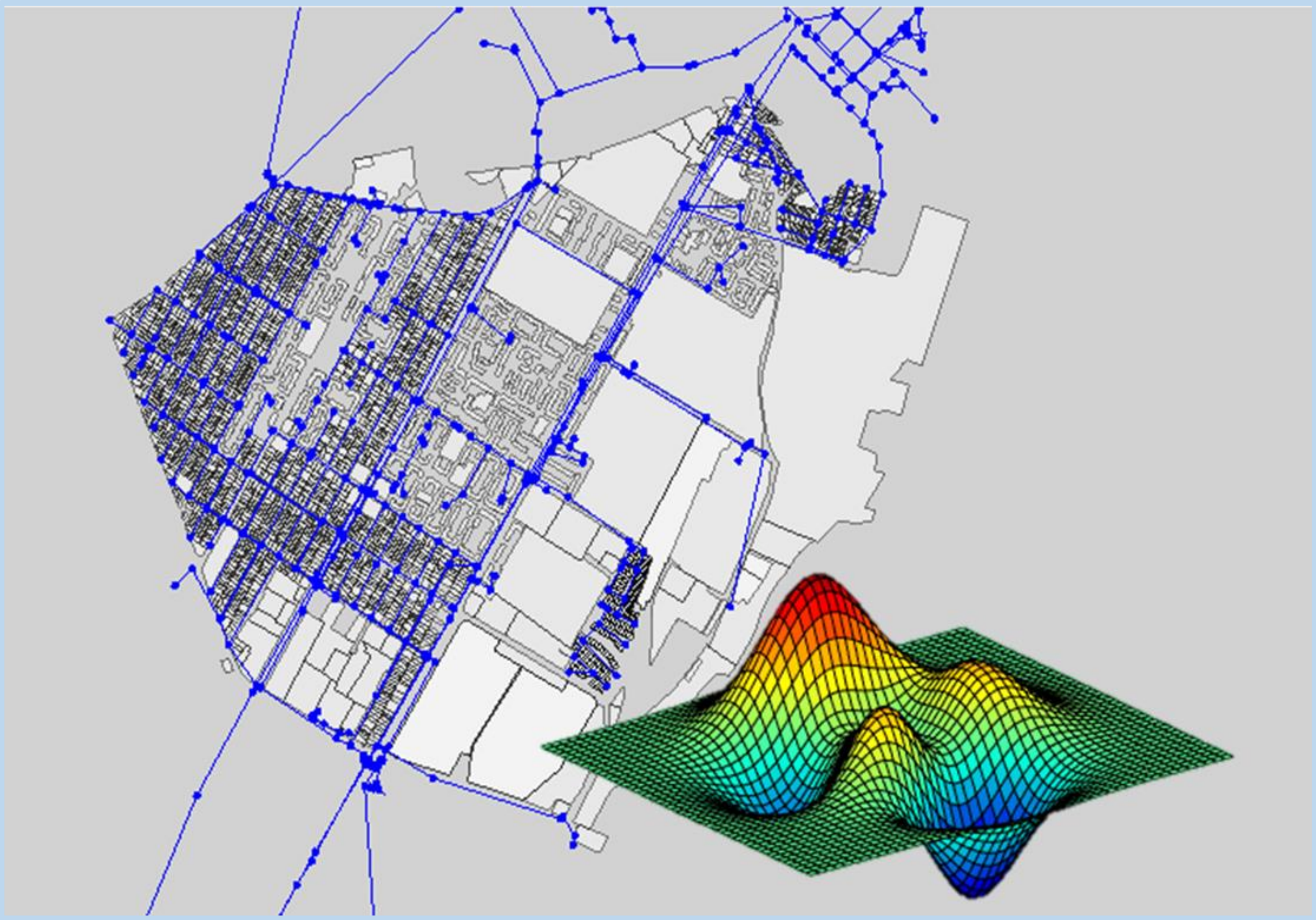
Methodology to Optimally Place Pressure Sensors for Leak Detection in Water Distribution Systems using Value of Information and Entropy.

EA López García

MSc Thesis Identifier WSE-HI-21-06
March 2021

Updated version, April 2021



Methodology to Optimally Place Pressure Sensors for Leak Detection in Water Distribution Systems using Value of Information and Entropy.

Master of Science Thesis
by
EA López García

Supervisors
Prof Dr. Dimitri Solomatine

Mentors
Dr. Leonardo Alfonso

Examination Committee
Prof Dr. Dimitri Solomatine
Dr. Leonardo Alfonso
Dr. Claudia Quintiliani, External Committee Member

This research is done for the partial fulfilment of requirements for the Master of Science degree at the IHE Delft Institute for Water Education, Delft, the Netherlands.

Delft
30/03/2021

This document will be partly/fully reused by the author in scientific papers.

Abstract

The presence of leaks in water distribution networks (WDN) is a very common problem that impacts these systems in a variety of ways, including the loss of drinking water, the deterioration of the quality of supply water due to the intrusion of substances into the system, the structural damage caused by infiltrations, and the uneven distribution of pressures and flows in the network, among others. For this reason, the use of sensors to identify possible where leaks could be located, and to operate the network is imperative. However, placing the sensors in the network is not a trivial problem, due to the complexities of the networks, the uncertainties around them, and the limited resources.

Although many researchers have proposed different methodologies for locating sensors, methods accounting for the costs of the decision-making situation and the nature of the decision-maker are very limited. Moreover, the existing methods largely rely on demand-driven analysis approaches, which limit their applicability.

This thesis presents a methodology to determine the optimal location of pressure sensors for identifying the existence of leaks in water distribution systems. The methodology is based on two unrelated concepts, Value of Information (VOI) and Entropy. The former is common in areas such as medicine and economics but with very limited application in water distribution systems. The latter is a concept from Information Theory, which have been used in the past for locating sensors in the water environment. The proposed methodology, which has a simplified framework with practical interpretation, involves the judgment of decision-makers and switches from the assumption of a demand-driven analysis to a more realistic approach in which the demands are a function of the pressure's system (pressure-driven analysis). Additionally, the methodology considers, to a certain extent, the possible sources of uncertainty and parameter variation along time, towards robust solutions.

The methodology is applied in the case study of the city of Braila, in Romania, which has the goal of reducing water losses caused by leaks by a value of 50 L/h/km. Results show that the sensor locations obtained agree with previous methods in simpler networks, and that well distributed sensors are obtained, although the performance of the method is depending on the selection of the threshold in pressure that is considered critical. A by-product of this research is the Sensor Detection Value (SDV) index, which is based on VOI concepts and that facilitate the application of the methodology and the interpretation of the results.

Acknowledgements

My sincere gratitude to the professors and staff of IHE who supported us in every moment, not only academically but also psychologically in this year of pandemic, to the Dutch government for giving me the opportunity to achieve this dream, and to my mentor Leonardo Alfonso for his valuable advice, who helped me many years ago in the beginning, middle and end of this project.

I also want to highlight the fact of being so fortunate, that despite the difficult conditions we have experienced in these times of pandemic, and witnessing how unfortunately the whole world stopped, I was part of this group of students at IHE who were able to advance professionally, with the culmination of this master's degree.

Friends who have supported me during these last 18 months in the culmination of a great project, without them this would not have been possible.

Finally, and most importantly, I would like to thank my family that from the distance have been all my support. without them this would be a pointless project, briefly I will say that they were my main engine.

Table of Contents

List of Figures

List of Tables

Abbreviations

DDA	Demand-driven analysis
EPA	United States Environmental Protection Agency
IE	Information entropy
IT	Information theory
Lf	Leak flow
MLR	Minimum Leak Required
NSGA	Non-sorted genetic algorithm
PDA	Pressure-driven analysis
SDV	Sensor detection value
VOI	Value of Information
WDS	Water distribution System
WNTR	Water Network Tool for Resilience

List of Symbols

Bd	Base demand
$C_{a,s}$	Consequences of each action.
F_d	Final demand
i	Leak node
j	Sensor node
J	Set of sensor nodes
k	Number of leaks
LD	List of detection
L_f	Leak flow
L_p	Number of nodes where leaks are placed
$\boldsymbol{p}$	Pressures
P_d	Percentage of detection
P_r	Number of nodes where pressures are read
π_s	Prior Belief
So	Initial state of the system
S	Modified state of the system
$q_{m,s}$	Conditional probability
Th	Threshold

Chapter 1 Introduction

1.1 Background

The presence of leaks in a water distribution system (WDS) causes a variety of problems, including the loss of drinking water (Capponi et al., 2017), the deterioration of the quality of supply water due to the intrusion of substances into the system (B. Farley et al., 2009), the structural damage caused by infiltrations (Ponce et al., 2014) and the uneven distribution of pressures and flows in the network. These problems may trigger others, such as the loss of trust by users towards water entities and increased risk of grant accreditations by water regulatory entities due to the system's malfunctioning (Quintiliani & Vertommen, 2020).

For this reason, water utilities regularly work on developing methods that help locate leaks in the least possible time. These methods can be active and/or passive, depending on the strategy used. On the one hand, the active methods include a physical inspection of the pipes, the use of acoustic sensors or gas tracers, ground penetration radar, infra-red imaging, among others (M. Farley & Trow, 2015). On the other hand, the passive methods depend on secondary evidence of leaks, such as checking unexpected changes in the network's pressure values (Raei et al., 2019). A common way to estimate the expected pressure values in a network is by using computer models, which are mathematical representations of the entire network. Furthermore, real pressure values are obtained from measurements in the field, for example, with the help of pressure sensors.

The difference between an expected pressure value and a measured pressure value can warn about leaks' potential existence (Raei et al., 2018). Although installing many sensors in a network could be ideal for leak detection, this is not always possible due to limited resources. Indeed, it is common to have only a few sensors available and is fundamental to optimally decide where the available sensors should be located to maximize their usefulness. Optimal distribution of sensors will help identify the presence of leaks with more extensive network coverage and, combined with methods for finding their specific location, can reduce the time in which leaks are detected, the loss of water, and associated damages (Bohorquez et al., 2020).

Multiple investigations have been developed to build computational frameworks for the optimal location of sensors. B. Farley, Boxall, and Mounce (2009); Pérez et al. (2009); Sanz et al. (2016); Raei et al. (2018); Quintiliani and Vertommen (2020), presented alternatives of sensor deployment using the theory of model-based fault diagnosis, where pressures of the system without leaks area compared with the pressures in the system once it is tested under leak conditions. Analysis of uncertainty were included by Taravatrooy et al., (2020); Jung & Kim, (2017); Steffelbauer & Fuchs-Hanusch, (2016), and the use of novelty concepts as the use of information theory have been explored by Mohammad Sadegh Khorshidi et al., (2020).

Although progress has been made in developing methodologies, there are still gaps that can be addressed. For example, most of them assumed demand-driven approaches (DDA), where all nodes' demand could be supplied regardless of the network's pressure values, this assumption could be distant for reality, because in leak scenarios exist nodes where the pressure drops at a point that demands are reduced or even not supplied, consequently, the solutions obtained in previous studies may not be adequate for the real world (Raei et al., 2019). This research proposes to use a pressure-driven approach where demands are function of the pressure

On the other hand, most of the methodologies based their criteria for sensor selection on the hydraulic behaviour of the system, parameters such as detection times, water loss, number of sensors, number of leaks detected among others were the basis for the choice of the best set of sensors. However, all these methodologies have the objective to generate tools to support the decision-maker to execute a specific measure, but in most of the methodologies the opinion of the decision-maker was not considered. In practice, not all nodes in the system present the same level of risk to leaks and not all leaks have the same impact on system losses, so each of the entities (decision-makers) will know where to locate these points of particular attention and will have their own needs, so, the inclusion of a parameter that considers the decision-maker's perception is required.

For this reason, concepts such as the value of information and principles of information theory such as entropy will be used. The goal is to obtain nodes that has the maximum value for the decision-maker, and simultaneously, that provides the maximum amount of information from the system. The concept of the value of information (VOI) has been used in many disciplines, for example, in the oil and gas sector for the evaluation of operators (Rice, 2014), for groundwater quality monitoring (Hosseini & Kerachian, 2017), for flood monitoring and impact assessment (Alfonso, 2010), for water quality assessment (Shaqadan, 2008) and recently, a first attempt to apply VOI for the optimal location of sensors was made by Mohammad Sadegh Khorshidi et al., (2020) who assumes that each node can be assigned a VOI, and those with the maximum value would be selected as the best candidates to locate a sensor. However, despite using the concept of the value of information, these authors failed to incorporate the perception of the decision-maker.

Apart from selecting the nodes with higher VOI, which is an indicator of the quality of the information obtained, it is also necessary to determine the quantity of information obtained, selecting the best combination of multiple nodes where sensors provide more information or more extensive coverage in detecting pressure changes in the system. To this end, concepts such as information entropy (IE) (Singh 1997) can be applied.

This methodology aims to combine different approaches, considering decision-maker’s judgement, based on VOI and entropy concepts, including analysis of sources of uncertainty evaluating different leaks and threshold values and using a pressure-driven analysis. The methodology is expected to be useful in practice because its results will be applied for the location of pressure sensors in a real distribution network localized in the water distribution systems (WDS) in Braila, Romania, under the framework of the H2020 NAIADES project.

1.2 Research Objective

The main objective of this research is to formulate a methodology to optimise the localization of pressure sensors for leak detection, considering a pressure-driven analysis, sources of uncertainty and using value of information and information theory concepts. This objective will be accomplished by answering the following research questions:

- How to formulate a methodology to optimise the location of pressure sensors, using concepts of value of information and information entropy?
- How can pressure-driven analysis be performed, using the available modelling systems?

According to the previous questions, two sub-questions can be formulated:

- To what extent it is appropriate to use the concept of the value of information as a parameter that includes the perception of the decision-maker?
- How relevant is the incorporation of different leaks and pressures threshold as a source of uncertainty in the evaluation of sensor networks?

1.3 Innovation and practical value

According to the gaps identified, we can justify that the methodology developed in this project will have as innovation the following aspects:

- The use of a pressure-driven model in developing a computational framework based on a hybrid value of information and information-entropy approach for the optimal location of sensors.
- The application of a multi-objective optimisation considering the perception of the decision-maker.

As practical value, an improved insight for the deployment of pressure sensors for leak detection is expected; this could reduce water volume that different distribution systems are currently losing. In particular, for the case of the city of Braila (Romania), which has the goal of reducing water losses caused by leaks by from a value of 750 L/h/km to a value of 700 L/h/km.

Another practical value of this methodology could be attributed to applying it in a real scenario; this will help clarify its validity. Real tests can be performed in future scenarios, simulating leaks in the network, and verifying if the selected nodes detected the leaks.

1.4 Thesis outline

This thesis is structured in 6 chapters:

Chapter 1. presents the introduction to the topic, objectives, investigation gaps and research questions as already explained.

Chapter 2. shows the state of the art of the different methodologies used, explains the theoretical foundations on which our methodology is based and justifies the different assumptions established.

Chapter 3. describes the generalities and characteristics of the distribution system where each of the analyses will be realized, the initial information provided and the objectives of the study sector regarding the results of this thesis.

Chapter 4. describes in detail the methodology proposed to achieve the research objective, formulates each of the steps needed to perform the different simulations and indicates the workflows involved in the implementation of the model simulation. Additionally, it defines the optimisation problem and specified the objective functions and the resolution algorithms required.

Chapter 5. provides examples of calculation, analysis and discussion of the results obtained, as well as the use of the methodology in a real scenario, comparing its functionality against an already installed sensor network.

Chapter 6. shows conclusions on the different results obtained, and provides an answer to the questions formulated in chapter 1. In addition, it provides recommendations for future research.

References and appendices are included at the end of the document.

Chapter 2 Literature Review

Chapter 2 discusses different ideas about the optimal deployment of pressure sensors. It describes some limitations, findings and proposals that motivated the elaboration of this thesis. Similarly, it illustrates the equations needed to develop the proposed methodology. This chapter also presents concepts such as the relationship between pressure and demand, the requirements for calculating system leaks, the pressure-based models, as well as the selected objective functions and the process for their optimisation.

All the concepts specified in this chapter will be used in the development of the methodology described in Chapter 4

2.1 Methods to detect leaks in the system

Water utilities regularly work on developing methods that help locate leaks in the least possible time to take the corresponding decision-making actions to reduce water loss and other related problems. These methods can be active and/or passive, depending on the strategy used. On the one hand, the active methods include a physical inspection of the pipes, the use of acoustic sensors or gas tracers, ground penetration radar, infra-red imaging, among others (M. Farley & Trow, 2015). However, although the leak can be pinpointed directly, these approaches are not appropriate for large systems because of economic limitations (Sanz et al., 2016). On the other hand, the passive methods depend on secondary evidence of leaks, such as checking unexpected changes in the network's pressure values (Raei et al., 2019). Generally, the presence of leaks increases the flow and velocities in the network, causing larger hydraulic losses and generating pressure drops (Tavamani, 2016). Therefore, if the pressure values detected at a node are outside the normal range, we could attribute it to the presence of leaks. A common way to estimate the expected pressure values in a network is by using computer models, which are mathematical representations of the entire network, that simulate after proper calibration, their behaviour for a range of functioning scenarios. Furthermore, real pressure values are obtained from measurements in the field, for example, with the help of pressure sensors.

The use of passive methods are preferred because their operational costs are lower than the active methods (Raei et al., 2018), however, these optimisation methodologies are affected by the high computational load, many simulations are required for networks with many nodes and consequently, the problem has to be simplified, decreasing its decision space (skeletonization), also, the use of faster but less accurate optimisation algorithms are required (Mohammad S Khorshidi et al., 2018). However, nowadays, with the development of more powerful computers, the computing burden has been radically reduced, allowing methodologies to be implemented without adjusting the original problem. For this methodologic all the nodes of the system will be explored as possible candidates.

One of the most widely implemented methodologies to locate pressure sensors is based on the comparison between the pressure obtained from the simulation without incorporating leaks (healthy state) and the pressures calculated in each of the nodes subjected to leak conditions (modified state). Sensitivity analyses have been incorporated to these pressure deltas in which the differences are normalized dividing them by the value of the leak that caused the disturbance, thus generating a sensitivity matrix composed of residual vectors. (Pudar & Liggett, 1992; B. Farley, Boxall, and Mounce 2009; Pérez et al. 2009; Sanz et al. 2016; Raei et al. 2018; Quintiliani and Vertommen 2020). Additionally, diverse authors have incorporated in their analyses sources of uncertainty such as variation in demands, leak sizes, threshold variation, physical properties of the pipelines, among others (Nejjari et al., 2015; Steffelbauer & Fuchs-Hanusch, 2016; Taravatrooy et al., 2020).

Blesa et al., (2014), obtained important findings such as that the location of the sensors are not sensitive to the size of the leaks and Raei et al., (2019) found that the variation of the threshold may have a negligible effect on the location of the sensors. However, most of the researchers applied methods based on a demand-driven (DDA), assuming that all nodes' demand could be supplied regardless of the network's pressure values, being an assumption distant from reality. When leaks occur, there may be interruptions in water service, i.e., nodes where the pressure is too low that demands are reduced or not supplied, this analysis can be performed using pressure-driven analysis (PDA) which offers more realistic results than demand-driven models (Braun et al., 2017), because the demands are function of the pressure, therefore, previous analyses need to be updated, incorporating this new approach.

The optimisation problem for sensor localization has been a widely studied topic, and for the reader's judgement it is indicated that not all references in this field were consulted, some of the most recent research was considered and an evaluation was conducted to assess what aspects might be missing and what could be complemented by this research, some of the methods investigated are presented below.

Table 2.1-1. Literature review, findings, gaps and proposals.

Author	What was proposed?	What is missing and what is proposed
(Mohammad Sadegh Khorshidi et al., 2020)	This research proposes a workflow for optimal sensor deployment, incorporating hybrid information-entropy approach, concepts such as VOI and Transinformation Entropy (TE). In addition, it proposes a function to select the best solution within the set of solutions resulting from the optimisation process.	In the VOI formulation, unintuitive parameters for the decision-maker were used, for example, the cost and decision matrix did not consist of either monetary values or real actions that the decision-maker could execute, so that the critical evaluation of what decision to take once the information from a sensor was received could not be executed. On the other hand, considerably low leaks (0.2 l/s - 0.5 l/s) were evaluated to generate pressure drops of one (1) mH2O, this response of the network may be associated with the normal variation that exists in distribution systems. This work will include the decision-maker's perception of the system and will incorporate realistic economic values for each action to be performed in the cost matrix for VOI calculation. Additionally, different types of leaks to review the behaviour of the network under different scenarios will be evaluated.

Author	What was proposed?	What is missing and what is proposed
(Quintiliani & Vertommen, 2020)	This work proposes a methodology that uses numerical optimisation techniques combined with the engineering judgement provided by the collaboration of water utilities to determine the most optimal number and layout of pressure sensors., additionally, it applies the methodology to a real WDS and formulates the project's own needs as an objective function.	The methodology was evaluated using a single constant leak value, concluding that approximately 50% of the network could not be detected, part of this lack of coverage was because some of the nodes where the leaks were incorporated did not cause pressure drops in the network higher than the selected threshold value, indicating that the leak tested influence just a certain part of the network. At the same time, the analysis was applied to a real-world problem; however, it is not clear if was assumed the use of a demand-driven approach, considering that node demands could always be supplied despite operating the network with water losses of 60 m3/h (≈17 l/s) caused by leaks. In this work is proposed to analyses several leak values to guarantee pressure drops in a larger number of network’s nodes, plus, the incorporation of a pressure-based model to evaluate variations in the demand caused by pressure drop resulting from the addition of leaks in the system.
(Raei et al., 2019)	This methodology was based on a sensitivity matrix approach, to create a list of potential sensor locations, exploring objectives such as minimizing the number of sensors and detection time. In addition, they incorporated the measurement of pressures as a source of uncertainty, exploring different errors implicit in the devices used to capture pressure values.	Different threshold values were considered to evaluate sources of uncertainty; however, the values were similar (0.25 - 0.75 mH2O) and too small to be used in real-life scenarios. It was concluded that 10 sensors could detect leaks in the entire network, but this results could be associated to the threshold values selected, because such small pressure variations can be produced at any point in the network, thus biasing the detection results. In this methodology we propose to evaluate threshold values more in line with those expected in reality and with a wider range of evaluation (0.5 to 3 mH2O), in addition we will incorporate leaks that are a function of the threshold to guarantee the pressure drop analysis in a higher percentage of the network.

2.2 Pressures and Demands in WDS

In water distribution systems, the fluid's energy is represented by three components (pressure, elevation, and velocity). The pressure and elevation conform the piezometric energy, and velocity conforms the kinetic energy. According to Bernoulli's law, it is established that the energy must remain constant along the streamlines, since it is assumed that the mass and density in a closed system (water volume) does not change (Tavamani, 2016). If the system gains kinetic energy, the system loses piezometric energy, and vice versa, consistent with the second law of thermodynamics, energy is not created or destroyed, it is only transformed.

When leaks are introduced at the system, an artificial increase in demand is caused, and variations in the system pressure are experimented. In conditions where there are no external sources of energy, such as pumps, the increment in the demand at a node would cause a reduction in pressure, not only at the node where the leak occurs but in the entire system (energy transformation), as the elevation of a node in a distribution network always remains fixed, from now on, every time we talk about an increase in demand, we must think about a pressure drop.

As explained above, the pressures of the distribution networks will depend mainly on the water consumption required in the system; this consumption is a function of the characteristics of the users who require the service and how the water is distributed over time; the latter is called consumption pattern. These consumption patterns are calculated through historical measurements of the network, which estimate on average the users' consumption at different times of the day, being a parameter that changes over time and needs to be updated as the network evolves (García, 2003).

This variation in demands is similarly reflected in the pressure behaviour of a network; Figure 2.2-1 shows how the pressure variation of a typical node could be within a 24-hour range; the pressure distribution varies over time creating a pressure envelope, where exist values of pressure over and below an averaged value.

This average pressure is called the initial or healthy state of the node (So), representing the pressure values without leaks and sets the comparison point with the new pressures generated when leaks are added.

