspiration and guidance I get from him are not only during the thesis writing itself but also during the class that I took in my first year which facilitate my interest in the choice of my thesis topic. I would love to express my gratitude towards him.

Meanwhile, Dr. Kegong Diao from De Montfort University also provides kind instructions to help me understand water quality simulation tool and discuss the objectives selection. Allowing me to access his research outcome which also contributes one of the key results from this paper. Additionally Dr. Kaifeng Yang kindly agreed to become the second supervisors and gave valuable feedback on the thesis.

Lastly, I would like to thank my family, parents, and friends, and present this thesis to them. I had a great experience throughout my master program which their encouragement is a significant factor.

Abstract

Water quality is essential to civilian daily life. There is an urgent need to place reliable and trustworthy sensors to minimize the impact of water contamination incidence resulting from deliberate attacks,systems failures or from careless mistakes. To achieve the sensor placement strategy, optimization methods are suggested to be the most effective due to its automatic placement ability. However, the problem of high computational cost of optimization methods(Genetic Algorithm in particular) is pressing and urgent. Meanwhile, there are different views on how many sensors are required for a complex water distribution system. From Diao's [1] finding, approximately 33% to 40% of nodes will help to have a good overview of a complex water distribution system while others claim substantially smaller numbers would work. In this thesis, we improved one of the current optimization methods, Genetic Algorithm, by applying Centrality Mutation and Seed Initiation. The performance from the improved algorithm provides a better solution set for Multi-objective Decision Analysis on Nodes Selection . Additionally, we develop a direct comparison over a set of different number of nodes to detect their performance on Water Contamination Detection. The analysis in nodes number will contribute to detecting how many nodes are needed in the case of Water Contamination Detection.

Chapter 1

Introduction

1.1 Problem Motivation

Clean water is one of our necessities for survival, besides food and sunlight. However, water pollution occurs when harmful substances—often chemicals or microorganisms—contaminate a stream, river, lake, ocean, aquifer, or another body of water, degrading water quality and rendering it toxic to humans or the environment.

Water is uniquely vulnerable to pollution. Known as a "universal solvent," water can dissolve more substances than any other liquid on earth. It's the reason we have brilliant blue waterfalls. It's also why water is so easily polluted. Toxic substances from farms, towns, and factories readily dissolve into and mix with it, causing water pollution. Among all these substances, accidental or intentional contamination events are common nowadays.

Because water is easily polluted, the pollution impact is catastrophic and irreversible; history has witnessed multiple incidences that many stakeholders are tremendously harmed. In 2016 Boca Bay, Florida, 300,000 gallons of sewage were dumped into Boca Ciega Bay in Florida. Storm Hermine dumped more than 900,000 gallons of waster on the bay was dumped again. The accident led to a huge problem in the red tide and continued to wreak havoc on the marine life [2]. Moreover, related to water contamination terrorist activities awaken public attention in the United States during the Civil War, in Europe and Asia during World War II and other countless deliberate accidents. Hickman(1999) and Deininger Meier(2000) researched the topics of deliberate contamination of the water supply system [3][4]. Regarding the pollution impact, take the Hinckley Water Contamination [5] as an example. Residents showed the increased risk of breast, lung, brain and gastrointestinal cancer, kidney and ovarian tumors, miscarriage and Hodgkin's disease.

These activities shall be properly responded to minimize the effect happens to the city and civilians. Early contamination is the key to solve these stated problems.

1.2 Current Approaches Overview

For the last few years, there is increasing interest in development Contamination Warning Systems(CWSs). The overall objective is to obtain a cost-effective approach to mitigating the tremendous impact listed in Section 1.1 [6][7][8]. A CWS is an integrated system that gathers data from online sensors with multiple detection strategies including public health surveillance systems, physical security monitoring, customer complaints, and routine sampling programs. A CWS facilitates efficient decision making procedures and timely response to the detected harmful events [8]. One of the remarkable advantages is CWS deployment would decrease a substantial cost related to the contamination incidents. Take the EPA's Threat Ensemble Vulnerability Assessment (TEVA) as an example, with CWS deployment, TEVA could reduce related fatalities by 48% and corresponding economic cost by over 19 billion dollars [9].

With multiple technical challenges in hand to facilitate CWSs as trustworthy and feasible water contamination prevention technology, one of the most important technologies is sensor placement within the distribution network. Since it would not be cost effective to place all sensors, the major challenge is to minimize the public health impact from any contamination intrusion with few sensors [10]. There two main types of knowledge domains to explore in sensors placements. One is technical domain characteristics, and the other is to use a computational model. Three approaches are widely accepted in the literature: expert opinion [11][12], ranking methods [13][14][15] and optimization methods. Expert opinion is solely guided by human judgment while ranking methods is to use empirical methods(expert opinions) to rank potential sensor locations. Spatial information is used to ensure good coverage in ranking methods.

Optimization methods stand out due to their ability to enable automated placement based on hydraulic and water quality simulation. There are an increasing number of researchers who devote their time on optimization method based sensor placement including a battle of the water sensor networks (BWSN) that compared 15 different approaches [16].

Among the 15 different approaches, the available options are mostly based on integer programming [17](e.g. Lee and Deininger 1992), mixed integer programming [18](e.g. Propato, 2006), heuristic-based algorithm [19](e.g.

Dorni et al. 2006), graph theory algorithms [20](e.g. Kessler et.al 198) and genetic algorithm schemes [21](e.g. Ostfeld and Salomons 2004).

1.3 Existing Problems

During the BWSN, nearly optimal sensors layout [22] were found. However, as pointed out by Hart and Murry, knowledge gap and obstacles still applies. Computational inefficiency is one of the biggest problems among all the gaps and obstacles. Distribution networks are usually complex networks which require high computational effort for proper water quality simulation while optimization algorithms require a high computational cost for a decent result. For example, Genetic algorithm schemes [21] (e.g. Ostfeld and Salomons 2004) require a huge number of iterations which is hard to obtain in many complex networks. Therefore, researchers would see a clear trade-off between network complexity versus optimization method complexity. Therefore, simplification of network and enhancement of existing algorithms are highly needed.

Beyond the general problems, during optimization methods, researchers are deploying a fixed number of sensors. However, number of sensors can be incorporated as one of the important objectives to optimize. Some sensors are relevant to the cost of labor and its physical cost as well. A better solution is to provide clear guidance over how many sensors are exactly needed during the process of Water Contamination Analysis.

According to controllability pre-selection method for sensor placement in water distribution system [1], researchers need approximately 33% to 40% of nodes to represent the entire network while other scholars argue for fewer nodes. There are no universally acknowledged facts over nodes number distribution in different networks.

1.4 Research Questions

The research questions discussed above, can thus be summarized as following:

1. Given the high computational cost of optimization algorithms(Genetic Algorithms in particular), how can we improve its efficiency by modifying existing algorithms?

2. Given the existing network and algorithms, how many nodes would it require to represent the entire network approximately?

1.5 Overview

The rest of the thesis report is structured in five parts. In Chapter 2, the objective is to discuss the simulation tool for water contamination and different objectives to evaluate contamination incidents in optimization methods. In Chapter 3, the goal is to improve the existing Genetic Algorithms and providing a comparison of improved algorithms, original algorithms, and optimal global result. In Chapter 4, different nodes and performance evaluations are listed to testify what would be a good approximation of given network.Lastly, Chapter 5 introduces recommendation for future development.

Chapter 2

Water Distribution Network

In this Chapter, we will discuss Water Distribution Network Simulation Tool and its usage. Moreover, we will define objective functions for the optimization.

2.1 Motivation

Regarding water quality detection, research claims that the quality of findings may fluctuate due to several real-world changes which include loss of disinfection residuals that lead to bacterial re-growth [23], development of taste and odour and corrosion. These concerns push the requirements of modelling to an increasingly tough standard.

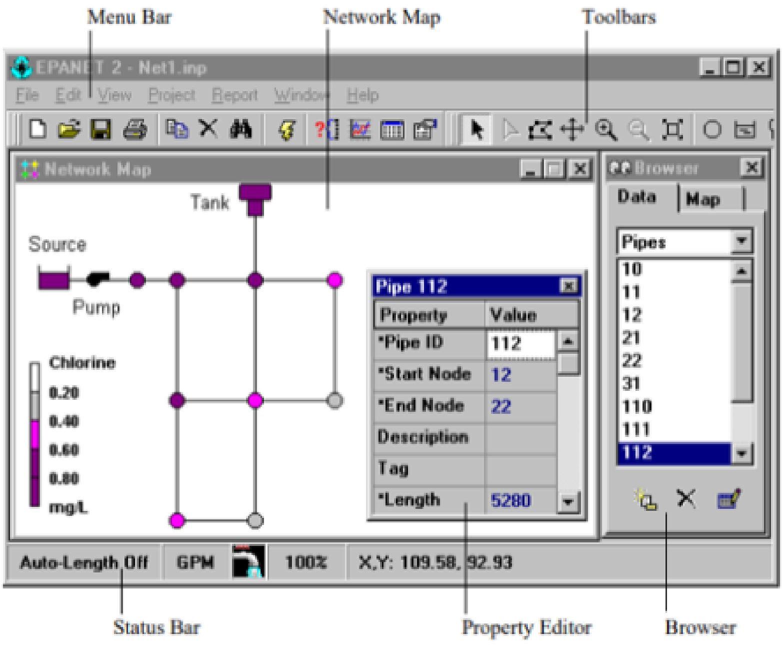
We have physical modelling and simulation modelling to choose from. Simulation modelling provides safe and efficient solutions to the real-world problem. Simulation modelling enables easy verification, communication and understanding [24]. Across different industries and disciplines, simulation modelling is crucial for providing insights into a complex network.

Particularly in the incidence of water contamination, real-world history and examples are not sufficient to deal with potential harm and impact which have not happened in history. A valid digital representation to provide a risk-free environment and dynamic insights is crucial for accurate representation of real-world scenarios. While physical modelling is money and time consuming, it would not be able to handle uncertainty due to its fixed structure. Therefore, a simulation tool for water contamination detection is highly needed.

2.2 Simulation Tool EPANET

There are many water quality processing simulation tools in Water Distribution Systems. Among them, EPANET is the most commonly used software application. EPANET is a public domain, water distribution system modeling software package developed by the United States Environmental Protection Agency's (EPA) Water Supply and Water Resources Division. This application consists of pipes, nodes(junctions), pumps, valves, storage tanks, and reservoirs which can perform an extended-period simulation of the hydraulic and water quality behaviour within a pressurised pipe network.

EPANET provides a visual network editor that can help engineers, consultants and researchers to understand the movement and fate of drinking water constituents within the distribution system, design and size new water infrastructure, retrofit existing ageing infrastructure, optimize operations of tanks and pumps, reduce energy usage, prepare for emergencies and many other real-world analytical questions [25]. Extension package of EPANET (e.g. EPANET-MSX) [26] enables simulation of multiple chemical species in bulk water as well as the pipe wall.



Pipe 112

Property	Value
*Pipe ID	112
*Start Node	12
*End Node	22
Description	
Tag	
*Length	5280

Figure 2.1: An overview of User Interface of EPANET

2.2.1 Hydraulic Modeling

Hydraulic Modeling is an entry bar for high-quality modelling. EPANET enables several functionalities [27]:

1. System operation complexity ranges from simple tank level to complex rule-based control
2. Allowing Complex Network and minor head losses for bends, fitting, etc.
3. Modelling constant or variable speed pumps and computing pumping energy
4. Modeling various types of valves and allow storage tanks to have any shape

2.2.2 Water Quality Modeling

Except for Hydraulic Modeling, EPANET also enables Water Quality Modeling as follows:

1. Percent of flow from a given node reaching all other nodes over time
2. Allows wall reaction rate coefficients to be correlated to pipe roughness
3. Movement of a non-reactive tracer material through the network over time

These two functions allow us to achieve the objective functions stated in later sections.

2.3 Network and Sensor Placement

2.3.1 Introduction of BWSN

Though there is increasing interest in the sensor placement optimisation method, the comparison between different optimisation methods is not known to the majority of the researchers.

Therefore, the Battle of Water Sensor Networks(BWSN) [28][29] as part of the Eight Annual Water Distribution System Analysis Symposium in Cincinnati is designed to explore the comparison between different methods. Fifteen research teams explored the same design objectives which design the performance metrics and contamination events characteristic with two networks.

2.3.2 Network Introduction

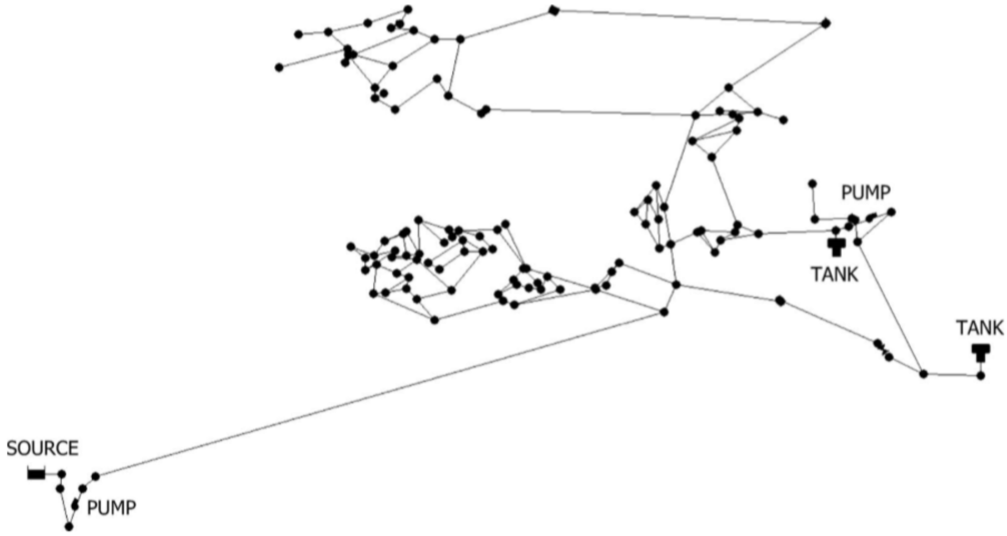


Figure 2.2: Graphical Outline of Network

The Network in this thesis consists of 126 nodes, 1 constant heat source, 2 tanks, 168 pipes, 2 pumps, 8 valves, and was subject to four variable demand patterns. The system was simulated for a total extend period duration of 96h.

This network was based on real water distribution systems and was "twisted" to preserve anonymity. Space limitations prohibit the description of all their details, including diameters, elevation, pipe lengths and many more [28].

2.4 Sensor Placement Evaluation Objectives

To evaluate sensor performance in Water Contamination Detection, we use the following quantitative design objective values:

2.4.1 Expected Time of Detection

In each possible contamination event, Expected Time of Detection is the elapsed time from when the contamination start until the event that the first sensor detects a nonzero contamination concentration.

The notation t_j denotes the first time of detection in the jth sensor location and t_d denotes the minimal t_j value which refers to the minimal time of detection among all sensors. n denotes the overall sum of sensor candidates number.

$$t_d = \min_j t_j, j = [1, n]$$

The objective function is to minimize the average over all t_d, which is the detection time of each contamination event. In general, a better sensor combination is aiming for a smaller Average Time of Detection. t_s denotes the allowed biggest simulation time. Note, the variable is subject to the limitation of finite-simulation durations used to compute their value, which means t_s cannot exceed 96 hours in the network we are using. If t_s is not within the accurate scope, the objectives including S_j are not valid anymore.

$$F_1 = \frac{\sum_{i=1}^{d} t_d}{d} \rightarrow min, d = [0, t_s]$$

2.4.2 Detection Network Coverage

In each possible contamination event, we denoted with S_j for the jth sensor how many water contamination events they can detect. In a sensors collection with total sensors number k, the objective function is to aggregate the sum of each node within the collection(k number of nodes). We are aiming for maximising this objective function.

$$F_2 = \sum_{i=1}^{k} S_j \rightarrow max$$

2.4.3 Objective Selection Motivation

EPANET allows us to explore more possible objectives. According to BWSN [28](last two), there are two more objectives which are Expected Population Affected Prior Detection(F_3), Expected Consumption of Contaminated Water Before Detection(F_4).

However, the study(cite) shows that F_1, F_3, F_4 are positively correlated to one and another while all of them are negatively correlated with F_2. One future development the study is aiming for is to pick one of the objectives among F_1, F_3, F_4 and then compare to the selection of F_2 in Multi-objective Decision Analysis. Therefore, in the research, we only select F_1,F_2 as our objectives.

2.5 Experiments and Results

To obtain objective function values, we need to utilise the EPANET dynamically linked library.

2.5.1 Dynamically Linked Library

The EPANET Programmer's Toolkit is a dynamically linked library (DLL) of functions that allow developers to customise EPANET's computational engine for their own specific needs [30]. The functions can be incorporated into 32-bit Windows applications written in C/C++, Delphi Pascal, Visual Basic, or any other language that can call functions within a Windows DLL.