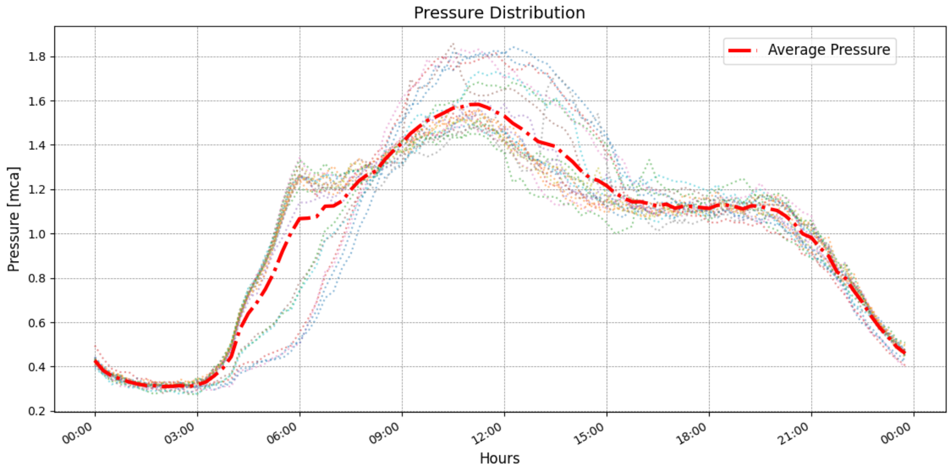


Figure 2.2-1 Pressure variation for a node along the day[1] at one particular network location.

[1] This figure is illustrative and does not correspond to the pressure values of Radu Negru system

2.2.1 Definition of the pressure threshold (Th)

The methodology used to identify leaks consists of localizing changes in the pressure system based on a model-based fault diagnosis, where leaks are added node by node, and pressure variation is read at other nodes. For each node, the new pressures are compared with the initial state (So), when the pressure differences are higher than the pressure threshold (Th), we could infer the presence of leaks.

This threshold value depends on three factors: the pressure sensors' sensitivity, pressure sensors' accuracy, and the system's pressure variation (Quintiliani & Vertommen, 2020). In practice, only when new pressures at one node are out of the values above and below the average (see Figure 2.2-1) could we consider that the anomaly is caused precisely by the presence of leaks; otherwise, any variation that is within a node's pressure envelope could be caused by the regular use of the system.

The estimation of threshold values based on historical pressure data provides certainty because the limits used are adjusted to the variation of the system pressure, considering the temporal evolution of the demands in the system (Sanz et al., 2016). However, in cases of insufficient information, it is not possible to establish an exact threshold value for the system. For this reason, various threshold values should be evaluated in order to assess the behaviour of the system at different pressure drops.

As the main objective of this study is to obtain signals of the presence of leaks, we defined a range of typical threshold values that could be attributed to the start of a leak, accepting that in some cases, pressure divergences could be attributed to daily use of the network and not due to leaks.

2.2.2 Leaks estimation

A way to estimate the leak flow in a pipe is correlating the pressure in the system with physical characteristic of the leak, for example, for hole type leaks, which in pressurized systems will act as an orifice, the flow can be represented as a function of the velocity and the area:

$$V = C_d\sqrt{2gP} \qquad [1]$$

Where V is the velocity of water through the orifice (m/s)

C_d is the discharge coefficient (dimensionless)

g is the gravity (m/s^2)

P is the pressure head in meters

For an orifice of a specific area (A), the flow in m3/s ($V * A$) is a function of a factor proportional to P^{α} ($\alpha = 0.5$) (M. Farley & Trow, 2015); however, in practice, it has been estimated that the proportional factor can range from $P^{0.42}$ to $P^{2.3}$ depending on the type of material, leak type and leak size (Greyvenstein & Van Zyl, 2007). In this way, leaks in a system can take a wide range of values because the orifice's size and shape in a pipeline may vary from pipe to pipe.

Several researchers have used the methodology of the orifice for leak estimation, Taravatrooy et al., (2020); Mohammad Sadegh Khorshidi et al., (2020); Raei et al., (2019) have suggested tuning the coefficients of C_d to generate leak in the range of 0.2 y 0.5 l/s arguing that these values ensure a minimum change in system pressures. In contrast Pérez et al., (2009), Quintiliani & Vertommen, (2020) have opted to tune the coefficients to estimate leak that correspond to values that have typically been measured in the system.

The challenge of assuming leaks as a function of pressure, is to obtain different leak flows for every simulation step, because pressures changes at every part of the day (demand pattern); therefore, it is uncertain which exact leak is being evaluated. In contrast, selecting a unique value of leak causes that the deployment of sensors will only be functional for the estimation of that unique leak. For this reason, the leak estimation will not be a function of the coefficients or pressures; on the contrary, a range of constant leaks that generate pressure drops above the selected threshold will be selected, they will be included in the system as extra demands, being constant throughout the modelling time. In this way, it is possible to have control of the leak being modelled in each part of the network and at each instant of the simulation; in addition, with the inclusion of multiple leaks, more robust solutions can be obtained, functional to multiple scenarios and not only to one as suggested by other researchers.

2.3 Demand-Driven Analysis (DDA) and Pressure-Driven Analysis (PDA)

A common way to estimate the expected pressure values in a network is by using computer models, which are mathematical representations of the entire network that simulate –after proper calibration, their behaviour for a range of functioning scenarios. Currently, many tools allow us to perform these hydraulic simulations where the solution algorithms mostly solve the equations based on a demand-based analysis (DDA), assuming that the system's demands are always supplied and are independent of the pressure that exists in the system. (Reddy & Elango, 1989).

This assumption is adequate when the network is in typical situations, where pressures are high enough to supply the system; some studies (Cheung et al., 2005, Braun et al., 2017, Germanopoulos, 1985) have shown that the use of DDA is not adequate when it is desired to simulate abnormal scenarios such as the simulation of leaks where the system is induced to have pressure drops, considering that the demands must be a function of pressure and not constant. This type of models is called pressure-driven analysis (PDA) and have as main change the assumption that the demand at a node is constant if the pressure of this node is above a fixed value, the demand is zero if the pressure drops below a critical value and the demand as a function of pressure if it is within an intermediate value of pressures, (Raei et al., 2019).

One of the main tools used for network simulation is EPANET; this computational tool has allowed solving hydraulic equations based on the conservation of mass and energy through systems of linear and nonlinear equations, using solution methods such as the Newton-Raphson technique, based on the use of a DDA, (Rossman, 2000).

Several researchers have developed external tools to use the EPANET algorithm and implement a PDA approach by modifying their solution systems. (Cheung et al., 2005, Pathirana, 2012, Muranho et al., 2014), The United States Environmental Protection Agency (EPA) has also officially released the EPANET tool update, including the necessary modifications to perform pressure-based analysis based on Wagner's equation. (Wagner et al., 1988):

$$d_i = \begin{cases} D_i & p_i \geq P_f \\ D_i \left(\frac{p_i - P_0}{P_f - P_0} \right)^e & P_0 < p_i < P_f \\ 0 & p_i \leq P_0 \end{cases} \qquad [2]$$

Where:

D_i: Full normal demand at node i
p_i: Pressure at node i
P_f: Limit pressure; above this value, the pressure is supplied.
P_0: Limit pressure; below this value, the pressure is not supplied.
e: orifice exponent, normally equal to 0.5 (to mimic flow through an orifice)

With the incorporation of these equations, leak simulation analyses can be performed and the results obtained are more realistic. The EPANET 2.2 engine will be used in the development of this research.

2.4 Value of Information (VOI)

The majority of daily decisions are based on preconceived concepts or criteria derived from experience; for example, if we see a cloudy sky and feel the humidity in the environment, experience tells us that it is likely to rain; in numerical terms, we would say that, based on our experience, there could be a 70% to 80% of chance of rain. Although this estimation is subjective, it influences the final decision of taking an umbrella or going out without it.

Similarly, in WDS it is possible to estimate if a particular site is prone to leaks, based on a historical record of the failures in the system, with information from the residents of the neighbourhoods, analysis of the age of the pipes, checking their location on roads with heavy traffic or because there are tree roots in the vicinity, etc. Thus, all these factors can lead to the existence of a prior perception or belief about the condition of the network. A decision-maker uses these preconceived judgments to estimate the potential location of the network problem and based on these assumptions, will take the appropriate action, for example, go and check or simply do not take any action at all.

However, what happens if we tell to the decision-maker that his/her initial perception can be improved by investing in a sensor's network that gives information about the pressure variation in the system and could estimate with a certain probability that the pressure variation was caused due to the presence of leaks, it is likely that the decision-maker may decide to invest in this new source of information, in contrast, if the decision-maker receives the information that these sensors are also likely to report false alarms, sending messages of leaks when in reality the pressure variation was caused by normal system conditions, and in consequence, a work team is sent to a place where it is not required, wasting money and time, in this opportunity is possible that the decision-maker think twice in acquiring these sensors, having to evaluate the quality of the new information.

The process to evaluate the quality of new information is approached through the concept of the value of information (VOI). This concept appeared in the decade of the 1960s with the work realized by Grayson (1960); specifically in the economic sector of oil and gas and previously conceptualized in the work of Hirshleifer & Riley (1979). In the field of monitoring network design and the area of water management VOI has also been explored, it has also been used in the design of monitoring networks to detect and reduce flood impacts (Alfonso & Price, 2012), the estimation of probabilistic flood maps (L. Alfonso et al., 2016), the design of groundwater level monitoring networks (Mohammad S Khorshidi et al., 2018), and recently for the optimal location of pressure sensors in water distribution systems (Mohammad Sadegh Khorshidi et al., 2020; Mohammad S Khorshidi et al., 2018), These studies will be referenced in this document and form a base for this thesis.

In the value of information concept, the perception (based on own experience) about the state of a system (s) is measured by probabilities or percentages of occurrence and is denoted as prior belief (π_s). Depending on the available states, the decision-maker can choose among a group of actions (a) selecting the one which generates the maximum profit or the minimum loss. To evaluate the utility of each action, the Neumann-Morgenstern expected utility rule is used, in which the state's probabilities are multiplied by the costs or consequences of each action. $(C_{a,\,s})$:

$$u\,(a, \pi_s) = \sum_s C_{a,\,s} * \pi_s \qquad [3]$$

Subsequently, the decision-maker select the action that is most useful to him/her, in this way, the action with higher utility is obtained as follow:

$$u\,(a_0, \pi_s) = \max\{u(a, \pi_s)\} \qquad [4]$$

Once the decision-maker has the chance to acquire new information, she/he should be disposed to believe in this new information and give it utility, otherwise, the value of this information becomes zero and her/his final decision will be based only on her/his initial belief. The new information refers to the external opinion received, which in the VOI concept is represented as the message (*m*), and in this particular case, the messages come from the signals transferred by the pressure sensors. Accepting the new information implies that the initial perception must be updated; this update can be represented as follows:

$$\pi_{s,m} = \frac{q_{m,\,s} * \pi_s}{\sum_s q_{m,\,s} * \pi_s} \qquad [5]$$

Where $\pi_{s,m}$ refers to the updating of the perception and $q_{m,\,s}$ indicates the probability of receiving new information (*m*) in each of the system states, the signals sent by the sensors have the particularity that they can transfer information that is in line with the state of the system, for example, pressure drops signals when existing leaks or transfer false alarms, pressure drops signal when there were no leaks or no signals when existing leaks.

Once the system perception has been updated, the utility of taking actions using this new information must be evaluated in the same way that the initial utility was evaluated, thus:

$$u\left(a, \pi_{s,m}\right) = \sum_s C_{a,\,s} * \pi_{s,m} \qquad [6]$$

Lastly, the value of the message received is calculated as the difference between the utility of performing an action based on prior beliefs and the utility of acting once new information is acquired:

$$\Delta_m = \max\{u\left(a, \pi_{s,m}\right)\} - \max\{u(a, \pi_s)\} \qquad [7]$$

Because there is a combination of possible states (s) of the system and possible messages (m) that can be transferred for each state, the total value of the information will be the utility of each of the possible messages; hence VOI is equal to:

$$VOI = \sum_{m} \left(\sum_{s} q_{m,\,s} * \pi_s \right) * \Delta_m \qquad [8]$$

In this way, VOI is a function of three main variables, the prior belief (π_s), the costs of taking an action ($C_{a,\,s}$) and the likelihood of receiving new and accurate information about the state of the system ($q_{m,\,s}$) this for each of the states. In this methodology, the states of each of the nodes will be represented as 'Leak' and 'Not Leak', the costs will be associated with the actions 'Go to check' and 'Do not go to check', and the messages are transferred by the different pressure sensors to be installed, transferring 'Detected' and 'Non detected' signals. Each time we refer to a node that transfers a message, we will indicate it with ***j***, and the node causing the disturbance will be the point where the leak is added and will be denoted with ***i***.

Although the value of the information should be compared to the cost of receiving this new information, i.e., the cost of acquiring and installing these new sensors, it will be assumed to be a constant for all possible sensor deployments and only the parameters mentioned in the equation [8] will be used.

2.5 Information Entropy (IE)

In section 2.2 it was indicated that a sensor would transfer a signal if there are pressure drops greater than a certain threshold. These pressure drops can occur in any part of the network because pipes and joints are part of a single interconnected system. However, the nodes closer to a point where a disturbance is generated shall be more affected than the points located far away, thus the disturbance of a node has a specific area of influence; similarly, the pressure sensors will have more probability of detection if they are installed closer to the point of disturbance.

When installing a group of sensors, it is desired to obtain information from the majority of the network, which means that by adding the detection areas of each one of the sensors, it is possible to obtain the maximum coverage. A way to quantify the detection coverage of a sensor network is by determining how varied is the set of nodes that were detected, the more diversification, the higher the coverage. In information theory, this level of diversification is related to the level of uncertainty or entropy in the data set, being greater the variability of the data when the entropy is higher. Mathematically the entropy level of a random data set can be represented as:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \qquad [9]$$

The units of uncertainty are given by the base of the logarithm utilized, being 'bits' for the case of base 2, another consideration is that $0 * \log_2 0 = 0$ because values with zero probability are not adding or removing information, so the uncertainty remains the same.

As indicated above, this level of entropy will be higher if the data set is composed of a larger number of variables, e.g., in a basket with two balls of different colour, the probability of picking one ball is equal to 0.5 (1/2), so the uncertainty or entropy will be equal to:

$$H(X) = -((0.5 * \log_2 0.5) + (0.5 * \log_2 0.5)) = 1$$

If to this basket we add a new ball with a different colour, the probability of picking each ball is equal to 0.333 (1/3), and the entropy is equal to:

$$H(X) = -((0.33 * \log_2 0.33) + (0.33 * \log_2 0.33) + (0.33 * \log_2 0.33) = 1.58$$

The uncertainty or entropy of knowing which ball will be selected is now higher. In the case of leak detection, the greater the number of nodes detected, the greater the level of uncertainty of knowing which node was detected, so the higher the entropy, the greater the network coverage.

This concept of diversification and entropy is contrary to the assumptions presented by Mohammad S Khorshidi et al. (2018) and Mohammad Sadegh Khorshidi et al. (2020), where researchers intended to avoid redundant information, that is, to avoid sensors that transfer information from the same sectors of the system, this assumption could be improved, because there is no problem in sensors sharing information, as long as they have coverage, in fact, it is more efficient to have sensors that warn about pressure drops in the same zones because if any of them fails, the others can provide support, so a network of sensors that have greater coverage and that share information between them will be the most ideal and optimal.

2.6 Optimisation

In water distribution systems could exist areas that are more vulnerable to leaks or areas where a leak flow would be more critical for the system, either because of the number of users that are disconnected, the amount of water lost, structural damages, traffic interruption, among others, these types of variants suggest that identifying leaks at these points has a higher priority and it is more valuable for the decision-maker.

In some cases, entities only have access to a limited number of sensors, either due to lack of budget or technical difficulties, and it is challenging to capture pressure drops in the whole WDS, because of this, there is a need to utilize the resources that are available and optimise them in order to include as much area as possible considering the main areas of interest. However, quantity and quality can be opposites, a sensor network that guarantees to locate leaks in the most vulnerable points is not the same as a sensor network that has larger coverage, and having larger coverage does not imply that leaks in the most needed areas are captured, thus, in some cases coverage and quality cannot be guaranteed. Since it is impossible to have sensors all over the network, there is a need to use numerical techniques that allow us to identify the optimal distribution that satisfies each of the objectives.

For this type of exercises in which the objective functions cannot be represented analytically, it is required to use optimisation methodologies based on direct search, in which the most optimal solution is selected from a set of finite solutions, such as the use of evolutionary algorithms (Marquez-Calvo, 2020). This type of algorithm is based on Darwin's theory of evolution, simulating the biological evolution of the fittest or optimal solutions. Figure 2.6-1 explains the process to obtain the optimal solutions, the steps are mainly divided into five parts, I. Selection of random initial candidates (Parents), II. Choose the best Parents to reproduce, III. Vary genes of parents to generate a new solution, IV. Create parameter of evaluation to decide if new solutions (births) are better than initial solutions (Parents), V. Mixing the population, holding best initial parents, and including new optimal solutions (births), finally, the mixed population is used as initial candidates. The process finishes once the stopping criteria is reached (Marquez-Calvo, 2020).

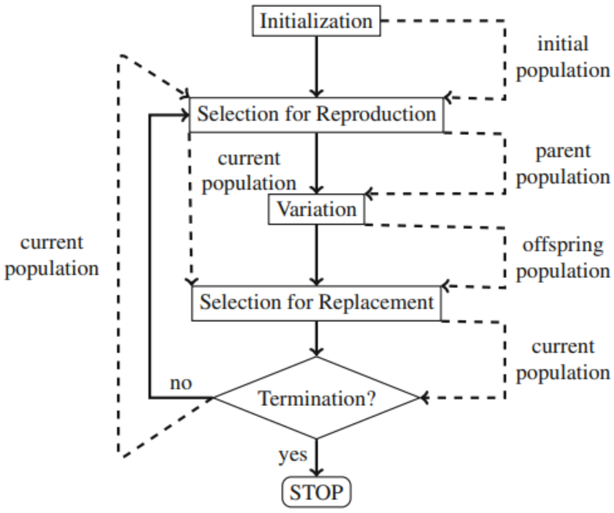
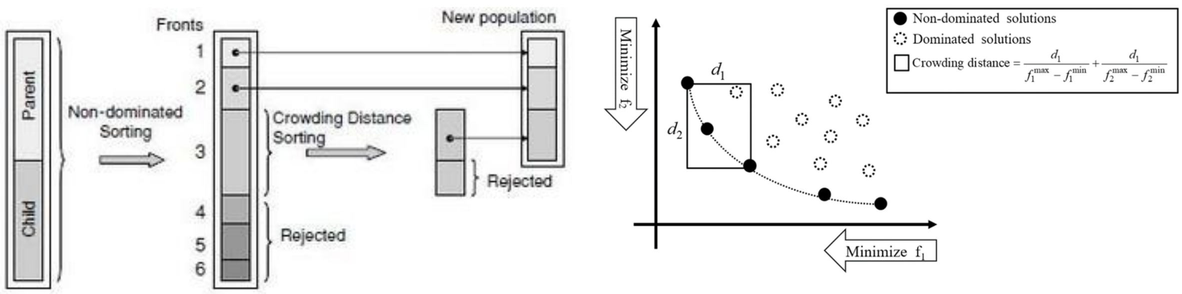


Figure 2.6-1 Flow chart of a genetic evolutionary algorithm.
(Solid arrows: Control flow, Dashed arrows: Data flow). (Jansen, 2013)

One of the most common evolutionary algorithms for the solution of optimisation problems is the NSGA-II (Non-Dominated Sorting Genetic Algorithm) which has the characteristic of being able to optimise multiple objective functions and ranks sets of decision variables simultaneously, based on dominance to find non-dominated solutions.

The result of this algorithm is a set of optimal solutions, also known as the Pareto-optimal set, in which each of the solutions identified is not better or worse than the others; they are solutions where the improvement of one of the objective cannot be achieved without degrading the other objective functions, in the case of the sensor arrangement, leak detection quality will be lost every time coverage is gained. Figure 2.6-2 illustrates the steps in which the algorithm generates new populations from initial candidates, formed by (parents) and offspring population (child) product of gene manipulation (a), and indicates how is the qualification process for the selection of the best solutions (non-dominated solutions) that are reused in the next generation (b).



(a) (b)

Figure 2.6-2 (a) Representation of NSGA II. (Deb et al., 2002), (b) Concept of non-dominated sorting and crowding distance approach. (Choi & Kim, 2019).

Chapter 3 Case study

This chapter presents the general description of the study area, the base information used to develop the different simulations, and the needs that exist in the sector. The information used in this chapter will be manipulated by using the methodology proposed in chapter 4, and the results will be discussed in chapter 5.

3.1 Study Area: water distribution system (WDS) of Radu Negru – Braila

Radu Negru is one of the district metered areas (DMA) of the city of Braila, Romania. This zone has 2.6 km^2 and is supplied from the Danube river, the raw water is collected through a bank intake and is sent to the Chiscani treatment plant, where it is treated and transported to the area of Radu Negru.

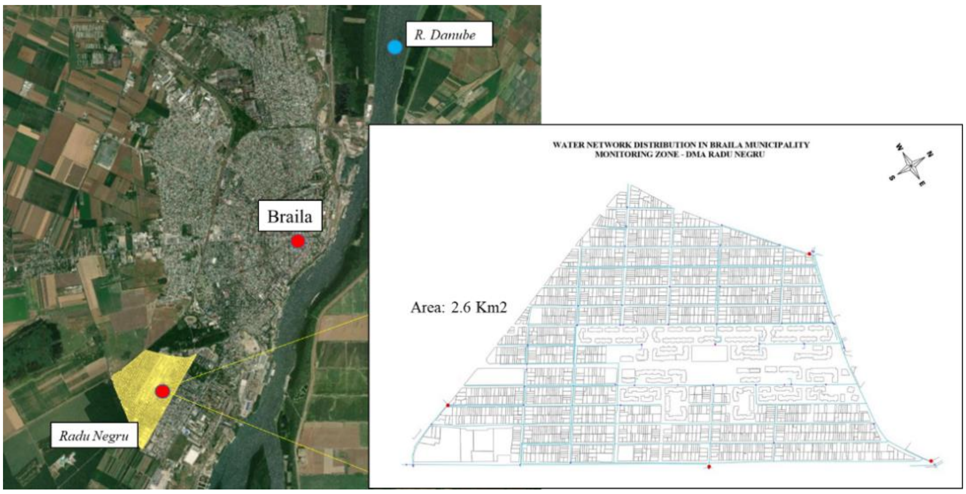


Figure 3.1-1 Location of the project.

For the distribution system, Braila's water authority has a calibrated hydraulic model; this is a model in *.inp* format and can be read by the modelling tool EPANET (Rossman, 2000). The model includes the whole WDS of the city; however, just the Radu Negru zone was used for our research. For this reason, Braila's water authority modified the complete model, selecting the DMA of Radu Negru and changing four of its inflow nodes for thanks and pumps in a way that the pressures and demands inside the new zone would remain the same as in the complete city model.

The new model is composed by:

- Number of Junctions........... 305
- Number of Pipes 254
- Number of Reservoirs......... 4

- Number of Pumps 4
- Number of Valves…………70

The part of the model used is the area is shown in Figure 3.1-2.