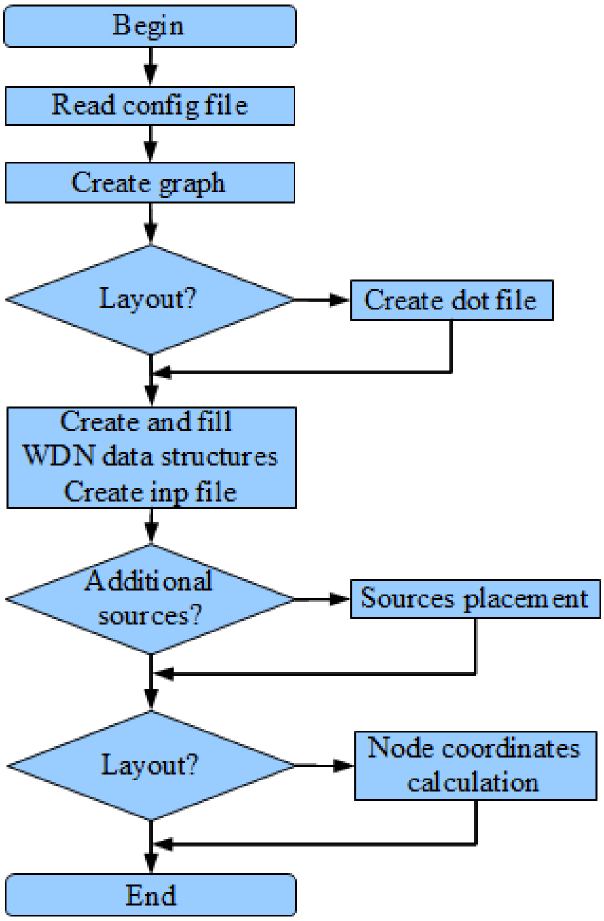


Figure 2.3: EPANET Data Flow

The description of the network can be simulated from an external input file(.INP), After parsed, interpreted and saved in a shared memory area, the hydraulics solver module carries out an extended period hydraulic simulation. The temporary result can be stored in a hydraulic file(.HYD). If a water quality simulation is requested the water quality module accesses the flow data FRP, the hydraulics file as it computed substances transport and reaction throughout the network over each time step. Lastly, after both steps, a binary output file(.OUT) can be used for reporting(.RPT). Any error or warning are also added to the file.

2.5.2 Objective Function Values

To identify each sensors capability to detect contamination, we need to investigate their detection coverage and detection time. Both of these objectives are exactly the optimisation objectives in this report. To aggregate the data, a contaminant is injected in every possible injection node of the entire network and the EPNAET model is executed. The simulation period is 96h, with time steps for both hydraulic and water quality equal to 5 mins. Once the contamination has reached to any node where the concentration of the contaminant at each driver node is > 0, the event is regarded as being detected, and the time, until detection is recorded as the t_j, and used to compute objective functions. Otherwise, the event would be regarded as undetected.

These steps are repeated until all nodes have been considered as the contamination source where there are 129 contamination events in total. The nodal injection of the contaminant is simulated by imposing a mass booster source(set as 300 mg/L) for the duration of 2h from the beginning of the simulation period.

To ensure the contaminant would not vanish in the system, the settings in the EPANET software are chosen as Bulk Reaction Order = 1.5; Global Bulk Coefficient = -1.0; Limiting Concentration = 0.01. Additionally, the dead-end nodes which referring to the downstream end of a branch pipe(nodes 13, 16, 36, 38 and 125) are assigned a base demand of 2.0 gallons/minute. Subsequently, all pipes become directed links as the controllability analysis deal with the directed networks.

Regarding coverage, once a node is detecting a contamination event, these incidence is marked. After the entire set of contamination events are done, we check the number of marked events in each node as the coverage indication of this potential sensors.

In the Full Result Table, the biggest detection time is 87900 s. This is roughly 24.4 hrs which is smaller than 96 hrs. Therefore, data from the simulation are all valid for optimization.

In total, there are 45 Junction Nodes which are valid for Water Contamination Detection. These Junction Nodes are the basis of our experiments in Chapter 3 and Chapter 4. Full data table are listed in Table 2.1 and Table 2.2.

Table 2.1: Full Result Table from Simulation Part1

Junction Nodes	Frequency(Coverage)	Detection Time(s)
26	8	3225
64	7	1056
118	7	6900
8	6	2500
91	6	6350
80	5	10800
101	5	4440
106	5	1380
112	5	9360
21	4	2475
45	4	7800
73	4	16575
76	4	41775
110	4	750
123	4	10650
130	4	3900
131	4	1600
36	3	62100
48	3	3500
84	3	1900
85	3	12700
10	2	87900

Table 2.2: Full Result Table from Simulation Part2

Junction Nodes	Frequency(Coverage)	Detection Time(s)
13	2	37950
39	2	10800
52	2	1500
66	2	7950
72	2	7350
93	2	1800
14	1	300
16	1	300
37	1	300
38	1	300
41	1	300
42	1	300
50	1	300
74	1	300
82	1	300
83	1	300
99	1	300
100	1	300
114	1	300
124	1	300
125	1	300
126	1	300
129	1	300

Chapter 3

Sensor Placement Evaluation

In this chapter, the aim is to provide a better optimization method for Water Distribution System Sensors Placement through Genetic Algorithm modification. A designed experiment and its result are discussed in this chapter to prove the improvement.

3.1 Genetic Algorithm

Algorithm 1 Genetic Algorithm

Input: Initial population candidates
Output: Desired Converged Offspring

```
Generate the initial population
Compute fitness
REPEAT
Selection
Crossover
Mutation
Compute fitness
UNTIL population has converged; STOP
```

A genetic algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction to produce offspring of the next generation.

There are five phases in a genetic algorithm, starting with an initial population, fitness function, selection, crossover, mutation. The algorithm terminates if the population has converged (does not produce offspring which are significantly different from the previous generation) [31].

3.1.1 NSGA II

Algorithm 2 NSGA II

Input: Pool of candidates to select

Output: Desired Converged Offspring to solve $f(x)$

```
Generate the initial population
Evaluate Objective Values
Assign Rank(level) based on Pareto-sort
Generate Child Population
Binary Tournament Selection; Recombination and Mutation
for i=1 to g do
    for EACH Parent and Child in Population do
        Assign Rank based on Pareto-sort;
        Generate sets of nondominated solutions
        Determine Crowding distance
        Loop by adding solution to next generation strating from the first front until N individuals
    end for
end for
Select points on the lower front with high crowing distance
Create next generation to and back to loop
```

NSGA II is one of the important algorithms that implemented in Multiobjective Decision Analysis. The basic loop of NSGA II [32] is given by Algorithm 2.

After initialisation of a population, it goes on with a repeated generational loop. The first part of the loop is variation where offspring are generated. For each offspring, two parents are selected. Each one of them is selected using binary tournament selection, we selecting better one out of two random individuals by selecting better one regarding its ranking. Parents are then recombined using a standard recombination operator. For real-valued problems simulated binary crossover is used before the mutation process. Afterwards, new offspring populations and parents are merged into a new population set.

The second part of the iteration is a selection part where a multi-objective ranking is selected. In the ranking procedure of NSGA II, first is non-dominated sorting which produced the Pareto Front while the second is comparing crowding distance. An estimate of the density of solutions surrounding that solution. The crowding distance value of a particular solution is the average distance of its two neighbouring solutions, the crowding distance of both candidates on the edge of the scale is marked as infinite. In the picture below, it is referring to y(1) and y(5) [34].

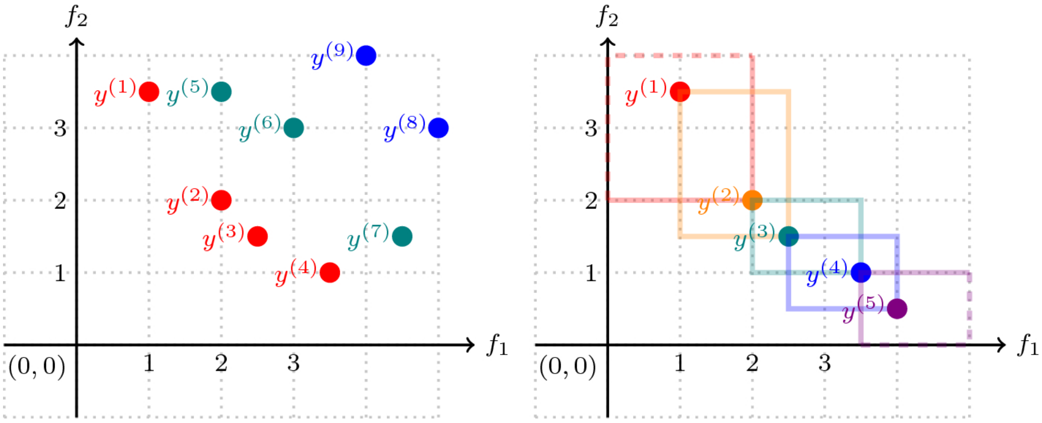


Figure 3.1: Crowding Distance

Parameters Settings

Below we discuss the parameters setting and motivation for our implementation of NSGA II and corresponding improved algorithms.

1. Population Size, a smaller population size is believed to give you quicker convergence speed but the algorithm might be more easily get trapped in local optima. The reverse thing applies to a large population size. Therefore, after comparisons, we select two times of the genes as our population size which is 90.

2. Crossover Rate, the general principle for selection is based on question size. For smaller candidates pool(i.e. five or forty nodes combination) we use 20% of crossover rate while for bigger candidates pool(i.e. twenty nodes combination) we use 30% of crossover rate.

3. Mutation Rate, the general principle is to keep Mutation Rate relatively low otherwise convergence may be delayed unnecessarily.The selected Mutation Rate is 5%.

4. Generation Number, the generation number is decided by the number of possible candidate pool. We divide the possible candidates pool to its population size, however, certain modification is needed after viewing the output.