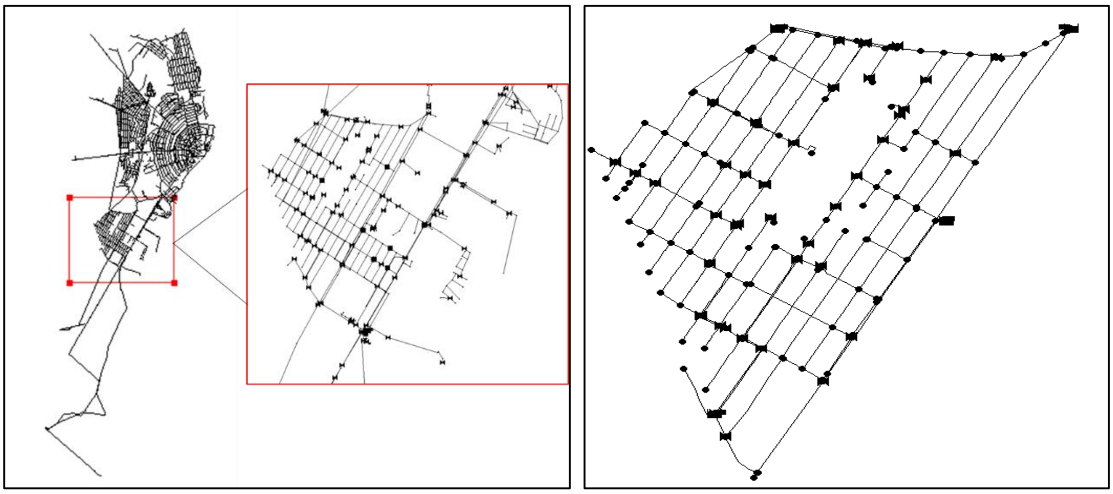


Figure 3.1-2 Model of the WDS of Radu Negru - Braila.

In addition to the hydraulic model, shapes of the cadastral information, the pipes' characteristics, service nodes, hydrants, roads, and primary connections was also provided.

This information was filtered to obtain only the information required in the sector of the Radu Negru sector.

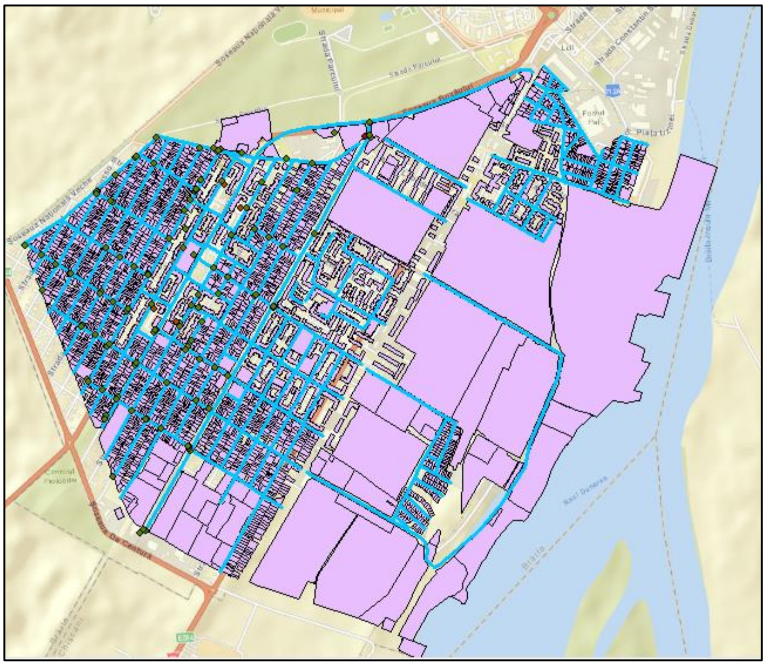


Figure 3.1-3 Shapes files of the WDS-Braila.

The DMA of Radu Negru has four pressure sensors already installed; these sensors' locations were selected because, at the same points, measurements of flow will be installed, therefore, the Braila authorities considered that this would be the best location. The distribution of the sensors is shown in Figure 3.1-4.

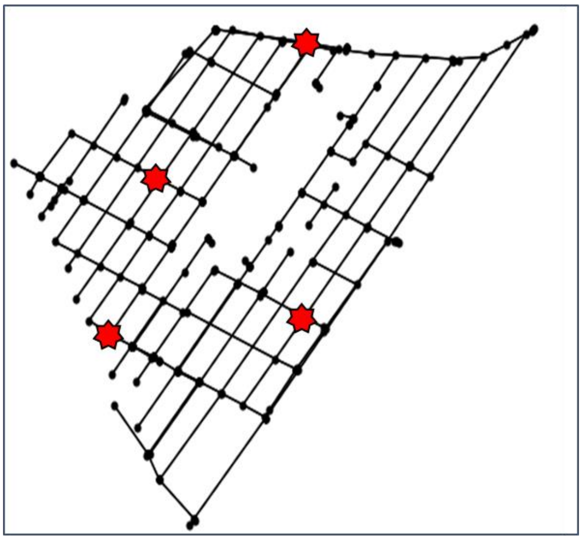


Figure 3.1-4 Location of the pressure sensors installed in DMA of Radu Negru.

One of Braila's objectives is installing four extra pressure sensors (for a total of 8 sensors); therefore, our methodology's results will serve as a basis for decision-makers to define these new sensors' final location, however, our methodology is generic enough to be used in other distribution networks.

3.1.1 Model characteristics

Radu Negru has a hydraulic model which was calibrated by the water utility using the EPANET tool; For calibration purposes, four pumping stations were added to the model, which in reality do not exist but serve to maintain the district's pressures as close to reality as possible (as reported in a personal communication).

The pumping stations are located as shown in Figure 3.1-5.

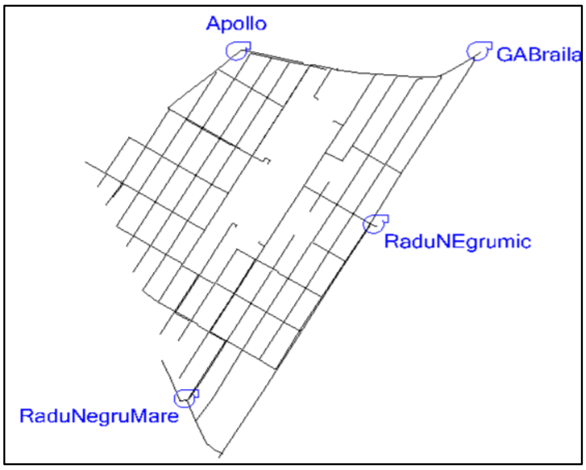


Figure 3.1-5 Location of virtual pumping stations.

Each node was affected by a unique consumption pattern to simulate consumption variation throughout the day; consumption above the average was shown from 8 am to 2 pm and from 6 pm to 12 pm.

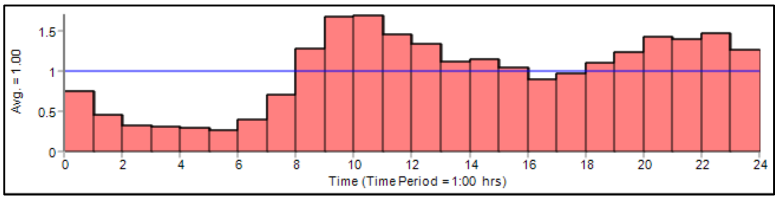


Figure 3.1-6 Consumption Pattern established by the water utility for the Radu Negru model.

According to the pressure zones, Radu Negru has low zone pressures at the north side, where the terrain level is high, the minimum pressure is equal to 8.31 mH2O. In the south part, the high zone pressure presents average pressures of 20 mH2O.

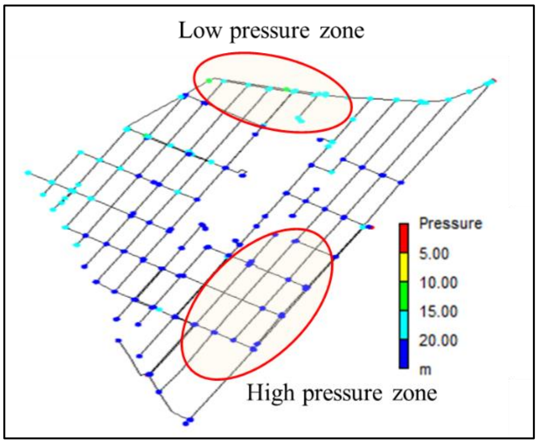


Figure 3.1-7 Pressure zone of Radu Negru.

The pressure system drops from 9:00 to 11:00 and 19:00 to 21:00 and is caused by the increment in consumption at these hours. On the other hand, in the early morning hours (00:00 am to 6:00 am) system has the highest pressures, with a constant value and presenting minimum variation in the sector's consumption.

An example of the variety of pressures and demand in one of the nodes located in the high-pressure zone is shown in Figure 3.1-8

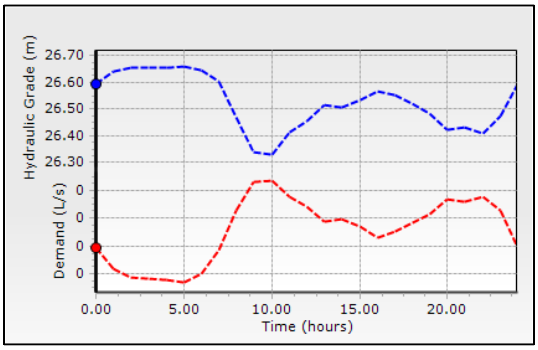


Figure 3.1-8 Pressures and demand variation of a node located in the high-pressure zone of Radu Negru.

In some methodologies, simulations are performed in the hours of low consumption, where nominal pressures are high, but pressure variation is low, being more difficult to detect leaks and representing the worst scenario (Quintiliani & Vertommen, 2020), however, in the proposed methodology, the pressure variation will be studied at all times of the day, the objective is to detect an initial perturbation at any time of the day. Chapter 4 will indicate how these leaks' values will be defined and how their respective simulation will be performed.

Chapter 4 Methodology

4.1 Introduction

The proposed methodology seeks to find optimal locations for pressure sensors, such that they can detect pressure drops higher than a selected threshold (Th) that can be associated with the occurrence of leaks. The process consists of producing a collection of initial pressures and pressure deviations per node and per time step, which are generated as a response to discrete changes in demands (leaked network). The deviations are established with respect to pressures obtained from a 'healthy network', i.e., a water distribution network without leaks. The set of pressures arranged by time of the day for all nodes is called 'state matrix' as shown in section 4.2.1.

To generate the leaked network, it is required to define the leaks that will be added to the system according to the required pressure drops and the desired coverage area. for this reason it is defined in section 4.2.2 how the leaks will be incorporated in each of the nodes $\boldsymbol{j}$, and in section 4.2.3 indicates how the leak size and the pressure threshold will be estimated for each of the comparisons.

After obtaining the pressure deviation matrices, each scenario is classified as 'detected' every time that a node $\boldsymbol{j}$ presents pressure drops higher than the threshold (Th) due to the addition of a leak flow ($\boldsymbol{L_f}$) at node $\boldsymbol{i}$, as indicated in section 4.2.4, each node of the system will be classified according to the number of times that its pressure drops above the threshold in each of the leak scenarios, thus defining a percentage of detection. Similarly, for each scenario, the minimum leak required for each node $\boldsymbol{i}$ to be detected by node $\boldsymbol{j}$ will be calculated.

Once the hydraulic parameters have been defined, the VOI and IE objective functions are obtained, section 4.3 describe how they are calculated using the information obtained in section 4.2 and the formulation of the optimisation problem is presented in which the goal is to locate pressure sensors by maximizing their value of the information (VOI) and information entropy (IE).

Section 4.4 describes the configuration process for the model simulation in Epanet and the libraries used for the development of the algorithm in Python that solved all the formulations described in sections 4.2 and 4.3. The general description of the methodology can be seen in Figure 4.1-1 specific steps and workflows will be discussed later.

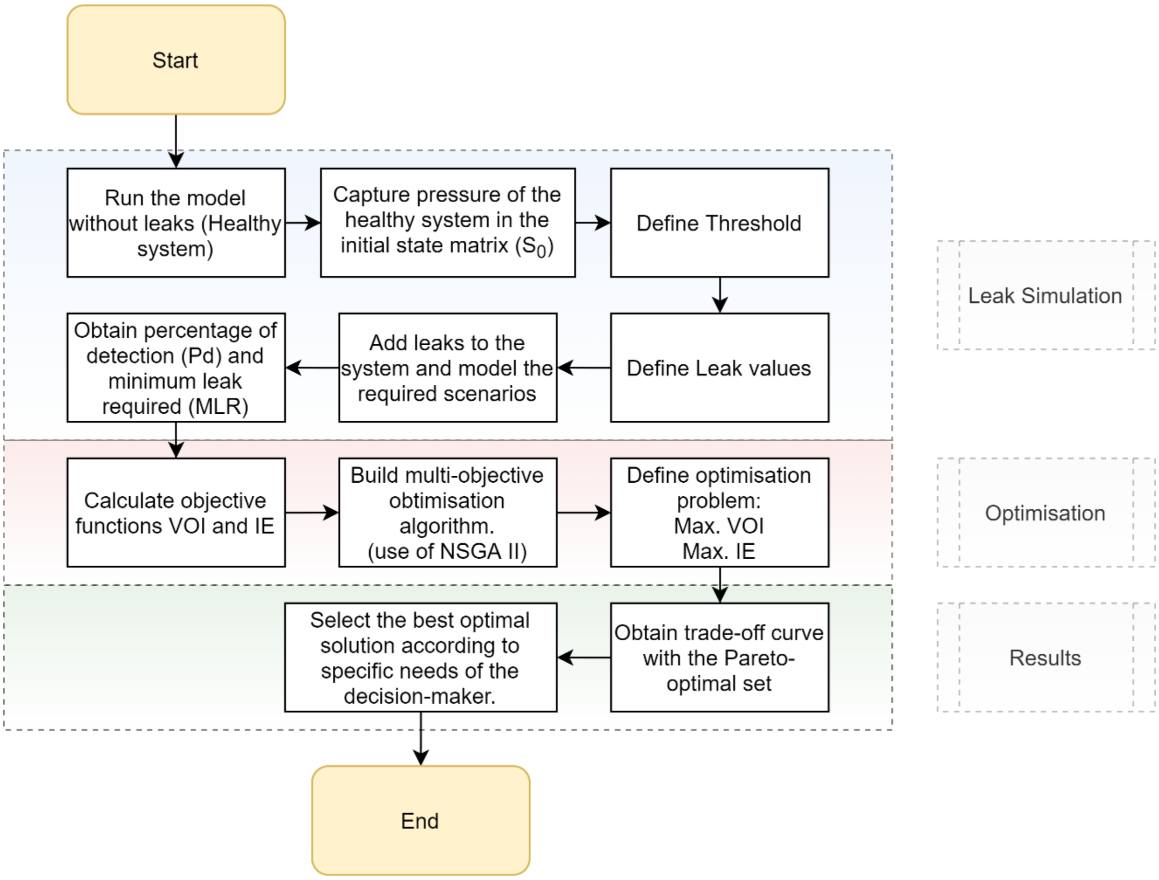


Figure 4.1-1 Overall workflow of the methodology.

4.2 Formulation for the leak simulation in WDS

Below the steps required to perform the leakage simulation in a WDS are presented.

4.2.1 Definition of the Initial State (So) and Modified State (S) Matrix

The initial state (So) of the network is obtained using the unmodified hydraulic model results; each node's pressure values are stored in a matrix and used as an initial point of comparison.

Assuming that the model provided represents the healthy system (without leaks), the process to obtain the pressures is to run the original model and read the resulting hourly pressures ***p*** per node ***i*** ($p_{i,T}^{So}$) in a distribution network of N nodes.

Each time a new simulation is performed due to a system's modification, the new pressures of the whole system are obtained in a similar way to those obtained in the initial state (So). This new matrix will be defined as the modified state matrix (S). The matrices representing the initial state and modified state consist of N columns indicating the number of nodes and T rows indicating the time steps. Each column shows the pressure at the node along the day, as indicated in Table 4.2-1.

Table 4.2-1. Initial state matrix, pressures without leaks, (So)

Time step (h)	Pressures in Node i	Pressures in Node i+1	.	.	Pressures in Node N-1	Pressures in Node N
0:00	x	x	.	.	x	x
1:00	x	x	.	.	x	x
2:00	x	x	.	.	x	x
3:00	x	x	.	.	x	x
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
22:00	x	x	.	.	x	x
23:00	x	x	.	.	x	x

The pressures obtained in (So) are the base for the comparison with the pressures obtained in each of the modified scenarios when leaks are included. In each node is analysed if in some time step the absolute difference between the pressures of the modified state and the pressures of the initial state are higher than a selected threshold; if this occurs, the leak is defined as detected. The following sections describe the procedure for the selection and addition of the leaks and the evaluation of the new pressures obtained.

4.2.2 Addition of Leak flow (L_f) as extra-demand

In the water distribution network, the user's final demand depends on the base demand (B_d) of the analysed node and its demand pattern. At the moment of adding a leak (L_f), this extra demand would be constant along the day and will not be affected by the demand pattern.

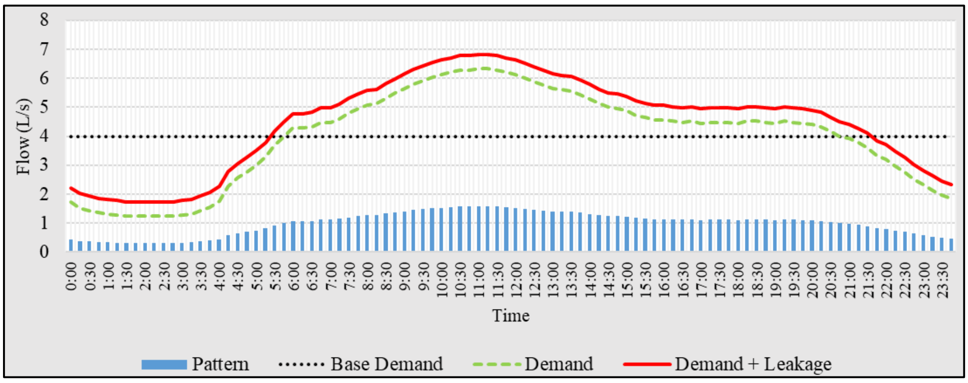


Figure 4.2-1 Final demand behaviour

Figure 4.2-1 illustrates the behaviour of the final demand in a node including leaks. The equation to calculate it is the following:

$$F_d = (B_d \; x \; Pattern \; Value) + L_f \qquad [10]$$

4.2.3 Definition of Leaks Flow Size and Pressure Drop Threshold (Th)

When a leak is added to one node, is expected to cause observable pressure disturbances at other nodes, however, a leak value that always causes observable disturbances regardless of the node where it is added is not always possible, in some cases, there are nodes that will require very high leak values to generate pressure drops higher than the threshold because they are less sensitive to the incorporation of leaks (Pérez et al., 2009). Thus, each of the nodes will require a different leak value to cause pressure drops above a fixed threshold.

To ensure that 100% of the nodes are detected, the leak that generates observable pressure drops at the least sensitive node (worst case scenario) must be evaluated. The problem is that the leak required for this node will probably be very high compared to the leak value required to caused detectable disturbances at other nodes. For this reason, this methodology proposes a procedure to identify a range of leaks that will ensure that they can be identified in at least 20% of the cases[2] ($\boldsymbol{L_{f\,20}}$) and in a maximum of 80% of the cases ($\boldsymbol{L_{f\,80}}$). This is to ensure that the selected leaks have minimum coverage and it is not required to incorporate very high leak values. In addition, this will allow us to evaluate the behaviour of the network under different leak scenarios.

Figure 4.2-2 shows the procedure to obtain the corresponding percentage of cases in which a leak is detected ; since it is challenging to find the exact percentages, it is proposed to generate a number of percentage values between min and max and graphically locate the leaks corresponding to the 20% and 80%.

As a starting point, a threshold of 0.5 mH2O will be selected; additional analysis will be performed using the threshold values of 1.0, 1.5, 2.0, and 3.0 mH2O, because $\boldsymbol{L_{f\,20}}$ and $\boldsymbol{L_{f\,80}}$ values are function of the threshold selected.

[2] 'Case' refers to the fact that the leak causes a drop higher than the threshold in at least one node in the system.

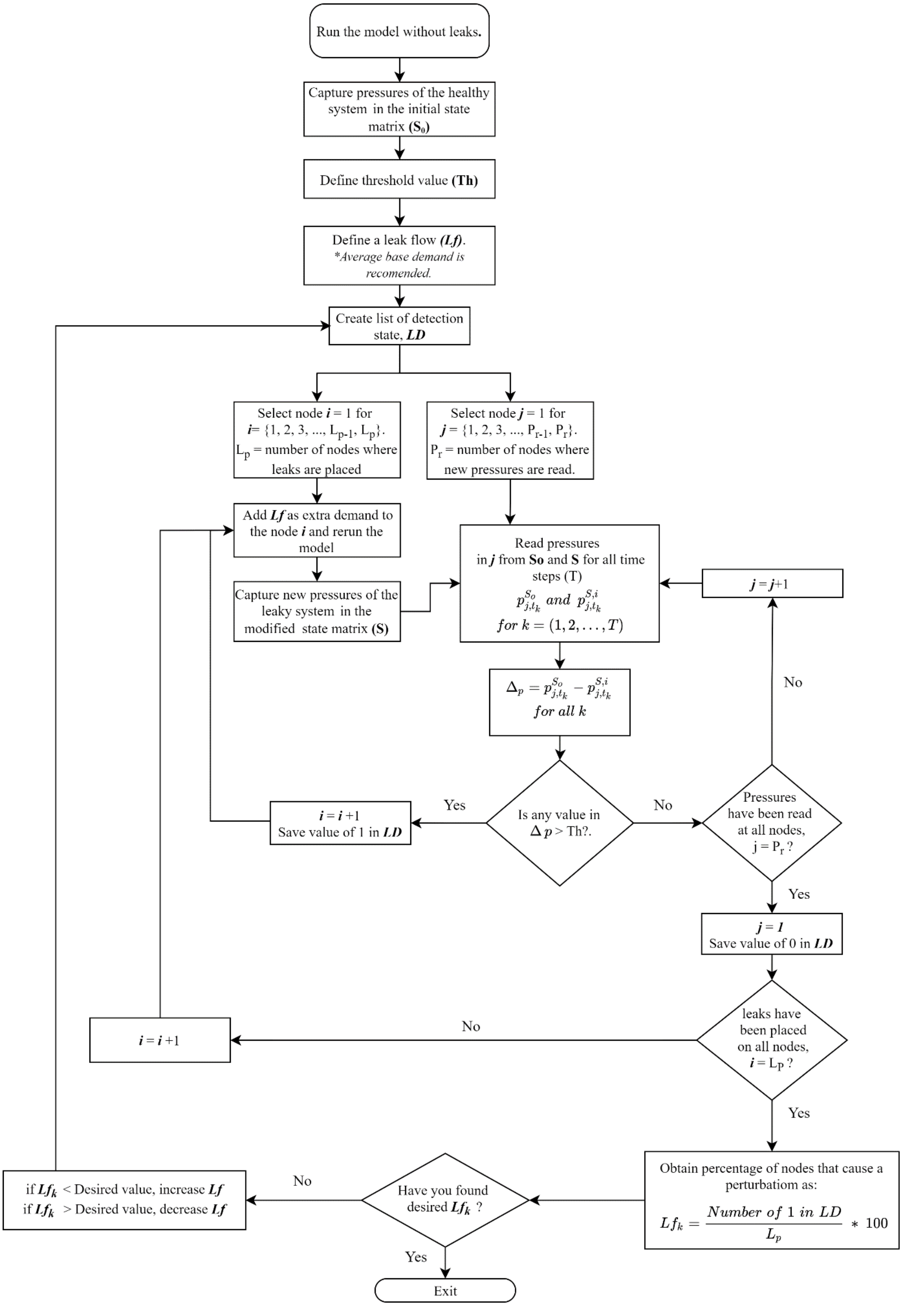


Figure 4.2-2 Flowchart to obtain leak flows (Ls)

Once the $\boldsymbol{L_{f\,20}}$ and $\boldsymbol{L_{f\,80}}$ are calculated, a range of leaks between these two values are created, which will increase by a value ΔL_f obtained as follows:

$$\Delta L_f = \frac{\boldsymbol{L_{f\,80} - L_{f\,20}}}{k-1} \qquad [11]$$

Where:

ΔL_f = increment in leak flow

k = Number of Leaks. (For this methodology, ten leaks were selected, k=10).

4.2.4 Definition of Detection and Non-detection state

Once the leak sizes have been selected, we have to add them one by one to each node, simulating each time that a new leak is added and reading the system's pressure variations **(S)**; these new pressures are compared with the healthy state **(So)** to identify pressure drops higher than the threshold. Every time a node $\boldsymbol{j}$ presents pressure drops higher than the threshold (**Th**) due to the addition of a leak flow ($\boldsymbol{L_f}$) at node $\boldsymbol{i}$, we state that the leak was detected; otherwise, we state that the leak could not be detected.