3.1.2 Limitations of Selected Genetic Algorithms

One of the important limitations of Genetic Algorithms is its computational cost to provide a sound solution. In the network (referencing network), if we are using 20 sensors, there are

$$C(45, 20) = 3169870830126$$

possible combinations as our candidates pool. The network we are using is a small network compared to another complex network. Therefore implementation of NSGA II is insufficient to provide an optimal solution.

3.2 Improvements

Given the limitation, to provide a better convergence in a shorter period, we suggest the enhancement in initialisation and mutation operator process where we can reach the better result quicker.

3.2.1 Initiation Enhancement – Seeds Approach

Algorithm 3 Initiation Approach

Input: Previous Optimal Solutions, Full Candidate List
Output: Population before Objective Evaluation

```
Initialize the population – Size M:
Generate population from previous Pareto front – size N
Random select M-N out of all candidates if not in size N selection
```

To investigate how many nodes are needed for water contamination detection, we iterate with size 1,3,5,10,15,20,30,40,45 nodes to give a general overview of different nodes performance (A detailed explanation in Chapter 4). Therefore, to fully utilise the optimal results from previous experiments, we use the previous Pareto front to replace the "random" initialised population for the new experiment. For instance, for five nodes optimisation process, three nodes' Pareto front is used to generate the initial population for later selection where the two nodes left are subject to random selection.

The motivation behind this approach is the optimal solution from previous experience is a proven record over good performance in both designed objectives functions, adding new nodes would improve its performance on the objective function which is related to coverage, perhaps even improve its performance on average time detection.

3.2.2 Mutation Operator Improvement

Eigen Centrality

Algorithm 4 Eigen Centrality Value

Input: a diagonalizable matrix A

Output: a scalar number h, which is the greatest eigenvalue of A, and a nonzero vector v, the corresponding eigenvector of h, such that $Av = hv$

```
Initialisation: initialise a vector $b_0$, which may be an approximation to
  the dominant eigenvector or a random vector, and let $k = 0$
for k is smaller than the maximum iteration do
    calculate $bk + 1 = A * bk/(|A * bk|)$
    set $k = k + 1$
end for
END
```

In graph theory, eigenvector centrality (also called eigen-centrality) is a measure of the influence of a node in a network. Relative scores are assigned to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. A high eigenvector score means that a node is connected to many nodes which themselves have high scores [35].

Let $A = (a_{i,j})$ be the adjacency matrix of a graph. The eigenvector centrality x_i of node i is given by:

$$x_i = \frac{1}{\lambda} \sum_k a_{k,i} \, x_k$$

where $\lambda \neq 0$ is a constant. In matrix form we have:

$$\lambda x = xA$$

Hence the centrality vector x is the left-hand eigenvector of the adjacency matrix A associated with the eigenvalue λ. It is wise to choose λ as the largest eigenvalue in absolute value of matrix A. By virtue of Perron-Frobenius theorem, this choice guarantees the following desirable property: if matrix A is irreducible, or equivalently if the graph is (strongly) connected, then the eigenvector solution x is both unique and positive.

The power method can be used to solve the eigenvector centrality problem. Let $m(v)$ denote the signed component of maximal magnitude of vector v. If there is more than one maximal component, let $m(v)$ be the first one. For instance, $m(-3, 3, 2) = -3$. Let $x^{(0)}$ be an arbitrary vector. [35]For $k \geq 1$:

1. repeatedly compute $x^{(k)} = x^{(k-1)}A$;
2. normalize $x^{(k)} = x^{(k)}/m(x^{(k)})$;

until the desired precision is achieved. It follows that $x^{(k)}$ converges to the dominant eigenvector of A and $m(x^{(k)})$ converges to the dominant eigenvalue of A. If matrix A is sparse, each vector-matrix product can be performed in linear time in the size of the graph.

The method converges when the dominant (largest) and the sub-dominant (second largest) eigenvalues of A, respectively denoted by λ_1 and λ_2, are separated, that is they are different in absolute value, hence when $|\lambda_1| > |\lambda_2|$. The rate of convergence is the rate at which $(\lambda_2/\lambda_1)^k$ goes to 0. Hence, if the sub-dominant eigenvalue is small compared to the dominant one, then the method quickly converges.

The Mutation Operator Improvement

Eigen Centrality is a clear indication of each potential sensor candidates importance in the network. Therefore, when the mutation process in Genetic Algorithm happens, we assign different weights for each potential sensor candidate according to their eigen-centrality value. The sensor candidates with higher weights are more likely to be selected in the mutation process.

This code substitute the random mutation in the original mutation algorithm which helps to give more probability(weight) to the nodes with higher centrality value thus achieve the goal of incorporating centrality in mutation .

Algorithm 5 Eigen Centrality Mutation

Input: Child Population After Binary Tournament Selection

Output: Child Population Before Ranking Method before next iteration

```
Initialize with sensors candidates and its corresponding centrality value
Start indication = 0
Index indication = 0
Random select a number between 0 and centrality value
Binary Tournament Selection; Recombination and Mutation
for Enumerate centrality value with both indication number  do
    Start += Centrality Value
    If random selected number ¡ = start: break from loop
end for
Return index indication to represent sensor
```

Algorithm 6 Fast Non dominated Sort

Input:All possible solutions in the candidate pool

Output: Non-dominated Solution Set

```
Sort all the solutions ($P_1 \dots P_N$) in decreasing order of their first objective function ($F1$) and create a sorted list ($O$)
Initialise a set $S_1$ and add the first element of list $O$ to $S_1$
for every solution $O_i$ (other than the first solution ) of list $O$ , compare solution $O_i$ from the solutions of $S_1..i$ do
    If any element of set $S_1$ dominate $O_i$; Delete $O_i$ from the list
    If $O_i$; dominate any solution of the set $S_1$; update set $S_1 = S_1 U O_i$
    If set $S_1$ becomes empty; add immediate solution at an immediate solution to $S_1$
end for
Print non dominated set S1
```

3.3 Experiment Design

There are three sets of experiment group. The $ControlGroup1$ is the optimal solution from the entire set of candidates; the $ControlGroup2$ is the solution deriving from original NSGA II; the last which is the $ExperimentGroup$ is the solution deriving from the improved algorithms mentioned above.

When considering computational cost, three sensors combination is where the experiments based upon. Since

$$C(45, 3) = 14190$$

is a relative smaller set as compared to the rest experiments(i.e. $C(45, 20) = 3169870830126$).

For $ControlGroup1$, after generating the full list of three-nodes combination of sensor candidates, we run through a fast non-dominated algorithm to compute inspired by the NSGA II(Algorithm 6).

We iterate from the entire set of three nodes candidates to produce the non-dominant set as the optimal solution. The solution of $ControlGroup2$ is running against original NSGA II Algorithm.

For $ExperimentGroup$, it initialise the population with the Pareto front from one nodes selection which is calculated from non dominated solution mentioned in from all sensors candidates. Therefore, before the selection, we compute the Pareto Front of One sensor by applying Algorithm 6. The final output of Experiment Group is running against the improved algoritm purposed in this Chapter.

* represents that Node 14 16 37 38 41 42 50 74 82 83 99 100 114 124 125 126 129 can replace each other

Table 3.1: One Node Optimization Solution

Junction Nodes	Coverage Index	Detection Time
26	0.125	3225
64	0.142857	1056
110	0.25	750
14*	1	300

3.4 Experiment Result and Conclusion

In Fig 3.2, the Coverage is reffering to F2 in Chapter 2. However, in order to visualize it, we turned the objective from maximizing to minimizing by ap-

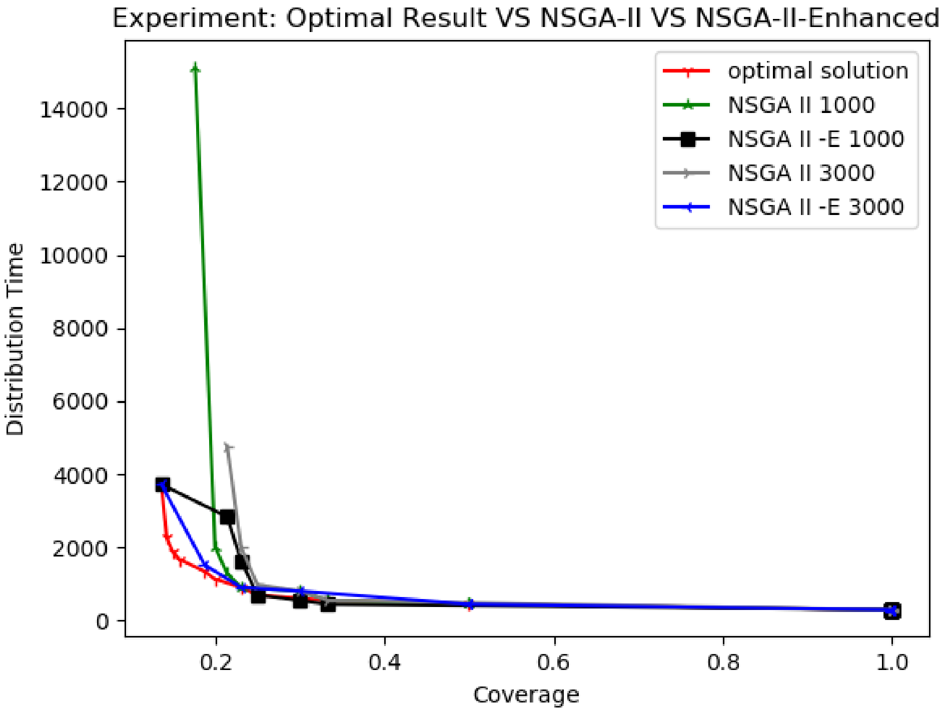


Figure 3.2: Experiment Result from Control Groups and Experiment Group

plying 1/F2. While Detectio Time is calculated in seconds and it is referring to F1.