To store these results, we chose a binary classification in which detection is classified as 1, and non-detection is classified as 0; every node $\boldsymbol{i}$ has therefore a leak detection list ($\boldsymbol{LD_j^i}$) for each node $\boldsymbol{j}$, the length of each list is equal to the number of leaks evaluated (k=10).

Figure 4.2-3 illustrates the procedure to obtain the detection lists in the system.

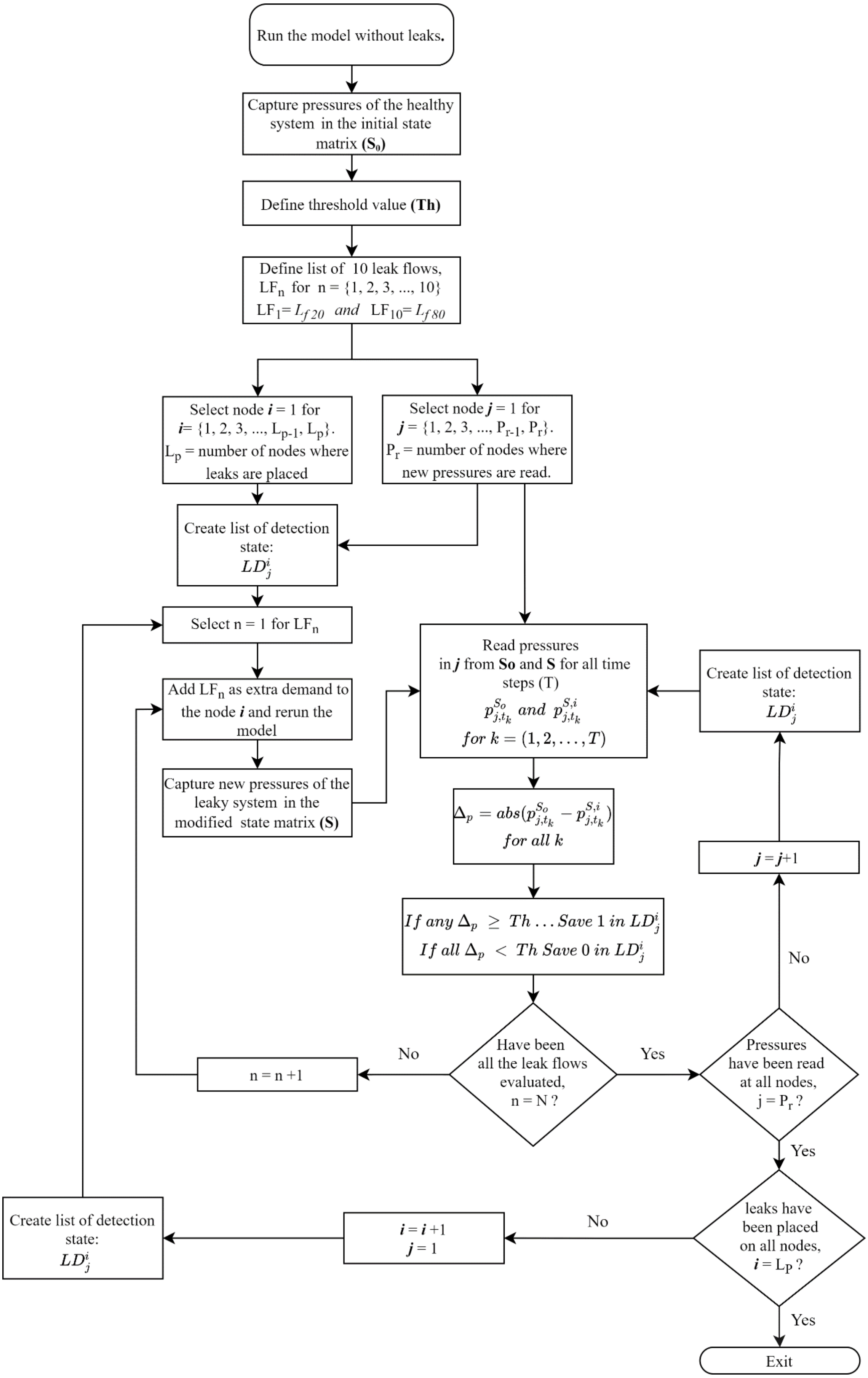


Figure 4.2-3 Flowchart to define detection state lists.

4.2.5 Percentage of Detection

Each time the pressure variation at one node is read, we simulate the 'message' that a sensor installed at the same node would report. Moreover, if we want to know the percentage of detection of a node ***j*** we must find the number in which a value of (1) in the detection lists (***LD***) concerning node ***j*** was reported. We define the probability of leak detection of a node ***j*** as:

$$Pd_j = \frac{\sum_{i=1}^{Lp} \sum_{k=1}^{k} LD_j^i(b_k)}{L_p * k} \qquad [12]$$

Where:

Pd_j: Probability of detection of node ***j***

Lp: Number of nodes where leaks are placed

k: Number of leaks evaluated

LD_j^i: List of detection state for node ***j*** when a leak is placed at node ***i***.

b: binary representation of 0 and 1 for detection and not detection state.

Naturally, the probability of not detection is equal to $(1 - Pd_j)$.

Multiple Percentage of Detection

The above probability is valid for the case of evaluating just one sensor; however, for the case of a group of sensors (***J***), the detection probability of the set must be established differently. In this case, a leak placed at node ***i*** could be detected by the set of nodes ***J***, if any element ***j*** of the set has a pressure drop higher than the threshold.

Since the detection lists (***LD***) of each element ***j*** are established in a binary manner, to obtain the list of detection of a set ***J*** is proposed to use the logical operator disjunction (∨) or the 'or operator' , for the case of two sensors (j_1, j_2) reading the pressure variation caused by adding leaks at node ***i***, the detection list of the set (***J***) is the next:

$$LD_{j_1,j_2}^i = LD_{j_1}^i \lor LD_{j_2}^i \qquad [13]$$

Table 4.2-2 shows the logical operator between the two sensors.

Table 4.2-2 Logical operator for merging lists of detection

$LD_{j_1}^i$	$LD_{j_2}^i$	$LD_{j_1}^i \lor LD_{j_2}^i$
0	0	0
0	1	1
1	0	1
1	1	1

This logical operator returns a false value only if all arguments are false, in our case, if all values are zero '0' and any of the sensor nodes present a pressure drop higher than the threshold.

If more sensors are required, then:

$$LD_J^i = \bigvee_{n=1}^{S_n} LD_{J_n}^i \quad where\ S_n\ is\ the\ number\ of\ sensors \qquad [14]$$

The probability of detection of a set of ***J*** nodes is equal to:

$$Pd_J = \frac{\sum_{i=1}^{Lp} \sum_{k=1}^{k} LD_J^i(b_k)}{L_p * k} \qquad [15]$$

4.2.6 Minimum leak required to obtain observable disturbances (MLR)

The *Minimum Leak Required* (MLR) corresponds to the minimum leak added to node ***i*** to cause a disturbance larger than the threshold at node ***j***. As explained in section 2.2.1, not all leaks generate observable perturbations; because of this, for each pair of nodes ***i***, ***j*** exists a minimum value of leak that causes drop pressures that are higher than the threshold.

These values are saved in an L_p x P_r matrix, where L_p is the list of nodes where the leaks are placed, and P_r is the list of nodes where the pressures are read or where we place a sensor.

Table 4.2-3 Example of the Minimum leak Required Leaks Matrix.

<table>
<tr><th rowspan="2"></th><th rowspan="2">Nodes</th><th colspan="6">Sensor Nodes j</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>.</th><th>.</th><th>P_r</th></tr>
<tr><td rowspan="7">Leak Nodes i</td><td>1</td><td>x</td><td>x</td><td>x</td><td>.</td><td>.</td><td>x</td></tr>
<tr><td>2</td><td>x</td><td colspan="4" rowspan="5">Minimum leak required to obtain a pressure difference larger than the threshold</td><td>x</td></tr>
<tr><td>3</td><td>x</td><td>x</td></tr>
<tr><td>.</td><td>.</td><td>.</td></tr>
<tr><td>.</td><td>.</td><td>.</td></tr>
<tr><td>.</td><td>.</td><td>.</td></tr>
<tr><td>L_p</td><td>x</td><td>x</td><td>x</td><td>.</td><td>.</td><td>x</td></tr>
</table>

For the case of having a set of sensors, the minimum leak corresponds to the minimum leak required between them to identify a leak at node ***i***. If none of the sensors can identify the leak, then a penalty value of 10^6 l/s is assigned, indicating that such node is not able to serve as sensor location.

4.3 Formulation of the Optimisation Problem

The results described in chapter 4.1 are the basis to formulate the objective functions used in the multi-objective optimisation problem, the goal is to locate pressure sensors for the detection of pressure differences and the identification of leaks by maximizing the value of the information (VOI) and the information entropy (IE), In summary, the optimisation problem can be formulated as follows:

Maximize: Z1 =

$$\sum_{i=1}^{Lp} VOI_{i,j} \dots (Pd_j, MLR,\ C_{a,\,s}^{i,j}, \pi_s^i, q_{m,\,s}^{i,j})$$

Maximize: Z2 =

$$IE(\,j\,) \dots (LD_j^i)$$

Subject to:

$$IE(\,j\,) = 0 \quad if\ \forall\, i \;\; LD_j^i = \{\}$$

$$VOI_{i,j} = 0 \quad if \quad Pd_j = 0 \;\; or\ MLR = 10^6$$

Where , *i* is the node where the leak is placed, Lp is the total number of nodes where leaks are placed, the VOI indicates the utility of the information that is sent by a sensor, using factors such as probability of detection (Pd_j), minimum flow required (MLR), prior belief (π_s^i), consequences associated ($C_{a,\,s}^{i,j}$) and conditional probability ($q_{m,\,s}^{i,j}$) and IE refers to the information entropy establishing the diversity that exists in a set of detected nodes, which is an indicator of leak coverage. In this thesis, the concepts of VOI and IE are used as objective functions, in both cases, to maximize quality and coverage, respectively.

In the following sections, the objective functions, decision variables and constraints will be discussed in detail.

4.3.1 Objective Function I, Value of Information (VOI)

The objective of installing pressure sensors in the WDS is to generate alarms when a possibility of leaks in the system exists. A decision-maker will then receive these alarms in the form of a message and will take a respective action/decision in order to reduce losses.

This approach for the installation of sensors involves variables such as system status (leak or no leak), new information (sensor messages), action (decision-maker's choice), and cost of taking a decision, similar to the variables involved in the concept of the value of information (VOI).

As explained in chapter 2.4 the VOI is a concept that groups different quantitative and qualitative variables that helps us to evaluate the quality of the information provided by a sensor about the state of the system. This concept will allow us to classify each node or group of nodes as the best or with the highest potential for leak identification.

According to Alfonso (2010) the VOI concept is defined in terms of three main components:

- The belief that the decision-maker has about the state of the water system before having any information, Prior Belief (π_s)
- The consequences associated with the decision of having to choose among several possible actions given the state of the water system (C_{as}).
- The evaluation and update of new information when it becomes available $f(\pi_s, C_{as}, q_m)$.

The ***prior belief*** (π_s) is associated with the probability that each node has to experience leaks; typically, this probability could be calculated using field data like the node location, road conditions, surrounding trees, pipe diameters, node pressures, pipe age, among other factors. In addition, (π_s) could also be associated with the decision-maker's belief, according to his/her knowledge of the system.

The states defined for each node will be *Leak* and *Not Leak*. Since in this work, we have no field information, and it is assumed that the decision-maker does not have extra information of the nodes' status, the prior belief will be assumed as 50%, representing a total ignorance of the decision-maker about the system state regarding leaks.

Table 4.3-1 Definition of the vector (π_s) for two possible states of a water system.

State (s)		**($\boldsymbol{\pi_s}$)**
s1	Leak	50%
s2	Not leak	50%

The ***conditional probability*** ($q_{m,s}$) is associated with the probability of receiving a message ***m*** given a state ***s***; the message will be the signal emitted by the pressure sensors when a 'Leak' or 'Not leak' event occurs.

In the case of a 'Leak' event a sensor can detect it or not, the probability of detection and no detection is associated with the probability (Pd_J), calculated in section 4.2.5.

In the case of a 'Not leak' event, sensors can emit a false alarm of leaks in the system, as explained in section 2.2.1, in some cases, pressure drops are associated with the normal behaviour of the network and not because of leaks, in this way, detection messages are possible without the presence of leaks.

For practical purposes of the exercise, we assume that 15% of the time, a sensor will send a false alarm, i.e., even if there is no leak in a node, the sensors will send a detection message. Therefore, algebraically the probability of non-detection when there is not leak will be equal to 85%.

The conditional probability ($q_{m,s}$) of a sensor ***j*** or group of sensors ***J*** is calculated as follows:

Table 4.3-2 Definition of conditional probabilities $q_{m,s}$

State	Message	
	Message m1: *Detection*	Message m2: *Non-detection*
Leak	Pd_J	$1 - Pd_J$
Not Leak*	0.15	0.85

* *The probabilities of 15% and 85% will be used for all sensors.*

In Table 4.3-2 we can identify that the messages sent by the sensors are correct when::

- There is a leak condition and the sensor detects it.
- There is no leak and the sensor does not detect.

Similarly, there are two types of error:

- Type I: There is a leak situation and the sensor does not detect it (false negative).
- Type II: There is no leak situation and the sensor detects it (false alarm).

Definition of consequences ($\mathrm{C_{as}}$)

Consequences ($\boldsymbol{C_{as}}$) are defined as the costs associated with taking an action ($\boldsymbol{a}$) given a state ($\boldsymbol{s}$). When a decision-maker receives a warning, he/she has to take a decision. In our methodology, two actions have been defined, namely: a1 (Go to check) and a2 (Do not check)

Table 4.3-3 presents an example of the matrix of costs for node $\boldsymbol{i}$.

Table 4.3-3 Definition of the C_{as} Matrix

State	Action	
	a1: Go to check	a2: Do not check
State (s1) at $\boldsymbol{i}$	cost of doing a1 when s1	cost of doing a2 when s1
State (s2) at $\boldsymbol{i}$	cost of doing a1 when s2	cost of doing a2 when s2

The definition of the costs of each action is not straightforward. They are a general approximation of the expected costs of performing each action and will vary according to the system of analysis. However, the following was discussed and adjusted in an interview with staff of the water utility of Braila.

The criteria used for the selection of each cost are explained below:

State s1 / action a1: this condition implies that there is a leak and it is decided to go to repair; the direct costs will be the reparation costs and the corresponding value of the water lost.

In order to estimate the reparation costs, Braila City personnel was consulted and was indicated that they ranged from 500 to 1000 Romanian Leu ($\approx 100 - 200\ Eur$) depending on pipe diameter, this price includes the materials used to repair the leak. For practical purposes, a cost of 5 (hundreds of Romanian Leu) is assumed.

In order to consider the water losses generated by leaks, the results obtained in Table 4.2-3 are used, (MLR at node $\boldsymbol{i}$ to cause observables disturbances at $\boldsymbol{J}$).

However, the values of the leak flow are not used, in change, the position n of the leak within the 10 leaks analysed is utilized, for n = $\{1, 2, 3, \ldots., 10\}$, The reason is because in this way the methodology can be generalized.

In section 4.2.6 we indicated that the value of 10^6 l/s is used when nodes are not identified. In these cases, the value of 10^6 is used, and as result the VOI is equal to zero (further discussions in result chapter).

Therefore, the costs are defined as:

$$C_{a1,s1}^{J,i} = -5 - n_J^i \quad for \;\; MLR_J^i \leq Lf_{80}$$
$$C_{a1,s1}^{J,i} = -10^6 \; for \;\; MLR_J^i = 10^6 \; l/s \qquad [16]$$

The values have a negative sign because they represent losses.

State s1 / action a2: this condition implies that there is a leak and it is decided do not go to repair, the direct costs will be the corresponding loss of water and the damages associated with the presence of leaks.

In this case, it is difficult to estimate the costs of repairing the damage caused by the leaks, it varies according to the area affected and how long the leak persists, for practical purposes and based in some information given by Braila City personnel, we assume an average reparation cost of 15 (hundreds of Romanian Leu).

For the case of water losses, the same procedure described in the case of **State s1 / action a1** was followed, the only change will be the penalization of do not go when leaks exist, to penalize this situation, a multiplier factor of three (3) was selected.

Therefore, the cost is equal to:

$$C_{a1,s2}^{J,i} = -15 - 3 * n_J^i \quad for \;\; MLR_J^i \leq Lf_{80}$$
$$C_{a1,s2}^{J,i} = -10^6 \;\; for \;\; MLR_J^i = 10^6 \; l/s \qquad [17]$$

State s2 / action a1: this condition implies that there is a no-leak situation, and it is decided to go to repair. The direct costs will include the costs of sending the reparation crew.

In this case, there are no costs for materials, but there are penalties for sending a work team that could be used in another sector where leaks really existed. The penalization cost is estimated as 15 (hundreds of Romanian Leu).

State s2 / action a2: this condition implies that there is no leak, and it is decided do not go to repair. In this case there is no associated cost.

In summary:

Table 4.3-4 Definition of the C_{as} Matrix for Braila case.

State	Action	
	a1: Go to check	a2: Do not check
State (s1) at $\boldsymbol{i}$	$f(MLR) = \begin{cases} -5 - n_J^i, & MLR_J^i \leq Lf_{80} \\ -1.000.000, & MLR_J^i = 1.000.000 \; l/s \end{cases}$	$f(MLR) = \begin{cases} -15 - 3 * n_J^i, & MLR_J^i \leq Lf_{80} \\ -1.000.000, & MLR_J^i = 1.000.000 \; l/s \end{cases}$
State (s2) at $\boldsymbol{i}$	-15	0

Once the parameters of prior belief (π_s), conditional probability ($q_{m,s}$) and the consequences (C_{as}) have been defined, we proceed to follow the flowchart describe in Figure 4.3-1 to calculate the VOI for the group of sensors. An example of this calculation will be discussed in Chapter 5

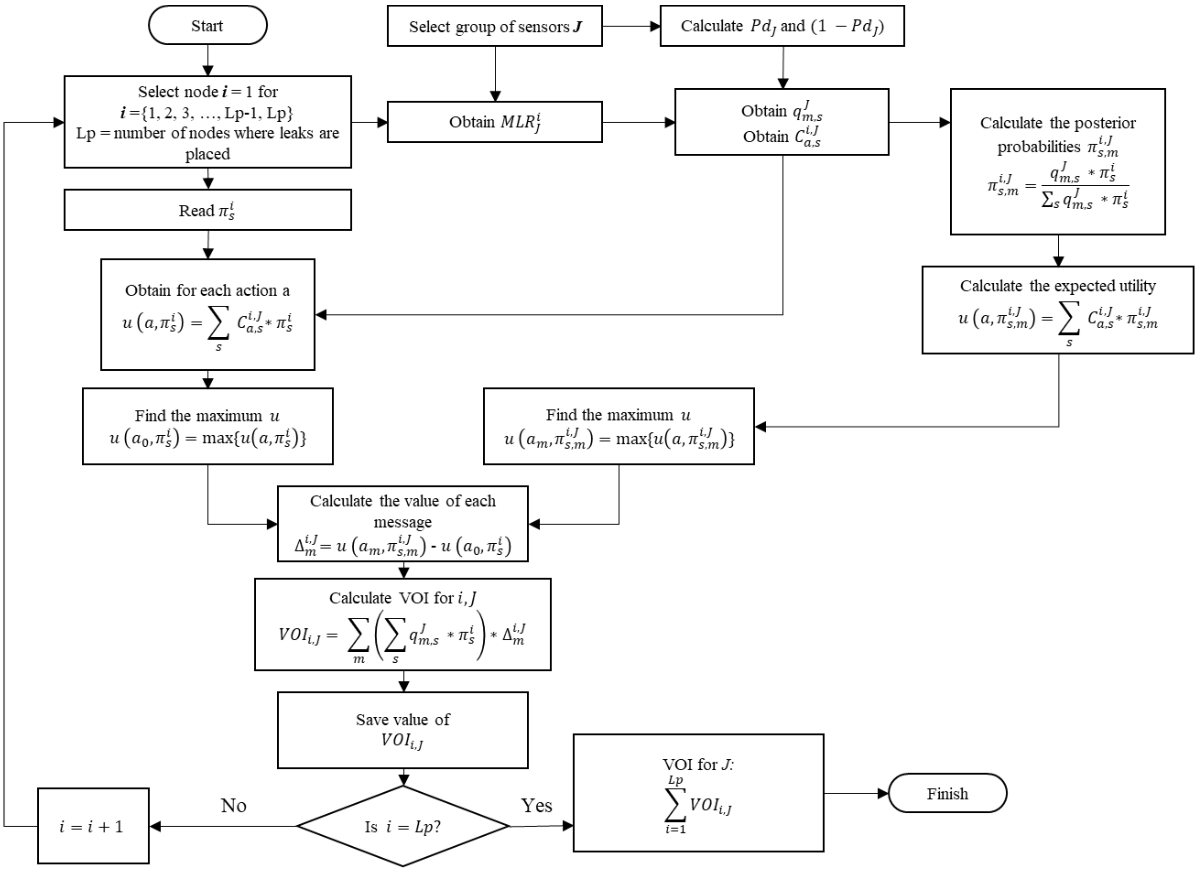


Figure 4.3-1 Flowchart for VOI estimation.

4.3.2 Objective function II, Information Entropy (IE)

In section 4.2.5, it was discussed how to obtain the percentage of detection for a set of sensors. However, it is not considered if a node ***i*** can be detected more than one time by the group of sensors ***J***.

The information entropy is a concept that allows us to quantify the information content that the sensor set can provide. In our case, it is a measure of the set's ability to identify leaks in a larger number of cases. In general terms, the maximum number of times a node is identified as a candidate location for a leak will be equal to k (Number of leaks) x Number of sensors, and it corresponds to the extreme case where all sensors in ***J*** are able to detect all the 10 leaks at node ***i***. This usually does not occur, so it is necessary to quantify how many of the leaks could be identified by a sensor group.

The process to calculate the information entropy of a sensor group ***J*** respecting all the nodes ***i*** is explained below:

1. Select a sensor group ***J***.
2. Select a leak node ***i*** =1, for ***i*** = {1, 2, 3, …, L_{p-1}, L_P} and L_p = number of nodes where leaks are placed.
3. Obtain LD_j^i for each sensor node ***j*** in the set ***J*** (see Figure 4.2-3)
4. Obtain how many times node ***i*** could be detected by sensor group ***J***.

$$D_J^i = \sum_{j=1}^{Sn} \sum_{k=1}^{k=10} LD_j^i(b_k) \qquad [18]$$

Where:

D_J^i: number of times that set ***J*** can detect node ***i***.

LD_j^i: list of detection state for node ***j*** when a leak is placed at node ***i***.

k: number of leaks evaluated (10).

b: binary representation of 0 and 1 for detection and not detection state.

S_n: Number of sensor nodes ***j*** in ***J***.

5. Save the value ***i*** as many times as the value of D_J^i in an list.
6. Repeat steps 3-5 until finish the list of leak nodes.

All the saved values of ***i*** are used for the calculation of the information entropy of the sensor group ***J***, for this the Shannon entropy equation is used:

$$IE(J) = -\sum_{i=1}^{L_p} p(x_i)\log_2 p(x_i) \qquad [19]$$

where $p(x_i)$ is the probability that the value of ***i*** appears in the list of sensors detected by the sensor group. ***J*** (values saved in step 5), a numerical example will be explained in Chapter 5

4.3.3 Decision variables

The decision variable in this methodology is the number and location of nodes where is desired to place a sensor (**Pr**); in general, all the nodes are studied as possible candidates; however, some nodes can be eliminated from the research space because of technical reasons (lack of electricity or pipe diameters) or because they cannot detect any leak in the system.

For the analysis of Radu Negru, all the nodes are candidate locations.