Observing from Fig 3.2, the $ControlGroup2$(NSGAII) does not compute the optimal solution as compared to the $ControlGroup1$ while the $ExperimentGroup$ provides a better approximation over the $ControlGroup1$. In particular, there is a huge gap between the nodes selection regarding average detection time.

Even if after longer iterations, both algorithms improve itself by providing better candidates, however, the $ExperimentGroup$ still provide closer approximation towards the $ControlGroup2$ as compared to the $ControlGroup1$.

Experiment result proves the ability of the purposed algorithm for providing better optimal solutions for sensors placement.

Chapter 4

Network Sensors Placement

In this chapter, we use different number of nodes as our experiment group to evaluate their performance in Water Contamination Detection. Meanwhile, except for existing controllability analysis, we additionally prove that approximately 30% to 40% of nodes are sufficient for the purpose of Water Contamination Detection.

4.1 Motivation

Water Contamination Detection sensors capability has been developed throughout these years. The sensor can measure the physiochemical parameters of water quality, such as flow, temperature, pH, conductivity, and oxidation-reduction potential. These physiochemical parameters are used to detect water contaminants. The sensors, which are designed from first principles and implemented with signal conditioning circuits, are connected to a microcontroller-based measuring node, which processes and analyses the data. The sensor is capable of reading physiochemical parameters, and can successfully process, transmit, and display the readings [36].

The increasingly sophisticated capability of sensors determines its costly price. If the deployment of contamination detection sensors reaches a larger scope (i.e. a more complex network), it is not sufficient to put sensors in every possible location. Moreover, if the detection time and consumption of contaminated water exceed a certain threshold or important undetected event might cause a huge impact, it is important to maximise the utility of sensors location. Therefore, sensors number shall be another important objective for optimization methods.

In BWSN, a fixed number of sensors are imposed. How a different number of sensors performance would influence the result of designed objectives is hardly ever being discussed.

4.2 Existing Research

Scholars provide different assumptions over how many nodes are needed in water contamination events. According to controllability analysis [1], it is estimated that in water distribution network, using a comparatively small amount of nodes is sufficient in the case study networks (between 15 to 20 nodes). Meanwhile, during the competition of BWSN, the network we are using has a fixed number of five sensors for optimisation.

4.2.1 Controllability analysis overview

Computational cost is a commonly stressed problem for optimisation method; online monitoring enjoys a widely acknowledged reputation regarding its ability to protect against the impact of contamination intrusions. This method suffers the same problem in computational cost. Therefore, in order to decrease the relevant computational cost, a pre-selection method based on Controllability is purposed.

Controllability

According to the control theory, a linear time-invariant system whose states are determined by the following equation:

$$\frac{dx(t)}{d_t} = Ax(t) + Bu(t)$$

Where the vector$\mathbf{X}(t) = (x_1(t), \cdots, x_N(t))^{\mathrm{T}}$, denotes the state of N nodes in the network at time t, A is the transpose of the adjacency matrix of the network, B is the input matrix that defines how control signals are inputted to the network, and $\mathbf{u}(t) = (u_1(t), \cdots, u_M(t))^{\mathrm{T}}$ represents the H input signals at time t. A node whose control signal is directly inputted is called a driver node. The minimum sets of driver nodes to control a network are called the minimum driver nodes sets (MDSs).

Controllability Application in network

The controllability of complex water distribution systems is explored by applying the linear equation listed above to help understand the real non-linear systems.

Controllability is an extremely efficient method which has a very low computational cost $O(N^{1/2}L)$ at most, where N and L denote the number of nodes and links respectively. The computational cost made it feasible to explore the large and complex network.

The pre-selection method successfully decreases the number of decision variables for approximately 30 to 40 percent.

4.3 Experiments and Results

4.3.1 Experiments Design

Since there is a lack of sensors number guidance, we purposed a clear comparison over a different number of sensors capability in dealing with two objective functions purposed in Chapter 3.

The total number of sensor candidate is 45 which is selected by full iteration of simulated contamination event purposed in Chapter 2. Therefore, this experiment needs to work through nine experiments with the improved algorithms. The number of sensors suggested for each experiment is: 1,3,5,10,15,20,30,40,45.

The motivation behind the selection is based on several criterions.

1. We select the sensors on its minimal and maximal scale to indicate the boundary of the sensor(s) performance.

2. Ten nodes difference is serving as the gap between each experiment.

3. According to BWSN, the network used in the thesis requires five sensors. However, according to controllability analysis, sensors number shall be in between 15 to 20. Therefore, 5, 15, 20 are also selected.

4.3.2 Hypervolume indicator

How to compare Pareto sets lies at the heart of research in multi-objective optimization. A measure that has been the subject of much recent study in evolutionary multi-objective optimization is the hypervolume indicator. It measures the volume of the dominated portion of the objective space and is of exceptional interest as it possesses the highly desirable feature of strict Pareto compliance [33].

We compare the optimal approximation factor with the approximation factor achieved by sets maximizing the hypervolume indicator. Therefore the larger the hypervolume index, the more optimal result we can get.

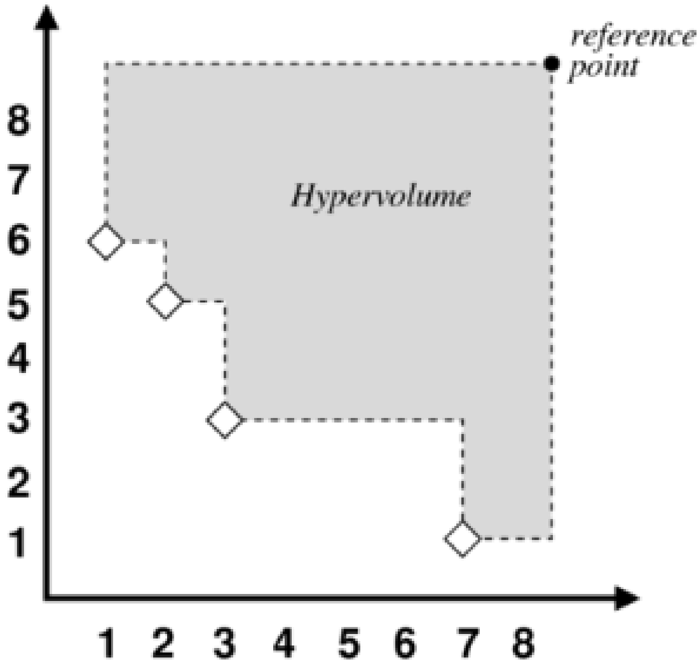


Figure 4.1: Hypervolume index example

Since Genetic Algorithm would not always compute the same result, after ten experiments, we select the Pareto Front with the maximal hypervolume indicator in each experiment separately.

4.4 Result Analysis and Conclusion

4.4.1 Algorithm Performance

For each experiment, we use the same reference point in order to directly compare the hypervolume index and run ten times to detect if our algorithm is consistent and reliable.

Node 1, 3, 45 is calculating from full list candidates(45 is exactly full list candidates) which produce the optimal results while the rest is running against the improved algorithm, therefore 1,3,45 would not be compared through Hypervolume index boxplot.

As we can from the boxplot, 15 sensors and 20 sensors provide a high hypovervolume index as compared to the rest. Starting from 30 sensors, the hypervolume index decreas due to its significantly longer average time

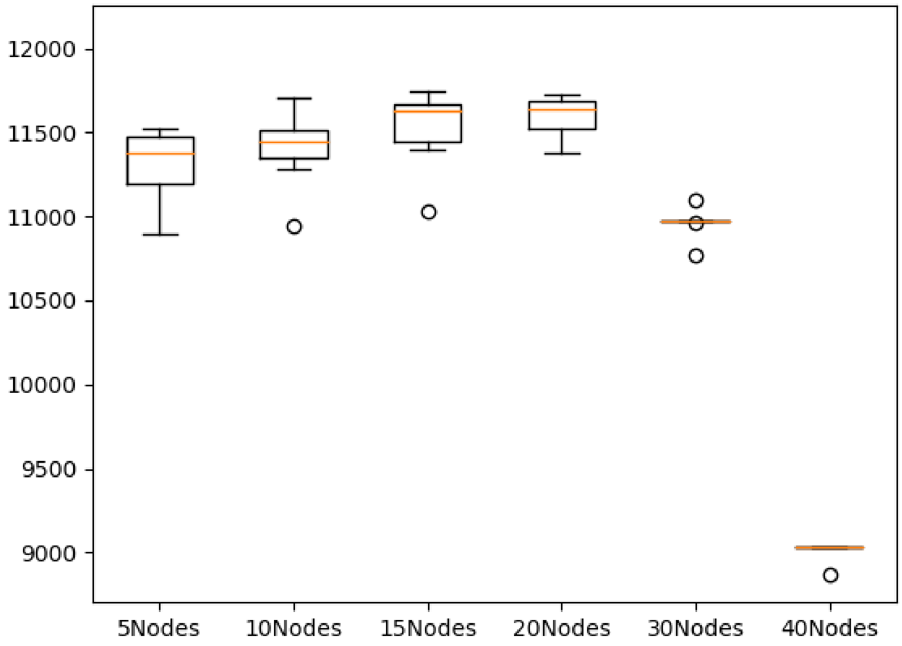


Figure 4.2: Boxplot: Hypervolume index for each experiments

detection in F2 while improves relatively little in coverage index (F1). As a result, it proves that approximately 30% to 40% nodes are sufficient for sensors placement in Water Contamination Detection.

In order to show that our algorithm can produce a consistent performance. We use Standard Deviation/ Average as an index to reflect if the hypervolume index is consistent.

Table 4.1: Algorithm Performance Comparison

Experiment Group	Standard Deviation/Average
5 nodes	0.01855
10 nodes	0.01797
15 nodes	0.01705
20 nodes	0.01056
30 nodes	0.00690
40 nodes	0.00601

4.4.2 Result Analysis and Conclusion

From the result collection Fig 4.3, all the Pareto front is selected based on hypervolume index in each experiment.

We can see the coverage continually improve when deploying more nodes. the distribution time values fluctuates in different nodes set.

We can observe starting from 15 nodes to 20 nodes, the selected network almost cover the entire network even with low distribution time. With more nodes deployed starting from 20 nodes, the Pareto Front performance does not improve sufficiently regarding coverage while the distribution time fluctuates.

Therefore, together with the finding in boxplot, it proves that approximately 30% to 40% nodes are sufficient for approximating network.

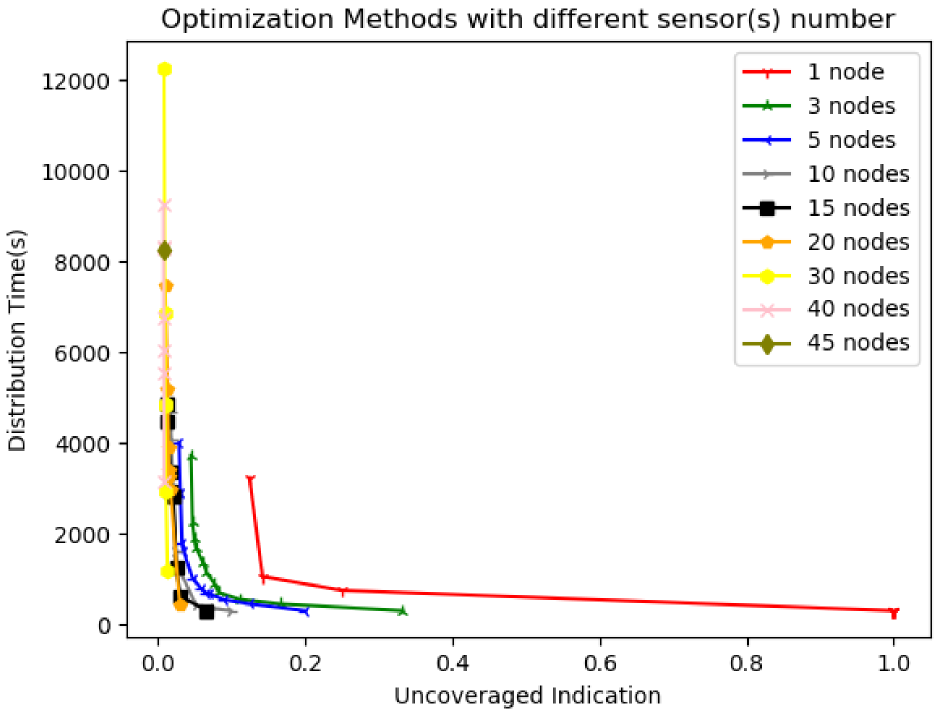


Figure 4.3: Pareto Front for different sensors number(s) (Sensor numbers:1,3,5,10,15,20,30,40,45)

4.5 Result Tables For Each Experiment

Full result Tables are in the Appendix, here attached the Pareto Front picture of each experiment.

1. Coverage Index is reversed by 1/SumOfCoverage to turn from maximize objective function to minimize objective function

2. Detection Time is calculating in second(s)

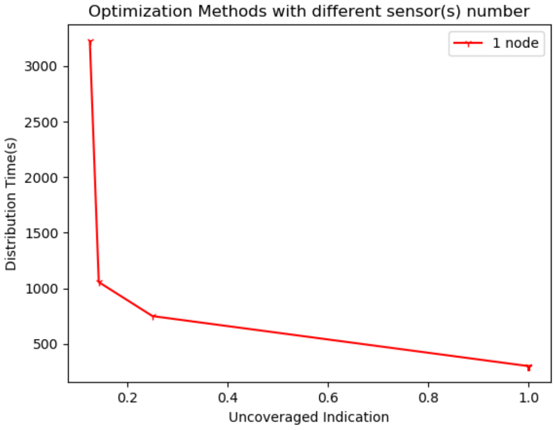


Figure 4.4: Pareto Front for different sensors number(s) (Sensor numbers:1)

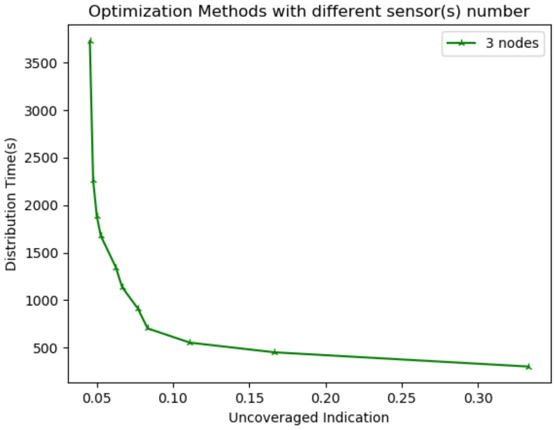


Figure 4.5: Pareto Front for different sensors number(s) (Sensor numbers:3)

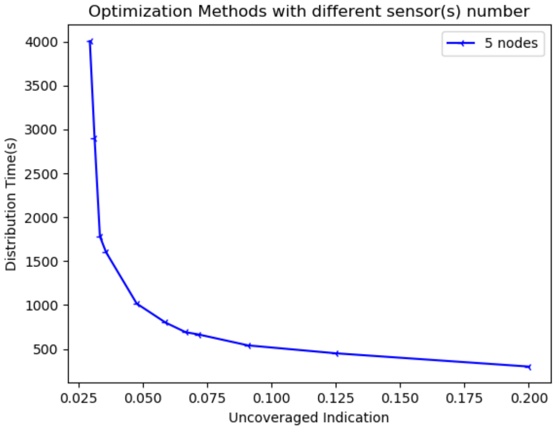


Figure 4.6: Pareto Front for different sensors number(s) (Sensor numbers:5)

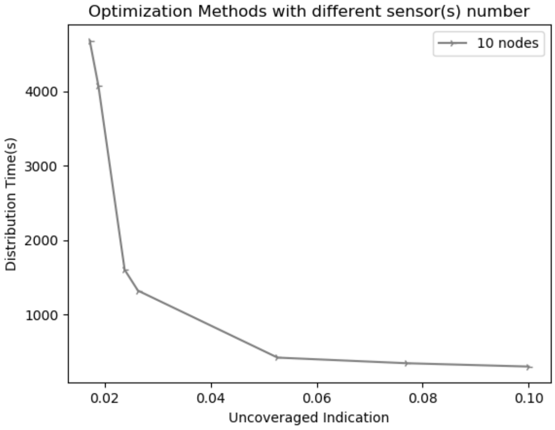


Figure 4.7: Pareto Front for different sensors number(s) (Sensor numbers:10)

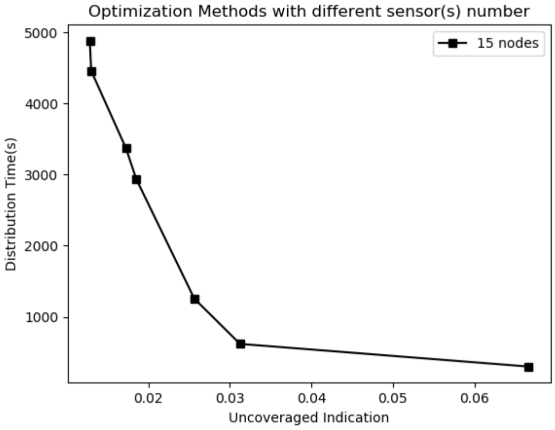


Figure 4.8: Pareto Front for different sensors number(s) (Sensor numbers:15)