4.3.4 Constraints

The goal of Braila water utility is to install four new sensors in a system where four sensors are already in place; although there are no constraints, it is mandatory to perform two analyses:

- Locate four sensors and compare the results with the sensors already placed.
- Fix the four sensors already installed and find the best extra four sensors location; in total, eight sensors will be analysed.

4.3.5 Algorithm for Optimisation

The Non-Dominated Sorting Genetic Algorithm II (NSGA-II) is used to solve the optimisation problem. This evolutionary algorithm simultaneously optimises multiple objective functions and ranks sets of decision variables based on dominance to find non-dominated solutions (Deb et al., 2002). The final result is to find the optimal global results as a Pareto-front curve/surface

Considering the installation of four sensors in a network of 305 nodes, our search space is the next:

$$\frac{n!}{n!-r!} = \frac{305!}{305!-4!} = 305 * 304 * 303 * 302 \cong 10^{10}\, combinations$$

Due to the size of our search space and due to time limitations, we considered to use a number of 10^4 generations and a population size of 10. Accepting that only 10^5 out 10^{10} of the search space can be explored. For future research, it could be improved by running more generations.

Table 4.3-5 shows the parameters used:

Table 4.3-5 Parameters for NSGA-II algorithm

Parameter	Value
Population size	10
Generations	10.000
Mutation rate	0.1
Crossover rate	0.9

4.4 Setting Up the Algorithm

To automatize the process of adding artificial leaks, calculate probabilities, obtain objective functions, perform multi-objective optimisation, among other steps, we develop an algorithm using the programming language Python.

In Table 4.4-1 is shown the characteristics where all the analysis was performed

Table 4.4-1 settings for the algorithm.

Parameter	Characteristics
Programming language	Python 3.7.9
IDE	Spyder 4.0
Ram	24 Gigabytes
Processor	Intel I5 – 4790
Operational System	Windows-10

The algorithm is composed by own functions and special libraries designed by external investigators, below we describe the libraries used:

4.4.1 Python libraries

Water Network Tool for Resilience (WNTR)

There are plenty of tools that allow us to simulate and analyse the water distribution networks, one of the best known and widely used is EPANET which presents a graphical interface that facilitates the visualization of the elements (links, nodes, pumps, valves, reservoirs, etc.) and their results. On the other hand, with the purpose of using these tools with greater versatility, API's (application programming interfaces) have been created to allow simultaneous changes in the network, multiple simulations with structural and parameter changes, real-time simulations or fictitious scenarios of accidents with recovery actions.

This is the case of WNTR, which is a package developed for Python with all the necessary tools for the implementation of our simulation. One of the advantages of using this package is the possibility of using its algorithm for the resolution of pressure networks using a pressure-driven approach, the required libraries and the whole installation process can be found at:

https://wntr.readthedocs.io/en/latest/index.html

Platypus

This tool, also developed for Python, allows us to formulate the optimisation problem, with the main advantage of presenting algorithms and analysis tools for multi-objective optimisation scenarios (Hadka, 2015). NSGA-II was the selected algorithm due to the results obtained by previous researchers and their good performance in localizing local solutions within a global space. Its configuration was performed using the guidelines presented in:

https://platypus.readthedocs.io/en/latest/index.html

4.4.2 Epanet model

To utilize the tool WNTR is required to create a file in format. inp using the tool EPANET, the file configuration has to be the next:

- The file has to be saved using EPANET v2.2.
- Flow units = LPS
- Total duration = 24 Hours
- Hydraulic time step = 1 Hour
- Demand Model = PDA
- Minimum pressure = Under this value, the demand is not supplied. For Braila city, this value is 5 mH2O
- Required pressure = Above this value, the demand is normally supplied. For Braila city, this value is 15 mH2O

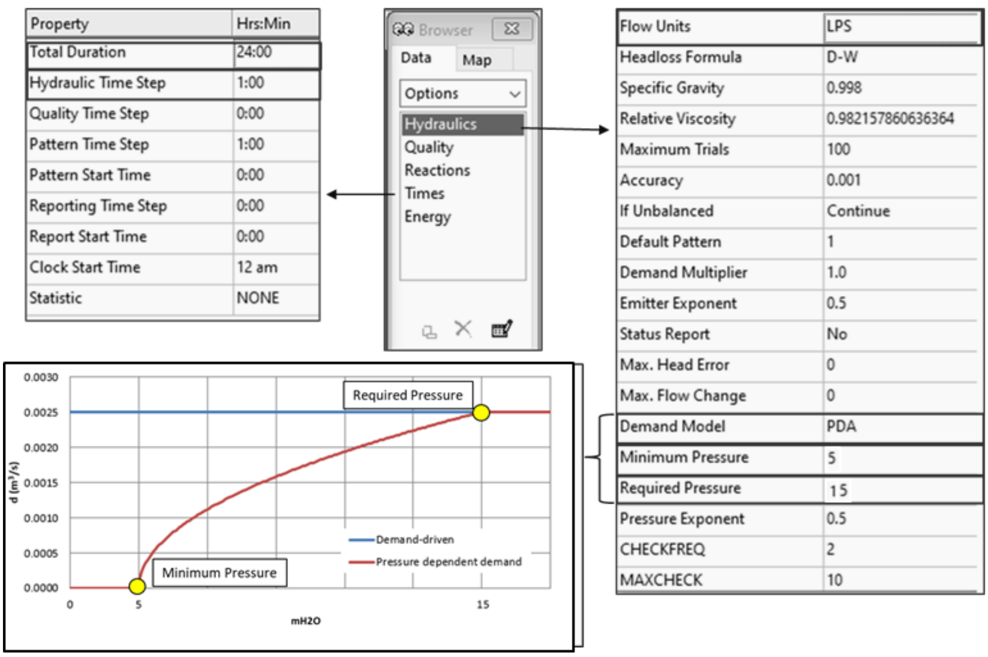


Figure 4.4-1 Configuration of the .inp file before being exported from EPANET

4.4.3 Algorithm

A detailed description of the developed algorithm is given in **(APPENDIX B),** specifying each of the parameters used, in Table 4.4-2 the list of parameters used for the initial simulation is shown:

Table 4.4-2 Parameters utilized to obtain final results

Parameter	**Value**
Daily flow (l/s)	40
Number of simulated leaks (10 per node)	3,050
Number of demand nodes	123
Objective Function I	VOI
Objective function II	IE
Decision variables	Number and location of nodes
Leak magnitude	list of 10 leaks according to characteristics of the system
Pressure threshold	0.5, 1, 1.5, 2.5, 3 mH2O
Eligible meter locations	305
Time step simulated	24 hours – 1 hour time step

Chapter 5 Results and Discussions

This chapter presents all the results and findings obtained from applying the method explained in chapter 4. Firstly, the leaks evaluated are obtained, once they have been added to the model and realize the respective simulations, the pressure deviation matrices are used to calculate the detection times, minimum flows required, and detection percentages of each node. Previously these results will be used to calculate the objective functions VOI and IE. The validity of these results and their functionality with respect to the objectives of installing pressure sensors for the identification of leaks will be discussed.

Finally, the different scenarios resulting from the optimisation process are shown, comparisons between the proposed sensor network and the currently installed sensor network are presented. Recommendations and analysis are provided for each of the steps.

5.1 Results

5.1.1 Preliminary analysis for detection, distribution and sizes of leaks

Leaks Evaluated

Before adding the leaks in each of the nodes, it is essential to clarify that the hydraulic model of the Radu Negru sector is composed of nodes where leaks should not be added. These nodes correspond to fictitious nodes which are required to incorporate pumping equipment, as a decision of the operators in Braila. These points do not exist in reality but are required in the EPANET model for calibration purposes. In total, there are four pumps, and each pump is composed of two fictitious nodes; therefore, eight nodes will be removed from the analysis space, leaving a total of 297 nodes for analysis.

Using a threshold of 0.5 mH2O and according to the process explained in section 4.2.3, a list of leaks from 1 l/s to 40 l/s were initially analysed to define the ($\boldsymbol{L_{f\,20}}$) and ($\boldsymbol{L_{f\,80}}$) respectively, the selection of 40 l/s is because this is the total daily flow for the Radu Negru's network, then the extreme situation in which the system loses all its water is simulated, the time of simulation was one day to explore all the values of the demand pattern, with a time step of 1 hour, the demand model used was PDA (Pressure dependent analysis)

Figure 5.1-1 shows the selection of the two leak points, and it is concluded that a leak of 4 l/s can be identified in 20% of the nodes, which means that approx. 59 out 297 nodes could be detected when a leak of 4 l/s was added to each of them, generating pressure drops greater than the threshold in any part of the network.

Similarly, a maximum leak of 20 l/s was selected because it was detected in approx. the 80% of the cases (238 out 297 nodes), is remarkable to see that a leak of 40 l/s was identified in 100% of the cases, concluding that the least sensitive node of Radu Negru's network requires a leak of 40 l/s to generate pressures drops above the threshold of 0.5 mH2O.

If we compare the flow rate required to identify 100% of the nodes (40 l/s) with the flow rate required to identify 80% of the nodes (20 l/s) we find that we would need to increase leaks by almost 100% to obtain an increase in coverage of 20%. This is the main reason why a percentage of 80% is selected, the leak flow required to cause pressure drops higher than this percentage could be very large and do not show a reasonable or justified improvement

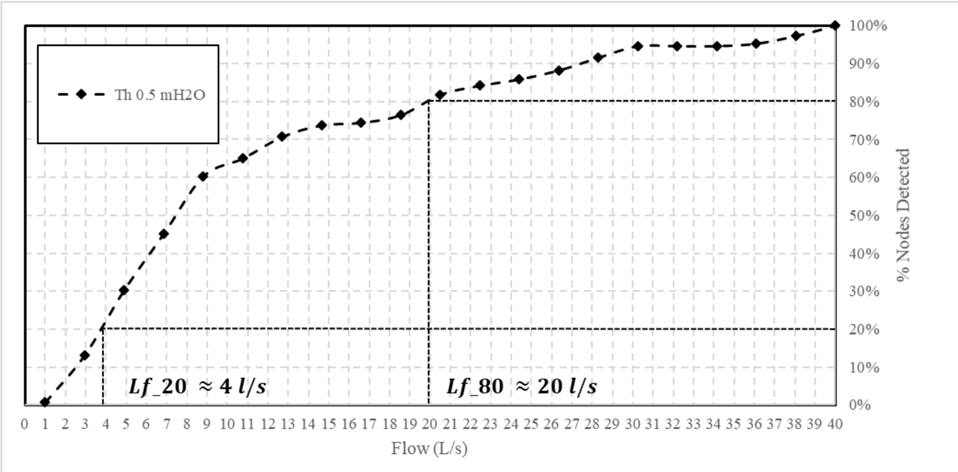


Figure 5.1-1 Relation between leak values and % of nodes detected, Th = 0.5 mH2O.

With the minimum and maximum leaks identified, a set of 10 (ten) leaks was established, using equation [11] the ΔL_f is equal to:

$$\Delta L_f = \frac{20-4}{10-1} = 1.78\frac{l}{s}$$

Furthermore, the set of leaks for a threshold of 0.5 mH2O is equal to:

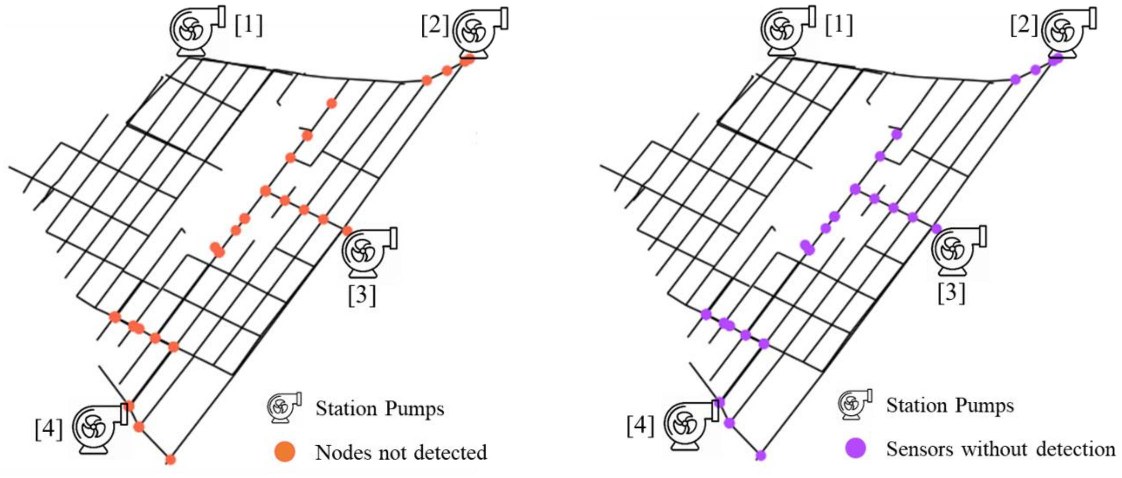
$$Lf = \{4, 5.8, 7.6, 9.3, 11.1, 12.9, 14.7, 16.4, 18.2, 20\}\ l/s$$

This set of leaks will be included in each of the nodes following the procedure presented in Figure 4.2-3.

Detection and Non-Detection Results

With the leaks evaluated, a total of 57 nodes were not detected by any point of the network. Figure 5.1-2 (a) shows where these nodes are located; this is caused because the nodes are close to the station pumps, so generating pressure drops in the surrounding points requires higher leaks than the maximum leak evaluated, additionally, each of the pumps has sufficient capacity to supply the extra flow required by the network due to leaks. On the other hand, 54 nodes never identified leaks in any part of the network (Figure 5.1-2, b), meaning that their pressures never drop above the threshold value, the reasons is because they are located very close to the primary source of energy (Pumps), since losses are a function of the distance, all the points close to the energy source will have lower losses than the rest of the network, and the pressures do not drop below the threshold.

The points located near pump number [1], were identified because this pump has the lowest capacity of the four pumps installed, so all the nearby points are more susceptible to pressure drops.



(a) (b)

Figure 5.1-2 (a) Map with nodes not detected, (b) Map of sensors without detection.

To save time, all nodes not detected will be removed from the search space. Similarly, nodes, where the sensors did not detect leaks will not be included in the analysis.

In terms of detection, Figure 5.1-3 illustrates the map of the percentage of detection, it is clear that the west zone of the network presents the higher percentage of detection; this is again because there are no pumping stations in this area and generating pressure drops at these nodes is easier compared to other zones.

The node with the highest percentage of detection has a value of 35%, contrary, some nodes could not detect any leak, presenting 0% of detection.

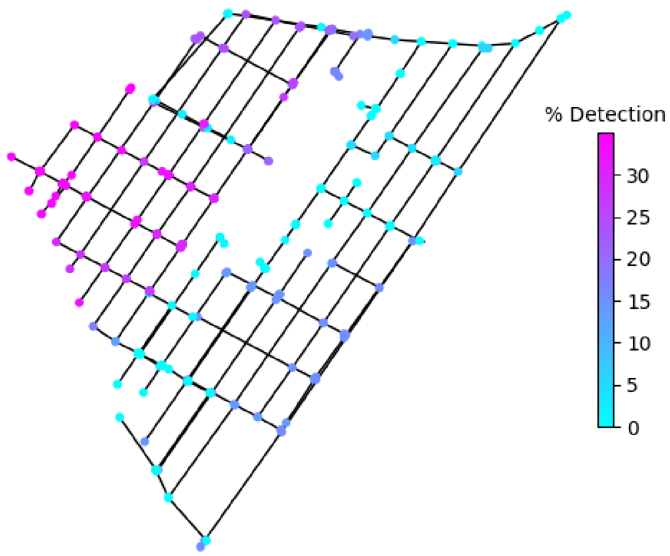


Figure 5.1-3 Map with the percentage of detection.

In section 4.2.3, we discussed the sensitivity of the nodes, indicating that each node reacts differently to the system perturbations. If we compare how is the variation in pressures of a node with a high percentage of detection against a node with a low percentage of detection, we can see in Figure 5.1-4 for the case (a) the pressures drop for the majority of perturbations, and in many cases above the threshold of 0.5 mH2O. On the other hand, case (b) shows how the pressures remain constant in most cases, and just some nodes in the system can cause considerable pressure drops.

The lowest pressure registered between these two nodes is close to 5.5 mH2O. This is due to the restriction of 5 mH2O used in the pressure-driven analysis, where demands are not supplied under these pressure values.

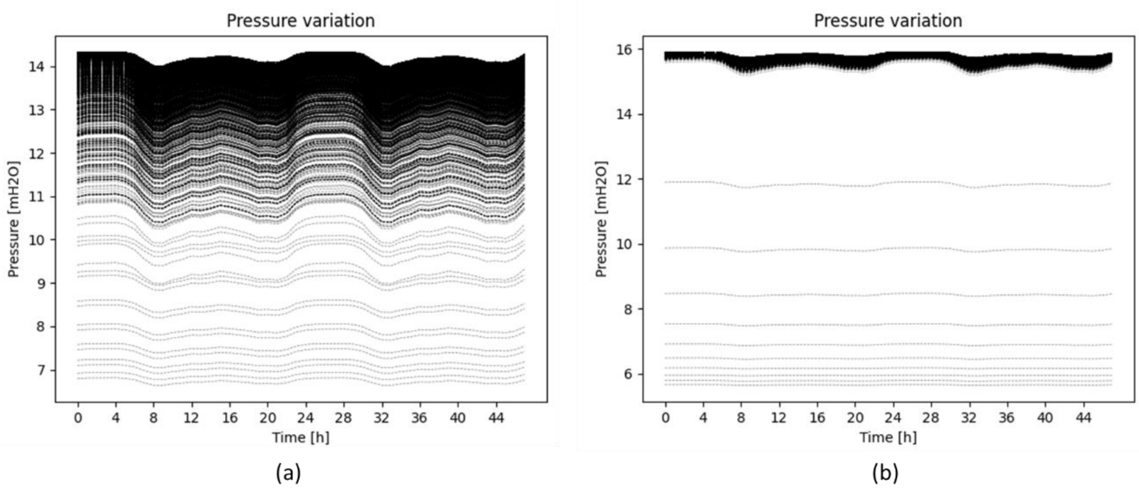


Figure 5.1-4 Variation of pressure for nodes with high (a) and low (b) percentage of detection.

Time of Detection

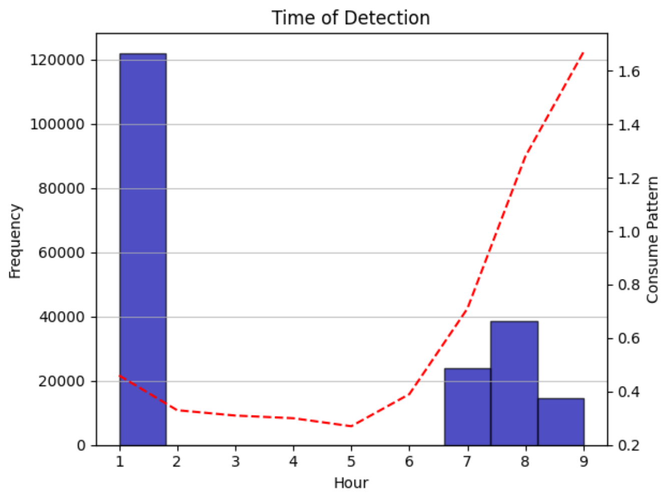


Figure 5.1-5 Times of detection for all the scenarios.

According to Figure 5.1-5 most leaks were detected at zero hours of modelling, this indicates that the leaks were sufficiently large to cause a pressure drop at some point in the system at the first moment of the simulation and does not have much to do with the consumption behaviour of the system. In addition, the leaks that were detected between 7:00 am and 9:00 am correspond precisely to the times when the hours of high consumption start and the demands are affected by the multipliers of the consumption factor, generating higher demands and consequently higher pressure drops. The particularity of this time range is that the demand multiplying factor is higher than the one presented at the zero hour, for this reason leaks are detected.

As a conclusion, any time a leak is not detected in a time step t_i, it can be detected until the next time-step t_{i+n} that presents a multiplying factor greater than the one presented in t_i, for this reason, using the detection time as a parameter to evaluate the performance of a sensor with respect to leak identification is not very effective, since only the hours with high demand multipliers will have an effect on the results. For this reason, it is recommended to perform the simulation of leaks in hours where the demand factor is lower (worst case), in this way we can guarantee that the leak can be detected at any time of the day.

Distribution of Minimum Leaks Required

Figure 5.1-6 shows how the majority of points could be identified between 4 and 10 l/s, the nodes close to the pumping stations required higher leak to be identified, again showing the influence of the pumping stations in the generation of pressure drops.

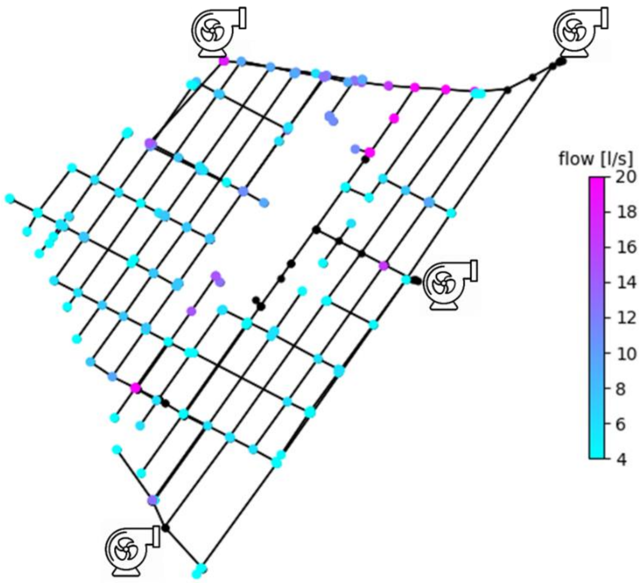
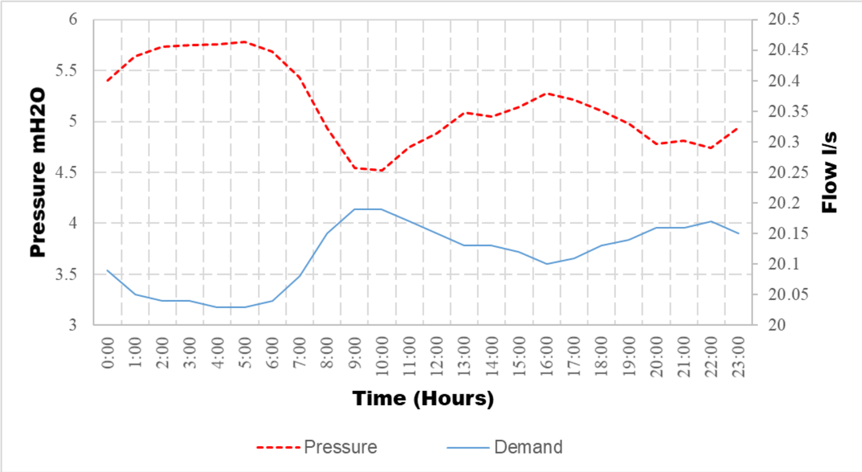
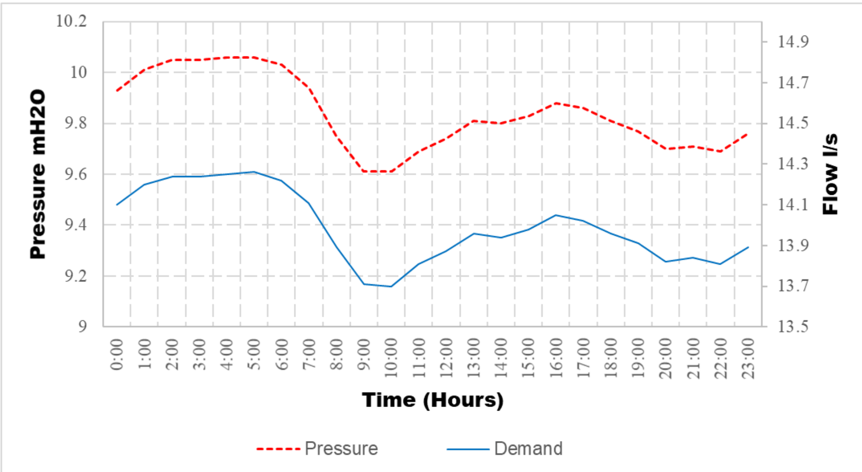


Figure 5.1-6 Minimum leak required (MLR) for leak detected.

To illustrate the impact of the use of Pressure-Driven analysis PDA, is shown in Figure 5.1-7 the difference between the demand distribution of a node simulated using DDA and PDA, for this simulation an extra leak of 20 l/s was included in the node where pressure were read, as we can see, for the case of DDA (a) the system could supply the demand of the node reaching at demand of 20.3 l/s at 10:00 a.m. (20 leak flow + 0.3 base demand) even when the pressures are below the minimal pressure required in the system (5 mH2O), for the PDA case (b) the demand in the same node dropped to a value of 16.5 l/s, and this is because the system pressures were below the required pressure and the demands could not be supplied, in this case, the results represented in a more accurate way the scenario.



(a)



(b)

Figure 5.1-7 Variation of the distribution of demand at node J-1874 using a Demand-driven analysis (a), and a Pressure-Driven analysis (b).

5.1.2 Numerical example of VOI and Information Entropy Results

Following the flowchart presented in Figure 4.3-1, a numerical example of the procedure to calculate VOI for a sensor group ***J*** identifying leaks at node ***i*** is presented.

In a previous section we estimated that the set of leaks to be evaluated were:

$$Lf = \{4, 5.8, 7.6, 9.3, 11.1, 12.9, 14.7, 16.4, 18.2, 20\}\ l/s$$

The prior belief of presence of leaks at node ***i*** is assumed as 50%, then the π_s^i is equal to:

Table 5.1-1 vector (π_s)

State (s)		(π_s)
s1	Leak	50%
s2	Not leak	50%

$$\pi_s^i = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

Assuming that the probability of detection for sensor group ***J*** in the whole system was 35%, and according to Table 4.3-2, the values of $q^{J}_{m,s}$ is equal to:

Table 5.1-2 Conditional probabilities $q_{m,s}$

State	*Message*	
	Detection	Non-detection
Leak	0.35	0.65
Not Leak	0.15	0.85

$$q^{J}_{m,s} = \begin{bmatrix} 0.35 & 0.65 \\ 0.15 & 0.85 \end{bmatrix}$$

The minimum leak required (MLR) to identify a leak at node ***i*** is assumed as 12.9 l/s, it corresponds to the number 6 in Lf, using the equations in Table 4.3-4, the matrix of consequences $C^{i,J}_{a,s}$ is:

Table 5.1-3 C_{as} Matrix

State	*Action*	
	Go to check	Do not check
Leak	$-5-6=-11$	$-15-3*6=-33$
Non Leak	-15	0

$$C^{i,J}_{a,s} = \begin{bmatrix} -11 & -33 \\ -15 & 0 \end{bmatrix}$$

According to the prior belief and the cost matrix, the utility of each action is calculated using equation [3] :

$$u\left(a, \pi^{i}_{s}\right) = \sum_{s} C^{i,J}_{a,\,s} * \pi^{i}_{s} = \begin{bmatrix} -11 & -33 \\ -15 & 0 \end{bmatrix}^{T} \cdot \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = [-13.0 \quad -16.5]$$

To find the maximum u, (the action that would have been chosen without information):

$$u\left(a_0, \pi^{i}_{s}\right) = \max\left\{u\left(a, \pi^{i}_{s}\right)\right\} = \max\{[-13.0 \quad -16.5]\} = [-13]$$

According to this result, the action that the decision-maker will select using his/her prior belief of the system is ‘Go to check’, this decision is related to the one which generates the highest profit, in this case, the decision which generates the minimum loss.

In this stage, the decision-maker evaluates if is valuable to receive new information from the sensors. First, he/she has to update the prior beliefs ($\pi^{i,J}_{s,m}$), using the messages sent by the sensors:

$$\pi_{s,m}^{i,J} = \frac{q_{m,\,s}^{J} \; * \pi_s^i}{\sum_s q_{m,\,s}^{J} \; * \pi_s^i} = \begin{bmatrix} \frac{0.35*0.5}{0.35*0.5+0.15*0.5} & \frac{0.65*0.5}{0.65*0.5+0.85*0.5} \\ \frac{0.15*0.5}{0.35*0.5+0.15*0.5} & \frac{0.85*0.5}{0.65*0.5+0.85*0.5} \end{bmatrix} = \begin{bmatrix} 0.7 & 0.43 \\ 0.3 & 0.57 \end{bmatrix}$$

Table 5.1-4 Posterior probabilities $\pi_{s,m}^{i,J}$

State	*Message Detected (by J)*	*Message Not detected (by J)*
Leak	0.7	0.43
Not leak	0.3	0.57

Table 5.1-4 can be interpreted as follow, initially, a belief of 50% was assumed for the leak state of node ***i***, once the sensors are placed, this belief increases to 70% if the sensors send a message of ‘Detected’, in contrast, the belief is reduced to 43% if the sensors send a message of ‘Not detected’.

Similarly, the belief of 50% for Non-leak state is reduced to 30% when the sensors send a message of ‘Detected’. On the contrary, if sensors do not send any message, the belief is reinforced to 57%.

The possible sensor’s messages generate two scenarios, one in which the sensors send a message of ‘Detected’ and another in which they send a message of ‘Not detected’, for each of them, the new utilities have to be obtained.

First, the expected utility of each message is obtained using the equation [6]:

$$u\left(a, \pi_{s,m}^{i}\right) = \sum_{s} C_{a,\,s}^{i,J} * \pi_{s,m}^{i,J} = \begin{bmatrix} -11 & -33 \\ -15 & 0 \end{bmatrix}^T \cdot \begin{bmatrix} 0.7 & 0.43 \\ 0.3 & 0.57 \end{bmatrix} = \begin{bmatrix} -12.20 & -13.27 \\ -23.10 & -14.30 \end{bmatrix}$$

For each scenario, the decision-maker will choose the action that gives the maximum utility, which could be calculated using equation [7]:

$$u\left(a_m, \pi_{s,m}^{i,J}\right) = \max\left\{u\left(a, \pi_{s,m}^{i,J}\right)\right\} = \max\left\{\begin{bmatrix} -12.20 & -13.27 \\ -23.10 & -14.30 \end{bmatrix}\right\} = \begin{bmatrix} -12.20 & -13.27 \end{bmatrix}$$

Table 5.1-5 Utilities of new decisions, $u\left(a_m, \pi_{s,m}^{i,J}\right)$.

Action	*Message Detected (by J)*	*Message Not detected (by J)*
Go to check	−12.20	−13.27
Do not check	−23.10	−14.30
Max.	**-12.20**	**-13.27**

According to Table 5.1-5, if the sensors ***J*** send a message of ‘Detected’, the best option is ‘Go to check’. This is because in this scenario the addition of sensors increases the prior belief to 70% for leak existence. The decision ‘Go to check’ generates lower losses compared to the decision of ‘Do not check’.

In the case when sensors ***J*** do not send any message, the best decision is also ‘Go to check’, and this is mainly because of two reasons; first, although in this scenario the leak perception is reduced to 43%, the cost of ‘Go to check’ is still lower than the cost of ‘Do not check’, second, the message comes from a sensor that 65% of the times reject the true state (there is a leak and the sensors did not detected), then, for the decision-maker is still very possible that the leak exists, leading to the decision of ‘Go to check’ .

After defining the new utilities, is required to determine the value of each message, for this, the new utilities should be compared with the utility of the action that would have been chosen before additional information (a_0):

$$\Delta_m^{i,J} = u\left(a_m, \pi_{s,m}^{i,J}\right) - u\left(a_0, \pi_s^i\right) = [-12.20 \quad -13.27] - [-13] = [0.80 \quad -0.27]$$

The utility is increased (0.80) if a ‘Detected’ message is received, this is because the new decision is in line with the initial decision ‘Go to check’, and the belief of leak is increased. On the other hand, receiving a ‘Not detected’ message reduces the belief of leak, but the decision is still ‘Go to check’, this new information only generates losses because the decision-maker does not change his/her initial perception.

Finally, to evaluate if placing sensor ***J*** in the system brings valuable information detecting leaks at node ***i***, the VOI has to be calculated using the values of each message.

Using equation [8] the VOI for $i,\ J$ is equal to:

$$VOI_{i,J} = \sum_m \left(\sum_s q^{\,J}_{\,m,\,s} * \pi_s^i \right) * \Delta_m^{i,J}$$

$$VOI_{i,J} = \left[\begin{bmatrix} 0.35 & 0.65 \\ 0.15 & 0.85 \end{bmatrix}^T \cdot \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right]^T \cdot [0.80 \quad -0.27]^T$$

$$= (0.35 * 0.5 + 0.15 * 0.5) * 0.80 + (0.65 * 0.5 + 0.85 * 0.5) * 0.27 \approx 0$$

For this case, the value of the information is equal to zero, and it is due to two reasons; the first one is that the probability of detection of the sensors is very low (35%), and in leak scenarios, this new information does not help to reduce the initial uncertainty of 50% of the existence of leaks, the decision-maker cannot rely on external information that has a lower probability of accuracy than his/her own belief. Second, the leak value required to detect node ***i*** is very high, causing that the initial decision will be ‘Go to check’ because the losses associated with this action are lower than those caused if ‘Do not check’ is selected. At the moment of receiving new information from the sensors, the decision to be taken is in line with his/her initial decision (without extra information), so for the decision maker this new information has a minimal value.

The same exercise is repeated, changing the prior belief of the decision-maker to 40% probability of ‘Leak’ state, in this scenario, sensors obtain a VOI of 1.73 and this occurs for two reasons, the first one is because now his/her belief (40%) is quite similar to the detection probability of leaks (35%). Therefore, part of this new information improves their initial perception. Secondly and most important is because the decision that should have been adopted without receiving any information was ‘Do not check’, because in this case, it is more probable the ‘Not leak’ state (60%) and the costs associated with ‘Do not check’ are lower than the costs associated with ‘Go to check’, However, once new information of the leak status is received and sensors indicate ‘Detected’, the initial decision changes to ‘Go to check’, so, in this case the information is more valuable. In Table 5.1-3 we can see that the cost of 'Do not check' in the state of 'No Leak' is zero and is only penalized if in reality a leak exists, so, in some cases doing nothing generates profits or in the case of this example, minimize losses.

In this exercise, the decision-makers prior belief of 'Leak' state changed from 50% to 40% and his/her decision changed completely, indicating that the outcome of the VOI is very sensitive to the prior belief parameter, Figure 5.1-8 shows the decision graph based on the prior belief and the costs associated with each action, it indicates that if the decision-maker considers that there is a 'Leak' state by more than about 41%, the best decision is 'Go to check'. (This result is obtained using the matrix costs shown in Table 5.1-3).

In practice, some nodes will always be more susceptible to leaks, but a change from 50% (total ignorance) to 60% (a slight belief that leaks exist) is an easy option, because for the initial perception exist a range of appreciation and also it is subjective,

so, allowing a parameter with such high uncertainty to be the one that most influences the initial decision is a problem that exists in this methodology.

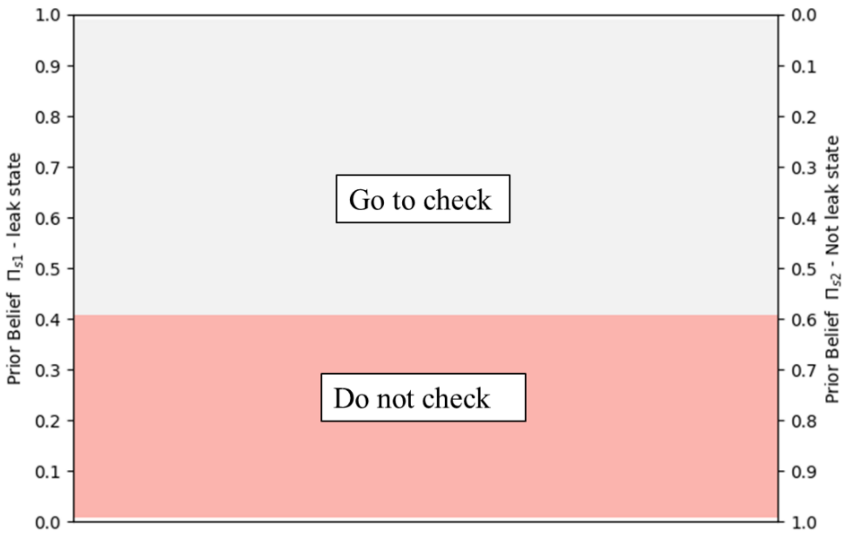
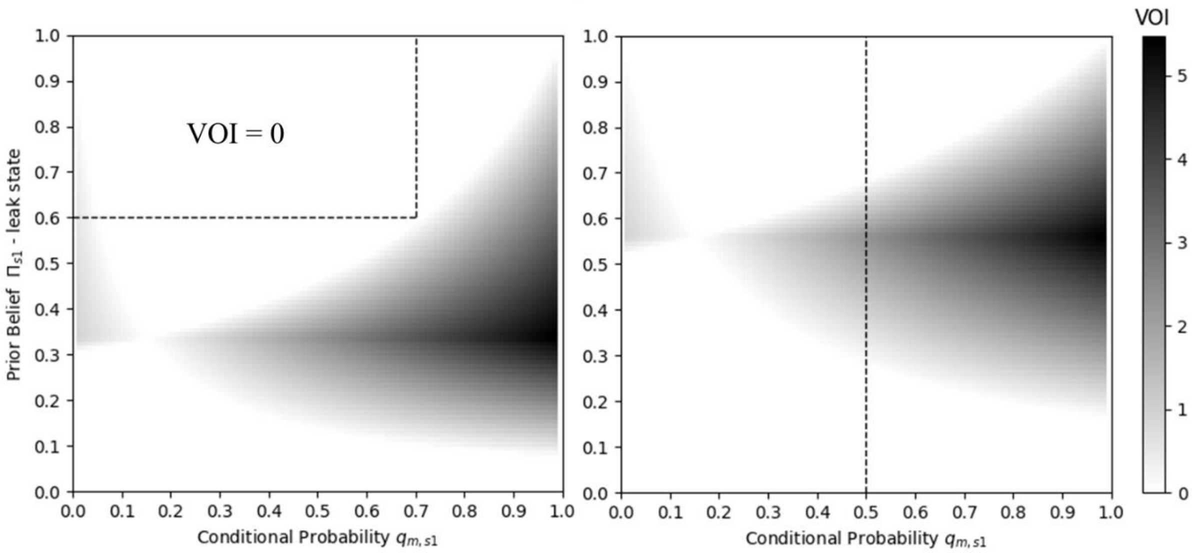


Figure 5.1-8 Decision graph changing prior belief π_s.

To analyse the variation of the VOI when using different values of prior belief and detection probabilities, we repeated two times the previous exercise, for both cases, it was assumed that the type II error or false alarm remains at 15% ($q_{m,s2} = [0.15\ \ 0.85]$), for the first case, it was assumed that all nodes were detected using leak number 10 (worst case scenario). In the second case, it was assumed leak number 1 (best case scenario), for both cases, the cost matrix was calculated using the equations shown in Table 4.3-4.

For case 1, Figure 5.1-9 (a) shows that if the detection probability of the sensors is less than 70% (considerably high), it will not provide information to the decision-maker if he/she considers a prior belief of 'Leak' state of 60% or more, again reinforcing the fact that personal opinion takes precedence over the information received.

Figure 5.1-9 (b) indicates that in cases when the detection probability for a group of sensors is less than 50%, the information is provided just in few scenarios, and is very low compared to the nodes that have higher detection percentages, so only nodes with high percentages of detection will be selected, this could be useful if we want to bias the groups of sensors with higher accuracy, but there would be problems in cases where any group of sensors can detect more than 50%



(a) (b)

Figure 5.1-9 Variation of the Value of Information when changing the prior probability π_s and the conditional probabilities $q_{m,s}$ for the consequence matrix shown in Table 5.1-3 , (a) n leak position = 10, (b) n leak position = 1.

The previous analysis shows how important is the prior belief of the decision-maker, and how his/her perception is hardly influenced by information that does not have good reliability, in this case by sensors that do not detect a large number of leaks; however if the VOI is only helpful for groups of sensors with high detection probabilities, it is difficult to generalize the methodology, because there will be cases in which the groups of sensors cannot detect a higher number of leaks, but they are the best solution among all possible groups of sensors.

For the Radu Negru WDS, the node with the highest percentage of detection was 35%. Figure 5.1-10 shows that just the nodes with a higher percentage of detection bring some valuable information. However, practically none of the nodes provide information, since the prior belief of ‘Leak’ state is 50%, much higher than the detection probability of the sensors.

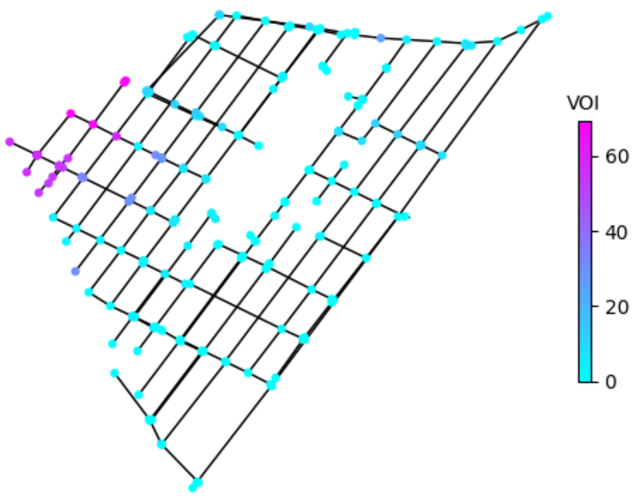


Figure 5.1-10 Results for VOI calculation, Th = 0.5 mH2O.

Because the VOI is highly dependent on the prior belief and the information from the sensors sometimes cannot be used, it is defined that the proposed formulation of the VOI concept is not adequate for the identification of the best set of sensors. For this reason, a new objective function must be proposed and this will be discussed further in section 5.1.3.

For the case of the information entropy in Table 5.1-6 is shown an example of calculation for a hypothetical situation; for this example, four sensors were considered in a network of 10 nodes. The number of leaks evaluated was 10; then, the maximum number of times a node can be detected by the sensor group is equal to 40 (4 leaks x 10 sensor nodes). In this case the information entropy for the sensors group is equal to 3.21 bits.

Table 5.1-6 Numerical example of information entropy.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sum. Times detected	Probability $p(x_i)$	IE (bits) $p(x_i)\log_2 p(x_i)$
	Number of leaks detected						
Node 1	0	1	0	1	2	0.02	0.13
Node 2	4	1	0	1	6	0.07	0.27
Node 3	5	3	0	1	9	0.10	0.34
Node 4	3	0	3	4	10	0.11	0.36
Node 5	0	0	4	6	10	0.11	0.36
Node 6	0	0	6	2	8	0.09	0.32
Node 7	0	5	6	2	13	0.15	0.41
Node 8	8	4	2	0	14	0.16	0.42
Node 9	4	4	0	0	8	0.09	0.32
Node 10	5	2	0	0	7	0.08	0.29
Sum. Total					87	1.00	**3.21**

A similar process was realized using the results of the model for Radu Negru and utilizing the steps described in section 4.3.2.

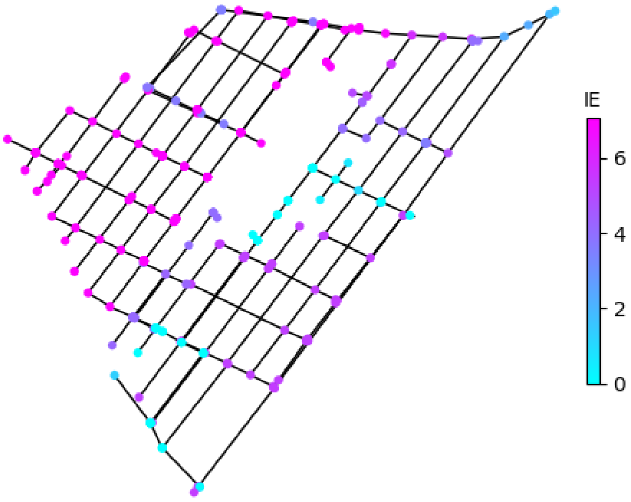


Figure 5.1-11 Results for IE calculation, Th = 0.5 mH2O.

Figure 5.1-11 shows the information entropy values for each node, In the areas where there is no influence of the pumping stations, the highest values of IE are obtained, this is because the number of nodes that can be detected in these places is higher compared to the other parts of the network, likewise it can be seen again how the nodes that did not detect any node have values of IE equal to zero (cyan color).

The use of this parameter makes more sense to evaluate a set of sensors and not an individual sensor, because using a set of sensors a more varied and much more extensive list of detected nodes can be obtained, however, there is a tendency to obtain higher IE values in the zones with higher percentage of detection, which is in line with the objectives of using this parameter.

5.1.3 Sensor Detection Value (SDV) – a proposal

Due to the problems encountered in using the value of the information (VOI, a new objective function is proposed; this new parameter is named Sensor Detection Value (SDV) and is based on the concept of the value of the information, using parameters such as π_s, Pd_J and MLR. The SDV of a group of nodes ***J*** respecting a node ***i*** can be represented as follows:

$$SDV_{i,J} = \frac{\pi_s^i * Pd_J}{n_{i,J}} \qquad [20]$$

Where:

π_s^i : It is the probability that a leak is present at node ***i***; it can be related to the prior belief that the expert judgment has about 'Leak' state in any part of the network.

Pd_J: It is the probability that a sensor ***j*** or group of sensors ***J*** send a message ***m*** about the occurrence of leaks at ***i***; it means detection probability, similar to the parameter $q_{m,s}^J$ used in the calculation of VOI but without considering false alarms and false positives.

$n_{i,J}$: the number of the leak in the position *n* of the leak within the 10 leaks analysed n = {1, 2, 3, …., 10} required for nodes ***J*** to detect a leak at node ***i***, (MLR), if the leak is not detected use 10^6

Equation [20] presents parameters similar to those used in the calculation of the VOI, with some differences, these are:

- The only state(s) evaluated is the 'Leak' state; therefore, only the prior belief of leaks is used.
- Sensors' false alarms are not considered. Only the detection probability in leak cases is used.
- If the leak at node ***i*** are not detected by the sensor group ***J***, SDV is approximately zero (denominator equal to 10^6).
- SDV is proportional to the belief about the existence of leaks and how effective the sensors are to detect leaks and is indirectly proportional to the leak number required for the sensor group ***J*** to detect a leak at node ***i***.

Once the SDV value is determined, the final SDV value is equal to the sum of these individual values divided in L_p (number of nodes where a leak was included), thus the maximum value of SDV will be equal to one (1):

$$SDV_J = \frac{1}{Lp} * \sum_{i=1}^{L_p} SDV_{i,J} \quad [21]$$

With this function, it is possible to trade-off the belief of the state of the system and the information provided by the sensors, removing the problem caused by the high dependence of the function on the previous belief.

Moreover, since the objective of using the VOI is because it was an indicator of the quality of the information provided by the sensors, which could be used to prioritize the detection of areas with greater vulnerability or those believed to be more susceptible to leaks, this parameter can also be used for the same purpose, because if higher prior beliefs are defined for the nodes with greater need, the SDV values will be higher, so that in the definition of the optimisation algorithm this parameter should be maximized. The disadvantage is that the decision diagram cannot be used, and the results of this equation are only an indication of the quality of the information given by the sensors, and not and indicator for decision making.

Once the objective functions are calculated, the optimisation process is applied to find the best set of solutions, next section describes the results obtained.

5.1.4 Results of the optimisation problem

Figure 5.1-12 shows the trade-off curve between the two objective functions defined in the NSGA-II optimisation model, where information entropy (IE, vertical axis) and sensor detection value (SDV, horizontal axis) are maximized. Each solution is composed for four sensors and the threshold selected was 0.5 mH2O.