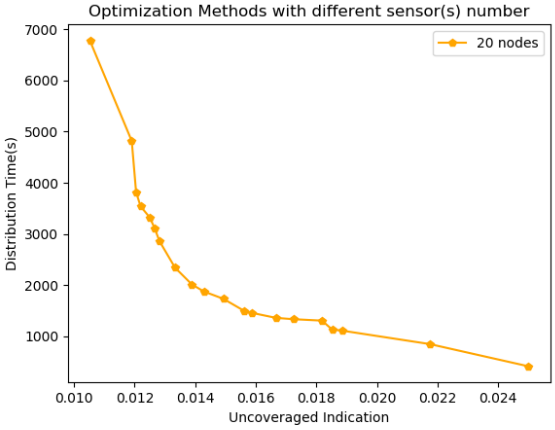


Figure 4.9: Pareto Front for different sensors number(s) (Sensor numbers:20)

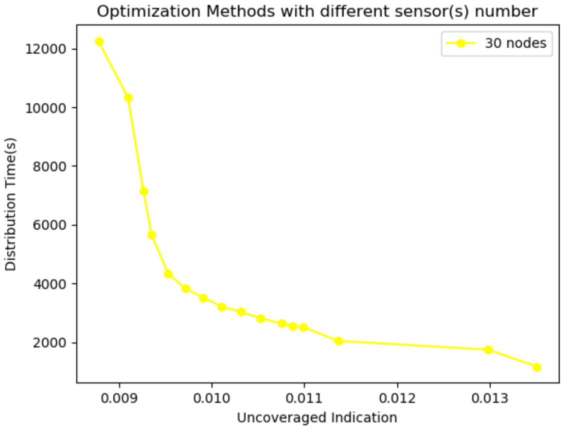


Figure 4.10: Pareto Front for different sensors number(s) (Sensor numbers:30)

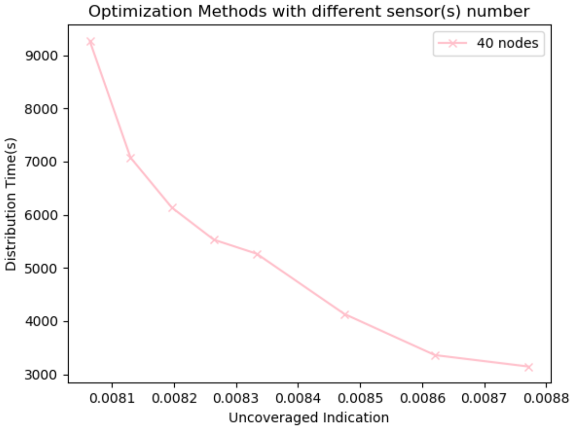


Figure 4.11: Pareto Front for different sensors number(s) (Sensor numbers:40)

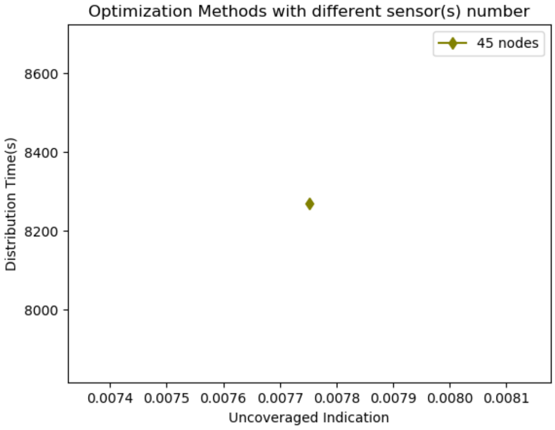


Figure 4.12: Pareto Front for different sensors number(s) (Sensor numbers:45)

Chapter 5

Discussion and Future Work

5.1 Case study on different network

At the moment, the network selected from this paper is relatively small. Realistically, a network for the city is more complex. A more comprehensive network would help us better check the capability in enhanced algorithm performance and purposed assumptions.

5.2 Sensors number as a new objective

Different experiments are running against multiple sets of sensors, but Sensors number is not calculated as an objective. Therefore, a new objective function related to sensors number shall be added.

5.3 Incorporates Risk Analysis

A sensor design currently is running against expected value by simulation process. However, there are multiple different sources of risk in a realistic world might not be covered in the simulation process. However, to provide a better simulation, risk analysis shall be considered.

5.4 Sensor Reliability

In reality, the correct functioning of sensors is not guaranteed. Considering both false positive and negative rates would serve as a starting point for checking the reliability of sensors. The challenge remains for incorporating the reliability as part of the sensor design process.

Bibliography

[1] Kegong Diao*, Wolfgang Rauch, Controllability analysis as a preselection method for sensor placement in water distribution systems, 6087-6108, 2013.

[2] Charlie Frago, https://www.tampabay.com/news/environment/water/st-petersburgs-black-cloud-of-sewage-woes-grows-by-58-million-gallons/2293625. Retrieved Sep 2018.

[3] Hickman, D.C., A chemical and bilogical warefare threat: USAF water systems at risk. Counterproliferation Paper No.3, USAF Counterproliferation Center, Maxwell Air Force Base, Ala, 1999.

[4] Deininger, R. A. and Meier, P.G., Sabtoage of public water supply system. Security of public water supply, Vol.66, 2000.

[5] https://ejatlas.org/conflict/hinkley-groundwater-contamination. Retrieved Sep 2018.

[6] American Water Works Association(AWWA), Contamination Warning Systems for Water: an Approach for Providing Actionable Information on Decision-makers, 2005.

[7] Megan Janke PH.D., Adam D. ,Douglas K., Modeling Change in Older Adults' Leisure Activities, 2007.

[8] ES B., MA Palmer, JD Allan, G Alexander, Synthesizing US river restoration efforts, 2005.

[9] Murray R., Uber J. and Janke R., Model for estimating acute health impacts from consumption of contaminated drinking water, 2006.

[10] Storey et al, Double layer in ionic liquids: Overscreening versus crowding. 2011

[11] Berry, J. Hart, W.E., Philips, C.A., Uber, J.G, Walski, T.M., Water quality sensor placement in water networks with budget constraints. In: Proc., World Water and Environment Resources Planning and Management 132(4), 218-224, 2005

[12] A Preis, M Propato, O Piller, GB Trachtman, The battle of the water sensor networks (BWSN): A design challenge for engineers and algorithms, 2008.

[13] Bahadur, R., Samuels, W.B., Gratman, W., Amstuz, D., Pickus, J., 2003. PipelineNet: a model for monitoring introduced contaminant sin a distribution system. In: Proc., World Water and Environmental Resources Congress and Related Symp. ASCE, Reston, Va. 2003

[14] SR Ghimire, BD Barkdoll, Heuristic method for the battle of the water network sensors: Demand based approach, 2006

[15] Z Xu, RR Yager,Dynamic intuitionistic fuzzy multi-attribute decision making, 2008

[16] DP Solomatine, A Ostfeld, Data-driven modelling: some past experiences and new approaches, 2008

[17] BH Lee, RA Deininger, Optimal locations of monitoring stations in water distribution system, 1992

[18] M Propato, Contamination warning in water networks: General mixed-integer linear models for sensor location design, 2006

[19] Dorni, C.A., Vidyalakshmi, K.S. Vasanthi, H.R, Rajamanickam, G.V and Dubey,G.P. Anti-inflammatory activity of Plumbago capensis. Pharmacognosy magazine, 2(8):239-243. 2006

[20] A Kessler, A Ostfeld, G Sinai, Detecting accidental contaminations in municipal water networks, 1998

[21] A Ostfeld, E Salomons, Optimal layout of early warning detection stations for water distribution systems security, 2004.

[22] WE Hart, R Murray, Review of sensor placement strategies for contamination warning systems in drinking water distribution systems, 2010

[23] Clark R.M. and Haught R.C. (2005). Characterising pipe wall demand: implications for water quality modelling. J. Water Resources Planning and Management, 31(3), 208-217.

[24] https://www.anylogic.com/use-of-simulation/. Retrieved Sep 2018.

[25] Rossman L.A., EPANET 2 users manual, Water Supply and Water Resources Division, National Risk Management Research Laboratory, Office of Research and Development, U.S. Environmental Protection Agency, Cincinnati, Ohio, 2000

[26] Shang F., Uber J.G. and Rossman L.A., Modelling reaction and transport of multiple species in water distribution systems. Journal of Environmental Science and Technology, 42(3), 808-814, 2008

[27] https://www.epa.gov/water-research/epanet. Retrieved Sep 2018.

[28] A Ostfeld, JG Uber, E Salomons, JW Berry, The battle of the water sensor networks (BWSN): A design challenge for engineers and algorithms, 2008

[29] M Propato, O Piller, Battle of the water sensor networks, 2008

[30] https://sites.google.com/site/waterdistributionusingdream/about-epanet. Retrieved Sep 2018.

[31] J. H. Holland. Adaptation in natural and articial systems. The University of Michigan Press, Ann Arbor, MI, 1975.

[32] Deb K, Pratap A, Agarwal S, Meyarivan T, A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197', 2002

[33] K. Bringmann, T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. Computational Geometry: Theory and Applications, Vol. 43, pages 601-610. 2010.

[34] Michael T. M. Emmerich, André H. Deutz, A tutorial on multiobjective optimization: fundamentals and evolutionary methods, 2018

[35] Michael Berry, Murray Browne, Understanding Search Engines: Mathematical Modeling and Text Retrieval. Second Edition, SIAM, Philadelphia. 2005.

[36] Niel Andre Cloete, Reza Malekian, Lakshmi Nair,Design of Smart Sensors for Real-Time Water Quality Monitoring, 2015

Appendix A

Full iteration results

Notes:

1. * represents that Node 14 16 37 38 41 42 50 74 82 83 99 100 114 124 125 126 129 can replace each other
2. Coverage Index is reversed by 1/SumOfCoverage to turn from maximize objective function to minimize objective function
3. Detection Time is calculating in second(s)
4. Node 1, 3, 45 is calculating from full list candidates which produce the optimal results while the rest is running against the improved algorithm (45 nodes are exactly full list of candidates, therefore not shown in the table)

Table A.1: One Node Optimization Solution

Junction Nodes	Coverage Index	Detection Time
26	0.125	3225
64	0.142857	1056
110	0.25	750
14*	1	300

Table A.2: Three Nodes Optimization Solution

Junction Nodes	Coverage Index	Detection Time
26,64,118	0.045454545	3727.0
26,64,8	0.04761905	2260.33333333
26,64,106	0.05	1887.0
26,64,110	0.05263158	1677.0
64,106,131	0.0625	1345.33333333
64,110,131	0.06666667	1135.33333333
64,106,14*	0.07692308	912.0
64,110,14*	0.0833333	702.0
64,14*,16*	0.111111	552.0
110,14*,16*	0.166666667	450.0
14*,16*,37*	0.33333	300.0

Table A.3: Five Nodes Optimization Solution

Junction Nodes	Coverage Index	Detection Time
26,64,118,8,91	0.029412	4006.2
26,64,8,91,106	0.03125	2902.28
26,64,106,8,110	0.033333	1782.2
26,64,110,131,106	0.03571429	1602.2
64,106,131,110,13*	0.04761905	1017.8
64,110,131,10*,13*	0.05882352	801.2
64,106,14*,10*,13*	0.06666667	667.2
64,110,14*,10*,13*	0.09090909	451.2
110,14*,16*,10*,13*	0.125	330.0
14*/16*/37*/10*/13*	0.2	300.0

Table A.4: Ten Nodes Optimization Solution

Junction Nodes	Coverage Index	Detection Time
64,125*,124*,114*,100*,14*,99*,41*,10*,16*	0.052	420.6
110,125*,124*,114*,100*,14*,99*,41*,10*,16*	0.077	345
26,64,106,8,110,118,91,101,112,80	0.017	4676.1
129*,125*,124*,114*,100*,14*,99*,41*,10*,16*	0.1	300
26,64,106,8,110,131,52,84,93,14*	0.024	1601.1
26,64,118,8,110,14*,52,84,10*,16*	0.026	1321.1
26,64,118,8,14*,106,101,112,80,110	0.0189	4071.1

Table A.5: Fifteen Nodes Optimization Solution

Junction Nodes	Coverage Index	Detection Time
110,64,(129,125,124,114,100,14,99,41,10,16,83,82,74,42)*	0.031	619.067
26,64,118,8,91,106,101,112,80,110,131,21,130,45,123	0.013	4879.067
(129,125,124,114,100,14,99,41,10,16,83,82,74,50,42)*	0.0677	300.0
42*,45*,84,52,21,106,8,112,118,100*,131,124*,82*,39,26	0.0185	3376.0
84,50*,52,125*,80,26,16*,82*,64,91,106,93,74*,130,123	0.0172	2937.4
8,73,118,106,52,76,38*,45,110,91,39,83*,84,130,101	0.0256	1255.067
26,64,118,8,91,106,101,112,84,110,131,21,130,45,123	0.013	4460.73
110,(129,125,124,114,100,14,99,41,10,16,83,82,74,50)*	0.0556	330.0
64,(129,125,124,114,100,14,99,41,10,16,83,82,74,50)*	0.0476	350.4

Table A.6: Twenty Nodes Optimization Solution

Junction Nodes	Coverage Index	Detection Time
37*,41*,114*,52,26,84,106,64,80,85	0.017543859649122806	2981.8
73,125*,8,48,93,129*,100*,74*,14*,82*		
124*,112,8,101,73,84,42*,91,66,21	0.014925373134328358	3905.3
131,48,45,72,93,52,50,64,83,110		
26,85,110,21,91,126*,100*,37*,72	0.015384615384615385	3411.8
93,42*,38*,123,106,84,8,64,50,39,48		
82*,129,85,106,126*,26,48,123,101	0.013513513513513514	5189.3
80,64,72,112,100*,131,8,73,91,38*,39		
74*,91,83*,112,72,129*,131,85,84,110	0.01639344262295082	3239.05
93,101,130,52,37*,82*,64,21,45,126*		
14,16,37,38,41,42,50,74,82,83,106	0.03030303	414.3
99,100,114,124,125,126,129,110,64		
26,64,118,8,91,106,101,112,80,110	0.010526316	7481.8
130,45,123, 73,76,84,48,85,21,131		

Table A.7: Thirty Nodes Optimization Solution

Junction Nodes	Coverage Index	Detection Time
(52,84,125,74,16,85,64,26,93,80,106,129,39,48,123	0.01041667	4809.033
110,42,118,130,21,82,112,41,66,37,50,8,76,91,100)*		
(64,72,85,52,126,118,83,106,125,99,130,37,80,131,8	0.01010102	6855.8667
76,36,73,82,38,84,48,123,21,14,26,74,91,114,101)*		
(52,80,83,114,39,74,50,106,14,131,82,26,37,85,118	0.0114942529	2918.3667
126,41,124,38,73,91,64,42,110,129,99,8,101,21,125)*		
14,16,37,38,41,42,50,74,82,83,106,52,131,93,84,21	0.013513514	1170.8667
99,100,114,124,125,126,129,110,64,8,26,48,130,101		
26,64,118,8,91,106,101,112,80,110,36,52,93,72,66	0.00877193	12252.8667
130,45,123, 73,76,84,48,85,21,131,39,13,10,14*,16*		
(131,83,123,73,42,100,74,26,21,48,130,124,125,80,129	0.0093457943	8797.3452
82,84,72,112,110,45,85,8,101,91,118,64,76,39,106)*		

Table A.8: Fourty Nodes Optimization Solution

Junction Nodes	Coverage Index	Detection Time
(74,125,114,85,118,21,130,110,131,64,76,37	0.008064516	9264.65
39,16,123,84,101,93,41,36,72,52,124,66,100,126		
73,106,26,48,45,91,8,129,13,10,80,112,83,50)*		
(99,14,82,38,36,125,8,126,100,16,131,39,124,85	0.008130081	7074.65
52,91,129,101,48,84,45,26,72,110,41,114,21,112		
130,80,123,93,106,76,50,64,37,118,83,74)*		
(101,52,74,125,37,99,50,13,45,26,130,64,118,	0.0081967213	6133.4
110,80,14,66,129,48,84,38,123,112,100,39,91,		
76,83,42,36,126,72,106,16,131,8,114,21,73,93)*		
(41,100,83,38,64,26,131,93,74,125,106,91,48	0.00826446281	5529.65
118,39,50,114,76,85,123,52,42,72,73,126,37,8		
45,80,130,112,21,84,101,66,36,99,10,110,124)*		
(21,26,123,16,110,38,72,14,37,83,64,129,85,45	0.00877193	3144.65
114,101,91,36,48,82,8,76,50,130,118,131,100		
106,73,80,39,125,74,10,84,66,42,112,52,41)*		
(14,37,121,64,129,85,45,21,26,123,16,110,38,72,	0.008620690	3360.275
119,101,91,120,48,82,8,76,50,130,118,131,100		
106,73,40,39,125,42,10,84,66,42,112,52,41)*		
(72,14,37,83,64,129,85,45,21,26,123,16,110,38,	0.00847457627	4134.65
110,80,14,66,129,48,84,38,123,112,100,39,91,		
106,73,80,39,125,74,10,84,66,42,112,52,41)*		

Appendix B

Eigen Centrality Value of Sensor Candidates

Table B.1: Eigen Centrality Value Result Table

Junction Nodes	Centrality
123	0,082226
130	0,301279
131	0,091494
36	0,672393
48	0,184193
84	0,657461
85	0,42535
10	0,328839
13	0
39	0,438871
52	0,415769
66	0,074522
72	0,185071

Table B.2: Eigen Centrality Value Result Table Part2

Junction Nodes	Centrality
99	0,199984
100	0,337818
114	0,419577
124	0,290348
125	0,30949
126	0,445138
129	0
26	0,389607
64	0,074546
118	0,264835
8	0,123821
91	0,144328
80	0,374543
101	0,199984
106	0,232671
112	0,075092
21	0,60618
45	0,082226
73	0,206232
76	0,655913
110	0,025722
93	0,136205
14	0,075469
16	0
37	0,752721
38	0,672393
41	0,198965
42	0,251222
50	0,184193
74	0,34258
82	0,646136
83	1