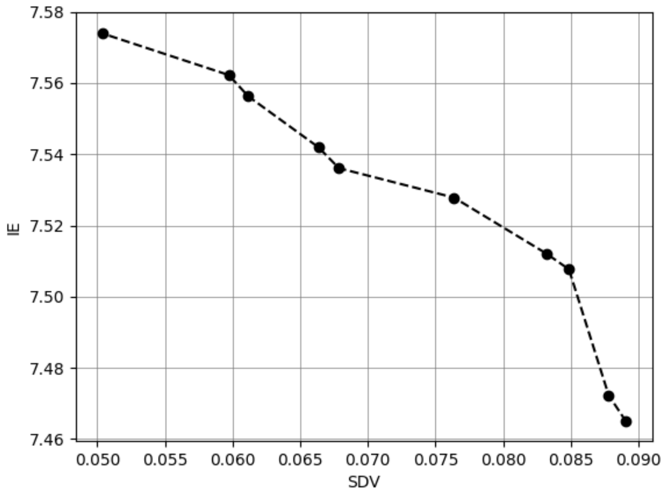


Figure 5.1-12 Pareto front representing the optimal solutions for the sensor deployment, number of sensors = 4, Th = 0.5 mH2O.

Although IE and SDV metrics were explained in previous chapters, for a decision-maker is easier to evaluate the performance of a sensor networks using practical parameters such as the probability of detection, average leak required to detect leaks and the quantity of nodes that could be identified.

Table 5.1-7 shows the results of the 2480 simulated leak scenarios (248 leak nodes * 10 leaks per node). Nodes detected indicates the numbers of nodes identified in at least one of the ten evaluated leaks, the % of detection correspond to the Nodes detected over the total of nodes analysed (248), the percentage of cases detected indicates what percentage of the 2,480 simulated scenarios were detected by the sensor group, and the average leak flow is the sum of all the MLR for each node divided by the total of nodes where leaks were placed.

Additionally, the values of these parameters are calculated for the set of sensors already placed (SAP) to compare the performance of our proposed solutions against the solution installed.

Table 5.1-7 Pareto optimal layout of pressure sensors

Solution	IE bits	SDV	Nodes Detected	% of detection	% of cases detected	Average leak flow (l/s)
1	7.46	0.089	194	78%	51%	8.57
2	7.57	0.050	203	82%	58%	10.64
3	7.53	0.076	207	83%	57%	9.51
4	7.47	0.088	194	78%	58%	8.59
5	7.56	0.060	205	83%	59%	10.40
6	7.51	0.085	207	83%	55%	9.19
7	7.51	0.083	207	83%	58%	9.29
8	7.54	0.068	205	83%	55%	10.05
9	7.56	0.061	205	83%	59%	10.30
10	7.54	0.066	205	83%	58%	10.16
SAP	7.14	0.045	171	69%	47%	9.58

In general, all solutions have a high detection rate in terms of detected nodes (greater than 78%), However, considering the percentage of detected cases there were no solution above 60%, if we relate this value to the average leak flow, we see that most of the leaks were detected in the range of 9 to 11 l/s, so leaks of 4, 5.78 and 7.56 l/s (which represent 30% of the analysed scenarios) were probably only detected in very few cases.

Moreover, very similar results were obtained for the information entropy parameter, varying between 7.47 and 7.56, indicating that all groups of sensors detect probably the same group of nodes and is mainly because the size of the network, this, indicates that the parameter is not a differential factor for the selection of final solutions and we could study the possibility of modifying it or replacing it with some other objective function that would provide a better indicator for the choice of solutions. Thus, the best solutions will correspond to those with the highest SDV, since in this parameter larger variability was found, showing high values for the cases with the major percentage of detections and the lower leak flow required, recalling that for all cases a prior belief (π_s) of 50% for 'Leak' state was assumed.

Figure 5.1-13 shows the ten sensor layouts selected from the Pareto front, it is important to note that none of the solutions considered, includes at least one of the sensors already installed.

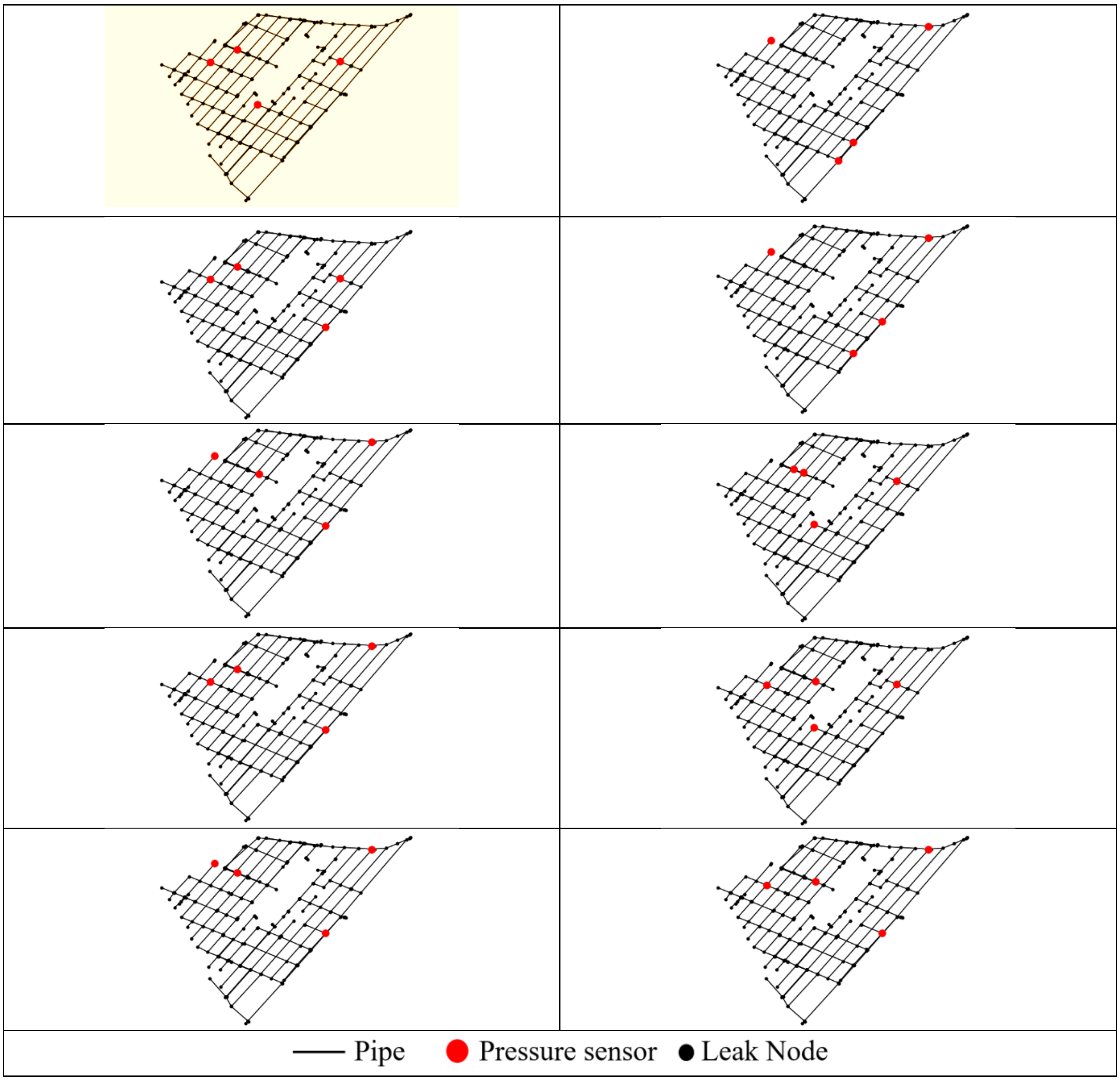


Figure 5.1-13 sensor layout selected from the Pareto front, number of sensors = 4, Th = 0.5 mH2O.

To visualize in more detail the behaviour of each of the sensors in a specific solution, the layout of sensors number one (with higher SDV) was taken as a sample. Figure 5.1-14 shows the detection area of each of the sensors in this solution.

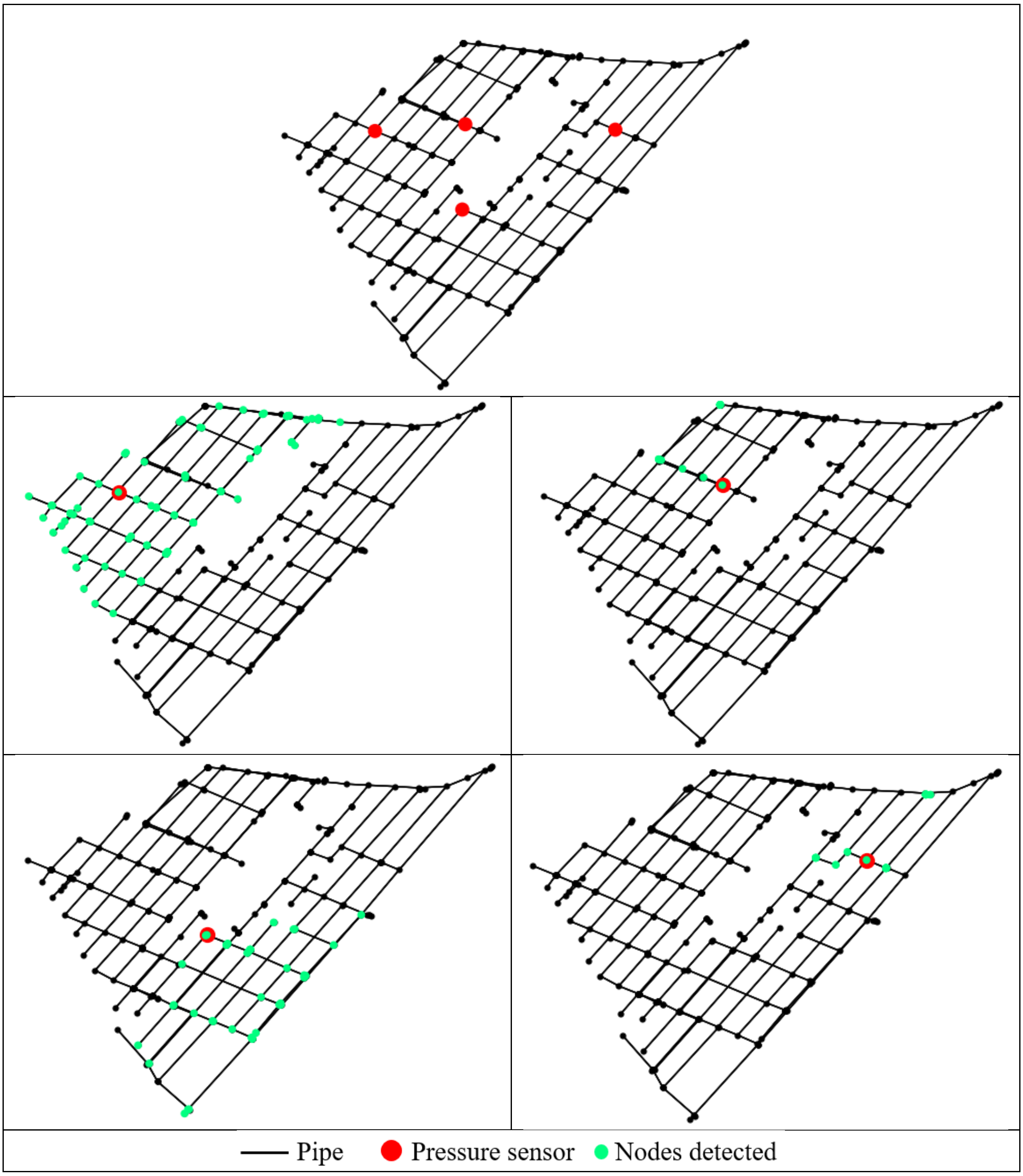


Figure 5.1-14 Sensor layout for solution 5, number of sensors = 4, Th = 0.5 mH2O.

As desired, sensors are equitably distributed in such a way that they can detect leaks in most of the nodes, providing a well complemented group of sensors. It can also be seen that two of the four sensors can detect most of the nodes, this type of behaviour was also seen in all the other solutions where two of the four sensors detected most of the leaks and these were combined with two other points that detected few nodes but corresponded to zones that the other two could not detect.

5.2 Discussion

5.2.1 Comparison with the existing sensor deployment, (Four sensors)

In relation to the sensor layout installed, we found that its performance is significantly inferior compared to the proposed solutions, according to Table 5.1-7 it has 47% of detected cases and detects leaks in 171 out 248 nodes (69%), also the parameters of IE and SDV were lower to the proposed solutions, therefore we can conclude that the solutions proposed by this methodology presented better results regarding the leak detection.

The same exercise of visualizing the behaviour of each of the sensors was performed using the four sensors already installed, Figure 5.2-1 shows which areas are detected by each node.

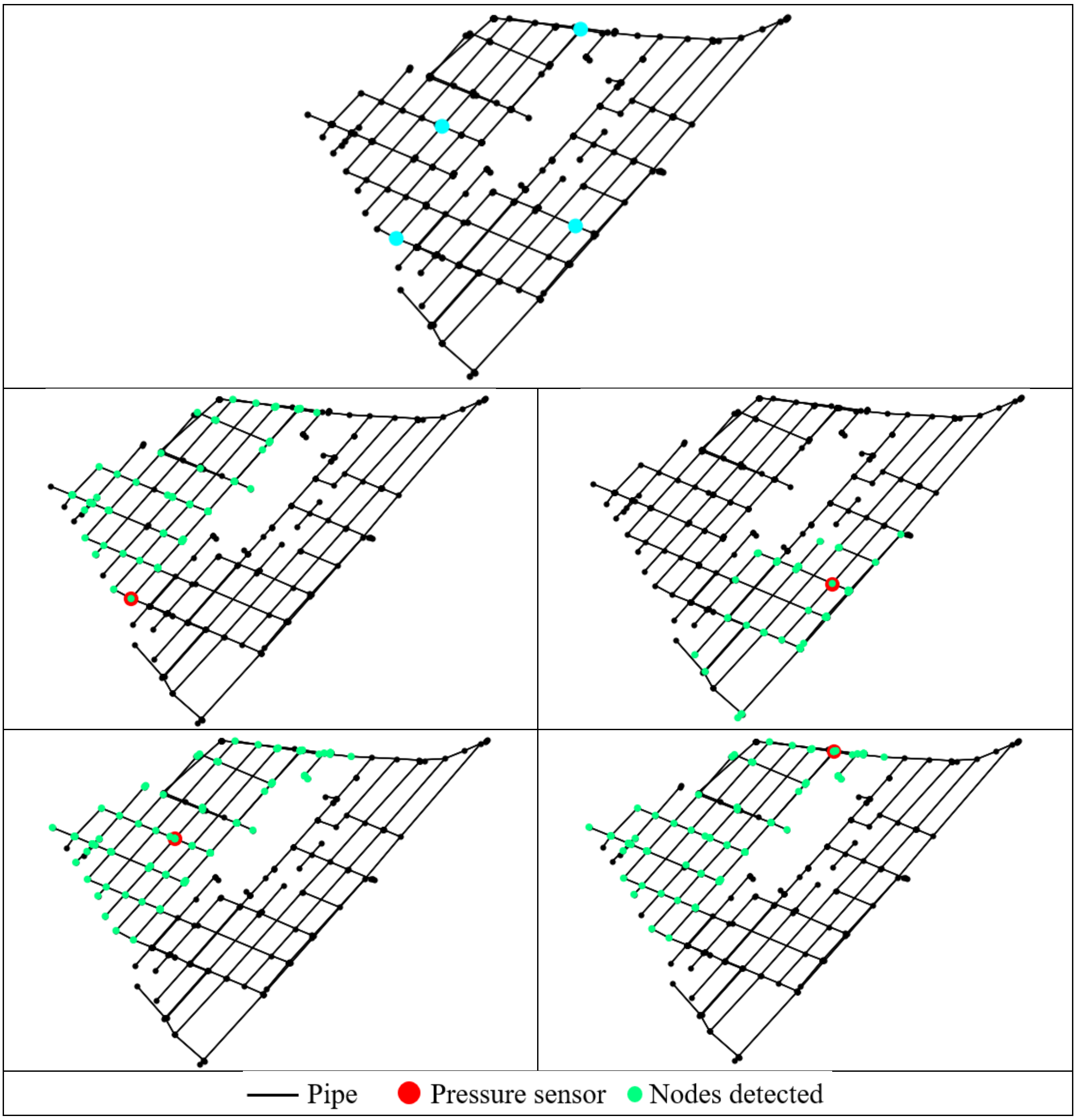


Figure 5.2-1 Sensor layout for sensors already placed, number of sensors = 4, Th = 0.5 mH2O

Although the deployment of sensors has a wide geographic distribution It is clear that the information provided by them is not diversified. Based on our methodology, the sensors are redundant, because 3 out 4 provide information from the same areas, so the coverage is lower compared to the solutions proposed.

5.2.2 Comparison with the existing sensor deployment, (Eight sensors).

According to the objectives of the water utility of Braila, it is desired to estimate the best arrangement of eight sensors, of which four are already installed and the remaining ones will be proposed. Using the methodology proposed in this thesis, we obtained the most optimal set of solutions shown in Figure 5.2-2.

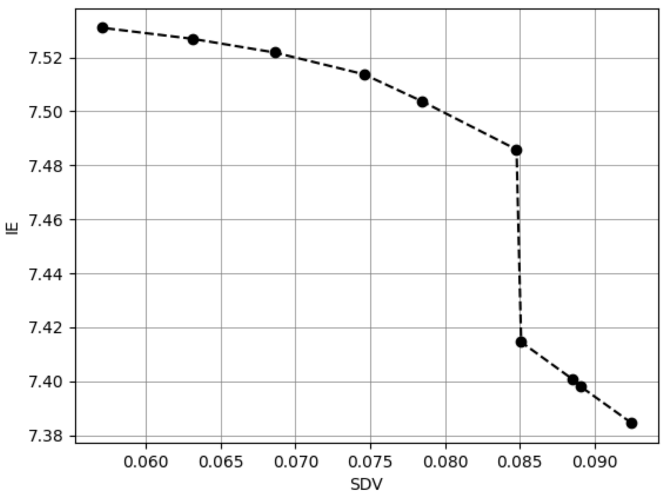


Figure 5.2-2 Pareto front representing the optimal solutions for the sensor deployment, number of sensors = 8 (4 new + 4 installed) , Th = 0.5 mH2O.

Figure 5.2-3 shows the distribution of sensors with the highest SDV, geographically they are well distributed and in terms of coverage, this group of sensors was able to detect 207 nodes out 218 nodes and 59% of the cases with an average leakage of 9.08 l/s.

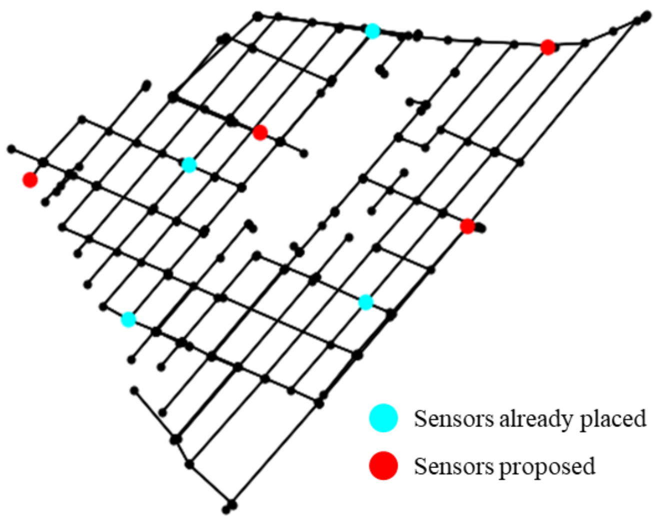


Figure 5.2-3 Sensor layout for eight sensors (Four already placed + Four proposed), Th = 0.5 mH2O

According to the results shown in Table 5.2-1, if we compare the performance of this distribution of eight sensors with those obtained by installing the four proposed sensors, we see that the improvement obtained is negligible, and this is because as explained above, 3 out of 4 installed sensors provide the same information about the network.

An extra analysis was performed, but this time the algorithm was allowed to choose the eight sensors without fixing the four sensors already installed. With the new arrangement of eight sensors it is possible to have 89% of detected nodes (220) and the percentage of detected cases is slightly increased. We can also observe that the difference between installing four sensors and installing eight sensors is negligible, the improvement obtained is about 5% (13 nodes), and this is mainly because of the size of the network, since it is relatively small and does not require the use of such large number of sensors to cover it, however, if the budget is not a problem, installing eight sensors will help to have a slightly better coverage and a greater support among the network of sensors.

Table 5.2-1 Performance of sensor deployment for solutions with highest SDV, multiple scenarios

Scenario	IE bits	SDV	Nodes Detected	% of detection	% of cases detected	Average leak flow (l/s)
4 Installed	7.14	0.045	171	69%	47%	9.58
4 Proposed	7.51	0.085	207	83%	55%	9.19
4 Installed + 4 Proposed	7.4	0.088	207	83%	59%	9.08
8 Proposed	7.7	0.093	220	89%	60%	9.45

Figure 5.2-4 shows the distribution of the eight proposed sensors, is remarkable that again in none of the solutions obtained from the Pareto set, any of the sensor already installed was selected.

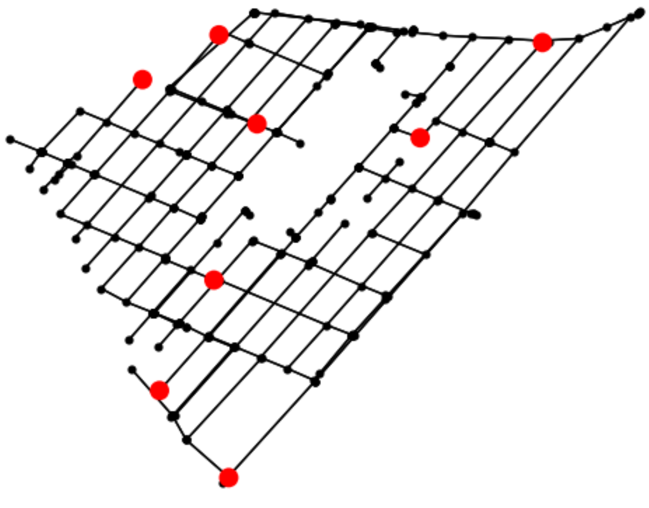


Figure 5.2-4 Sensor layout for eight sensors (all proposed), Th = 0.5 mH2O

All the solutions presented above correspond to the results of pressure comparison using a threshold value of 0.5 mH2O and was maintained fixed in each scenario, the next part shows the results obtained by varying the threshold.

5.2.3 Towards a robust design of sensor locations

In section 2.2.1 was defined that threshold value is uncertain when there is not enough information about the historical variation of the pressures, and for this reason it is necessary to evaluate multiple threshold values to assess the behaviour of the system at different pressure drops to select the most robust solution. Previously, the set of optimal solutions for a fixed threshold value of 0.5 mH2O has been obtained, however, this group of solution could be optimal only for the state in which the network was evaluated and could be sub-optimal when the model experiences variation in its parameters (Marquez-Calvo, 2020). Therefore, in the following section we will analyse the variation of the threshold and the different solutions obtained, in addition we will study a method to obtain the most robust solution within the set of solutions.

Initially, the process explained in section 4.2.3 has to be repeated to each of the thresholds, the results are displayed in Figure 5.2-5

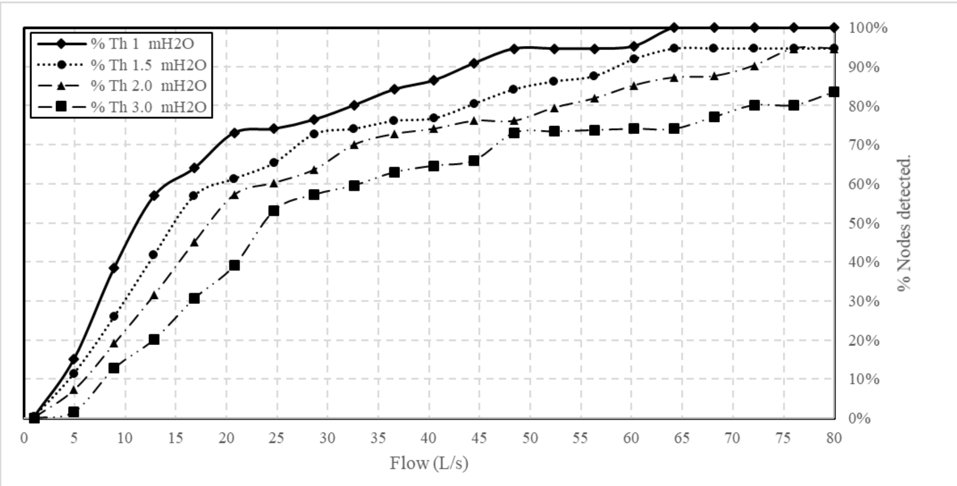


Figure 5.2-5 relation between leak values and cases of detection, Th = 1, 1.5, 2 and 3 mH2O.

Since the threshold value increased by six times (3 mH2O), higher leak values were evaluated, taking a range of leaks from 1 l/s to 80 l/s, to identify the ($\boldsymbol{L_{f\,20}}$) and ($\boldsymbol{L_{f\,80}}$) for the different threshold values.

It is important to clarify that according to the Table 4.4-2 the daily flow rate of the system is 40 l/s, so obtaining leaks above this value is very unrealistic, therefore, each of the leaks evaluated to obtain $\boldsymbol{L_{f\,80}}$ does not correspond to any real leakage but is the mathematical value that the model requires to generate the desired pressure drops. Table 5.2-2 indicates that as the pressure threshold increases, the flow rate required to cause pressure drops above this value is also increased.

Table 5.2-2 values of ($\boldsymbol{L_{f\,20}}$) and ($\boldsymbol{L_{f\,80}}$) for different threshold.

Threshold mH2O	Lf_{20} (l/s)	Lf_{80}(l/s)
0.5	4	20
1.0	6	33
1.5	8	45
2.0	9	52
3.0	13	72

It is important to note that the Lf_{80} corresponding to the threshold of 3.0 mH2O is 72 l/s (almost double of the demand required by the entire system), this is an hypothetical situation that indicates that the hydraulic model has the capacity to supply much more water than the system requires, without generating a hydraulic disequilibrium because when using a pressure-based model, the demands of the nodes where pressures drops above 5 mH2O are not supplied and all the pumping equipment and tanks work to supply the demand requested by the leak, again the model can solve the system but these results are far from reality. On the other hand, the value of Lf_{20} for the 3.0 mH2O threshold requires that the network loses 33% of its total flow (13 l/s) at a single point, which is excessive for a distribution system.

These analysis indicates that for analysing real systems, it is necessary to evaluate the sensitivity of the network and choose a threshold value that fits the characteristics of the system and the historical pressure variation of each of the nodes, and not to take typical values from other networks because in each network there will be different responses to the added leaks. On the other hand, these results alert us to the importance of the interpretation of the results obtained from hydraulic models, because mathematically all equations can be solved but the output can be quite different from reality.

Moreover, it can be concluded that to ensure leak detection coverage, the system must be subjected to significant water losses, which in practice would be impossible, so it is not important to capture leaks in 100% of the network, only to focus on those points that respond to a leak according to the characteristics of the system, and based on the identification of these points to establish the area of analysis.

In spite of the results obtained above, it is assumed that these are the values to be analysed and the respective optimization analysis and comparison of the response of the distribution system to the variation of the different thresholds of pressure will be performed.

Pareto set considering uncertainty.

Figure 5.2-6 shows the trade-off curve between the two objective functions defined in the NSGA-II optimisation model, where information entropy (IE, vertical axis) and sensor detection value (SDV, horizontal axis) are maximized. Each solution is composed for four sensors and each threshold evaluated is composed by 10 possible sensors' layout.

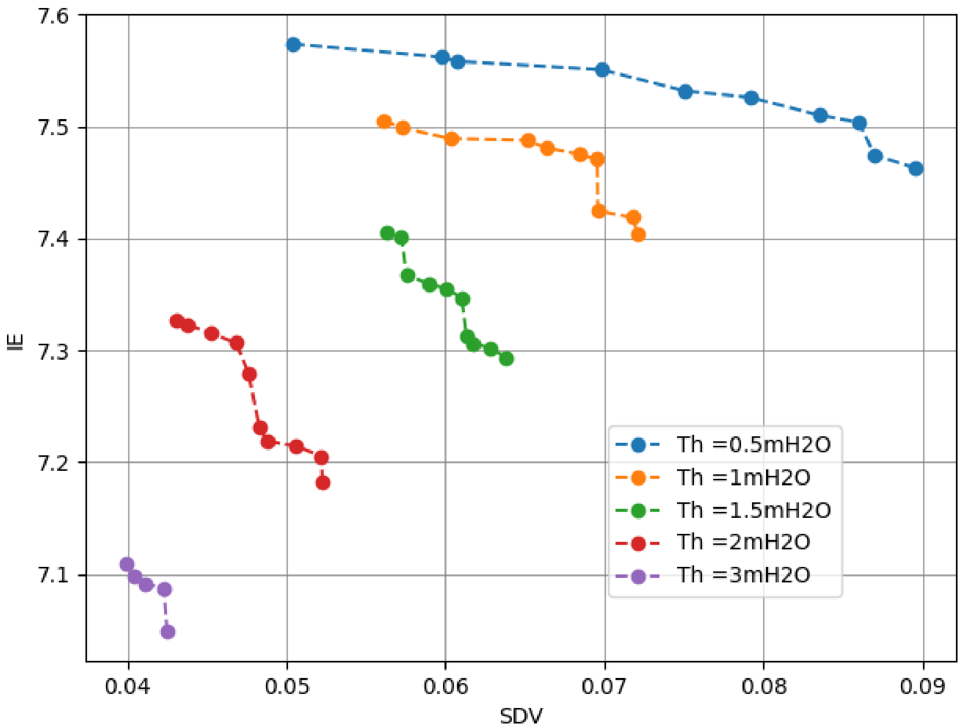
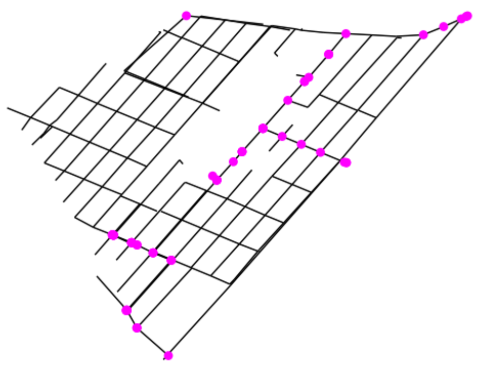


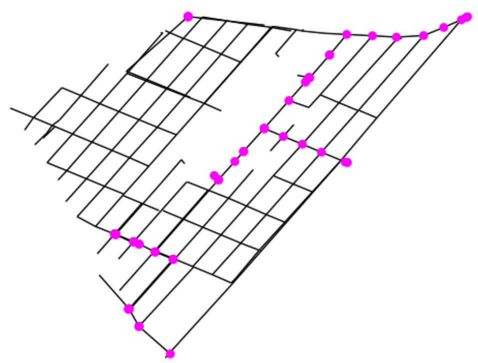
Figure 5.2-6 Pareto front representing the optimal solution for the sensor deployment, number of sensors = 4, Th = 0.5, 1, 1.5, 2, 3 mH2O.

According to the different Pareto sets obtained we can observe that as the pressure threshold value increases the entropy values decrease, this is caused because the number of nodes detected by the group of sensors is reduced, as fewer nodes present pressure drops above the higher threshold value. On the other hand, the SDV value is also affected, because for higher threshold values, higher leak values are also required to cause the pressure drops, and as the probability of detection by the sensors decreases, the SDV value is reduced.

With the list of solutions obtained, we verified if the number of non-detected nodes changed between the different scenarios, since we used the leaks of $(\boldsymbol{L_{f\,20}})$ and $(\boldsymbol{L_{f\,80}})$ a minimum coverage was always guaranteed, Figure 5.2-7 indicates that the number of non-detected nodes between the lowest threshold (0.5 mH2O) and the highest threshold (3 mH2O) increased from 57 to 65 respectively, so we can guarantee that the solutions obtained between the different scenarios can be compared in terms of percentage of detection.



(a) nodes no detected = 57
Th = 0.5 mH2O



(b) nodes no detected = 65
Th = 3.0 mH2O

Figure 5.2-7 Map with nodes not detected for two pressure threshold evaluated.

Selection of the most robust solution

In total, five scenarios were analysed, with ten solutions per scenario and four nodes per solution, for a total of 200 possible nodes for sensor location. Naturally, many of the nodes that were selected among the different solutions were selected more than once in each of the scenarios, so the objective for selecting the most robust solution will be to obtain the group of nodes that provide the best performance based on some ranking system.

Table 5.2-3 shows the list of the different nodes selected, the times that a node appeared in the different solutions and in the different scenarios.

Table 5.2-3 List of nodes selected in all the scenarios analysed.

Node	Number of times	Number of scenarios
236-A	8	4
Jonctiune-2204	6	2
Jonctiune-2195	6	2
Jonctiune-J-1	5	1
275-A	5	1
Jonctiune-3961	5	2
Jonctiune-J-21	5	2
Jonctiune-2194	5	2
Jonctiune-2192	5	1
Jonctiune-4706	5	1
Jonctiune-2203	4	1
Jonctiune-2729	4	1
Jonctiune-2191	3	1
Jonctiune-3967	3	2
256-B	3	2
Jonctiune-3425	2	1
257-A	2	1
Jonctiune-2777	2	1
255-A	2	2
236-B	2	1
237-A	2	1
203-A	1	1
202-A	1	1
Jonctiune-2179	1	1
246-B	1	1
Jonctiune-2738	1	1
Jonctiune-2774	1	1
Jonctiune-2197	1	1
256-A	1	1

As we can see, there are several nodes that were selected for the different solutions, being node 236-A the one that appeared more times in each of the solutions (8 times) and equally this node appeared in 4 out of the 5 scenarios analysed. It is important to clarify that when a node appears many times, this is not a reason to say that it can be part of the robust solution, because a node can appear 10 times, but those 10 times could correspond to the same scenario, which makes it a suitable node for that scenario, but not for the whole set of solutions. For this reason, the relevant score will be the number of scenarios in which the node appears and the number of times it appeared in each scenario will be analysed as a tiebreaker criterion, so the score of each node will be equal to:

$$Score\ (j) = Ns_j + \frac{Nt_j}{\max(Nt_J)} \qquad [22]$$

Where:

Ns_j: Number of scenarios in which a node $\boldsymbol{j}$ was selected.

Nt_j: Number of times in which a node $\boldsymbol{j}$ was selected in all the scenarios.

$\max(Nt_J)$: Maximum number of times that a node was selected between all the possible nodes (8 in this case)

In this way each node has an individual score, however, selecting the four nodes with the highest score as the most robust solution is not correct, since individually they have high scores, but as a group, they could have poor performance, so each solution of four nodes would be taken and the individual scores of each element would be summed, the solution with the highest score would be selected as the most robust solution.

Figure 5.2-8 shows the distribution of each of the selected nodes and the number of times they were selected, the most frequently selected nodes are located in the area where there is no pump influence, indicating that the incorporation of these hydraulic elements to the Epanet model definitely influences the final selection of the nodes where sensors will be installed.

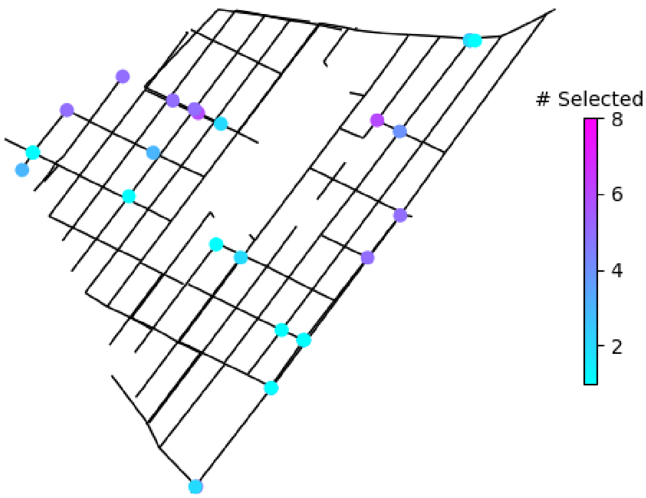


Figure 5.2-8 number of times that each node was selected as a possible location for a pressure sensor.

Using equation [22] we obtain that the solution with the highest score is the one composed by nodes J-2192, J-2195, J-21 and 236-A, obtaining a score of 12.01, the distribution of these sensors is shown in Figure 5.2-9

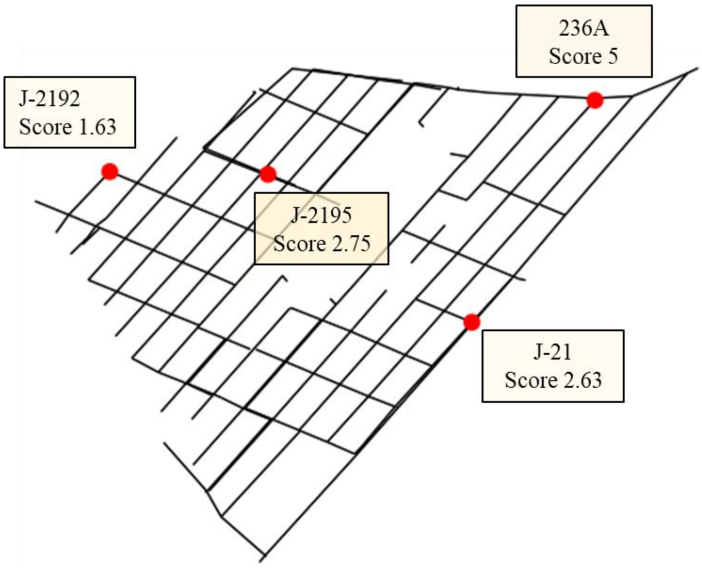


Figure 5.2-9 Sensor layout of the robust solution.

The performance of this solution with respect to the other scenarios is shown in Table 5.2-4

Table 5.2-4 Performance of the robust solution at other scenarios

Threshold (mH2O)	IE bits	SDV	Nodes Detected	% of detection	% of cases detected	Average leak flow (l/s)
0.5	7.53	0.077	207	83%	58%	9.5
1.0	7.48	0.064	200	80%	53%	16.44
1.5	7.39	0.056	191	77%	49%	23
2.0	7.33	0.043	178	72%	43%	28
3.0	7.09	0.030	153	62%	35%	41

Table 5.2-4 indicates that the average flow for the detection of each scenario is very high compared to the daily flow of the system (40 l/s), this as explained previously is due to the form in which the hydraulic model was elaborated and the installation of pumps that supply the system in failure scenarios.

The performance in each solution is excellent for the first 3 scenarios, where the detection percentage was higher than 77%. The worst detection percentage was obtained for the threshold scenario of 3 mH2O (62%) and this was because few nodes present such high pressure drops. However, the solutions of this scenario presented average detection percentages of around 65%, so, this solution has an acceptable performance in scenarios of critical pressure drops and an excellent performance in scenarios of minimal pressure drops (0.5 mH2O), concluding that the solution selected can be considered as a robust solution.

Chapter 6 Conclusions and Recommendations

A methodology to optimally place sensor for leak detection in WDS has been developed and successfully applied to one real study case, the novelty presented in this work was the use of a pressure-driven analysis for the different hydraulic simulations and the incorporation of human perspective about the state of the system, using concepts such as the value of information and information entropy. In addition, analysis of sources of uncertainty was included by varying pressure thresholds and evaluating different leak sizes.

The methodology was based on the comparison of the pressures of an initial state without leaks (healthy state) with the pressure deviations generated at each node in response to the incorporation of new demands (leaks), associating the occurrence of leaks to scenarios in which the pressure drops were greater than a selected pressure threshold (Th). Each of the simulations was performed using the Epanet numerical model incorporating pressure-based analysis in which the demands are a function of system pressures. The results of the hydraulic simulation were used for the formulation of the objective functions employed in the multi-objective optimisation problem, in which it was proposed to maximize the value of the information as a measure of the quality of the information transferred by the sensors and to maximize the value of the information entropy as a measure of coverage. This optimisation was performed by using evolutionary algorithms such as NSGA-II.

As a result of the implementation of this methodology, it was possible to propose the correct distribution of sensors for a distribution system located in the city of Braila, in Romania, comparing the sensor network obtained with this methodology and the sensor network already installed, concluding that the solutions proposed provided better results regarding the leak detection.

From the results obtained it is possible to answer the research questions formulated by this thesis:

- *How to formulate a methodology to optimise the location of pressure sensors, using concepts of value of information and information entropy?*

The formulation of this novel methodology incorporating parameters such as VOI and Information entropy is described step by step throughout the document, providing acceptable results. However, was found that the VOI was highly dependent on the decision-maker's perception of the system state and constrained the use of the information coming from the sensors. Therefore, a new objective function based on the VOI principle was proposed, nevertheless, all the procedure and formulation required for the use of the VOI parameter was explained and it was indicated that its use could be limited, leading to a revision for future research.

- *How can pressure-driven analysis be performed, coupling the available modelling systems and Python?*

All the calculation procedures required for the evaluation of leak scenarios, calculation of detection parameters, definition of objective functions and optimization were fully automated by using an algorithm developed in the programming language Python, the algorithm was composed by own functions and special libraries designed by external researchers, including the Water Network Tool for Resilience (WNTR) package, which has the possibility of using the solution algorithm developed by the United States Environmental Protection Agency (EPA) for solving pressurized networks using a pressure-driven approach.

- *To what extent it is appropriate to use the concept of the value of information as a parameter that includes the perception of the decision-maker?*

Based on the results obtained two conclusions can be drawn. First, the use of the value of information as a parameter that includes the perception of the decision-maker is very useful and totally appropriate, since it permits to relate multiple variables to evaluate the quality of the information coming from external sources, with the additional benefit of being able to establish a decision tree that helps in taking the actions that generate the highest benefit or the lowest loss, in this way optimal solutions can be found but also adjusted to the decision-makers' requirements.

Second, the use of the VOI is limited due to its influence on the results and high dependence on the prior belief of the decision-maker, which makes difficult the generalization of the methodology. However, VOI principles provided a useful tool to formulate the parameter of the sensor detection value (SDV), which was correctly implemented as an objective function in the optimization problem, being used as an indicator of the quality of the information provided by the sensors and as a prioritization parameter in the detection of areas with greater vulnerability or those believed for the decision-makers to be more susceptible to leaks.

- *How relevant is the incorporation of different leaks and pressures threshold as a source of uncertainty in the evaluation of sensor networks?*

This study demonstrated that the variation of pressure thresholds leads to multiple potential nodes for sensor location, requiring the use of a methodology to obtain the best performing solution in the different scenarios evaluated (robust solution), this finding is contrary to that found by Raei et al., (2019) who indicated that the variation of the threshold may have a negligible effect on the location of the sensors.

In addition, it was shown that the use of multiple leaks alters the magnitude of the coverage of the network and the performance of each node in the system in leak detection, so for methodologies based on the use of pressure thresholds it is highly relevant to use multiple values of leaks and therefore find the appropriate size that fits the needs of the desired pressure drop and minimum coverage required.

On the other hand, one of the importance of incorporating different values of pressure thresholds and leak sizes, is the possibility of obtaining solutions adapted to future conditions, considering a distribution system as a dynamic problem in which leaks could gradually increase (increase in demands) and the accuracy of the sensors are deteriorated over time, plans to install sensors can be incorporated, in which the sensor nodes with the best performance in long term can be installed from the short term.

Finally, we can conclude that the methodology proposed in this thesis for the optimal location of pressure sensors can be implemented in any distribution network and some suggestions for future studies are:

- As opposed to testing the network for leaks that cause the desired pressure drops at a single instant (sometimes the leaks are very high), a cumulative pressure delta could be studied, determining for each node the time required to cause that the sum of each pressure change over the time steps exceeds a desired threshold, thus smaller leak values could be used.
- By incorporating pressure-based models, the variation of demands at each of the nodes could be evaluated as a parameter for comparison and selection of potential nodes for sensor location.

  Perform more deep uncertainty analysis, using a larger number of pressure threshold values in order to generate a wider list of the Pareto sets and use robust optimization approaches such as the ones presented by Marquez-Calvo (2020).
- Eliminate nodes that require very large leaks to generate specific pressure drops and work with only those nodes that are sensitive to the addition of small leaks, thus increasing detection rates and optimizing the use of the value of the information approach.

References

Alfonso, L., Mukolwe, M. M., & Di Baldassarre, G. (2016). Probabilistic Flood Maps to support decision-making: Mapping the Value of Information. *Water Resources Research*, *52*(2), 1026–1043. https://doi.org/10.1002/2015WR017378

Alfonso, Leonardo, & Price, R. (2012). Coupling hydrodynamic models and value of information for designing stage monitoring networks. *Water Resources Research*, *48*(8), 1–13. https://doi.org/10.1029/2012WR012040

Blesa, J., Nejjari, F., & Sarrate, R. (2014). Robustness analysis of sensor placement for leak detection and location under uncertain operating conditions. *Procedia Engineering*, *89*, 1553–1560. https://doi.org/10.1016/j.proeng.2014.11.453

Bohorquez, J., Alexander, B., Simpson, A. R., & Lambert, M. F. (2020). Leak Detection and Topology Identification in Pipelines Using Fluid Transients and Artificial Neural Networks. *Journal of Water Resources Planning and Management*, *146*(6), 1–11. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001187

Braun, M., Piller, O., Deuerlein, J., & Mortazavi, I. (2017). Limitations of demand-A nd pressure-driven modelling for large deficient networks. *Drinking Water Engineering and Science*, *10*(2), 93–98. https://doi.org/10.5194/dwes-10-93-2017

Capponi, C., Ferrante, M., Zecchin, A. C., & Gong, J. (2017). Leak Detection in a Branched System by Inverse Transient Analysis with the Admittance Matrix Method. *Water Resources Management*, *31*(13), 4075–4089. https://doi.org/10.1007/s11269-017-1730-6

Cheung, P. B., Van Zyl, J. E., & Reis, L. F. R. (2005). Extension of Epanet for pressure driven demand modelling in water distribution system. *Proceedings of the 8th International Conference on Computing and Control for the Water Industry, CCWI 2005: Water Management for the 21st Century*, *1*(January).

Choi, Y. H., & Kim, J. H. (2019). Self-adaptive models for water distribution system design using single-/multi-objective optimisation approaches. *Water (Switzerland)*, *11*(6). https://doi.org/10.3390/w11061293

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017

Farley, B., Boxall, J. B., & Mounce, S. R. (2009). Optimal locations of pressure meters for burst detection. *Geotechnical Special Publication*, *187*, 747–757. https://doi.org/10.1061/41023(337)63

Farley, M., & Trow, S. (2015). Losses in Water Distribution Networks: A Practitioners' Guide to Assessment, Monitoring and Control. *Water Intelligence Online*, *4*(0), 9781780402642–9781780402642. https://doi.org/10.2166/9781780402642

García, V. (2003). *Modelación de la Demanda Urbana de Agua*. 551.

Germanopoulos, G. (1985). A technical note on the inclusion of pressure dependent demand and leakage terms in water supply network models. *Civil Engineering Systems*, *2*(3), 171–179. https://doi.org/10.1080/02630258508970401

Grayson, C. J. (1960). *Decisions under uncertainty; drilling decisions by oil and gas operators.*

Harvard University, Division of Research, Graduate School of Business Administration.

Greyvenstein, B., & Van Zyl, J. E. (2007). An experimental investigation into the pressure - Leakage relationship of some failed water pipes. *Journal of Water Supply: Research and Technology - AQUA*, *56*(2), 117–124. https://doi.org/10.2166/aqua.2007.065

Hadka, D. (2015). *Release*.

Hirshleifer, J., & Riley, J. G. (1979). The Analytics of Uncertainty and Information-An Expository Survey. *Journal of Economic Literature*, *17*(4), 1375–1421. http://www.jstor.org/stable/2723720

Hosseini, M., & Kerachian, R. (2017). A data fusion-based methodology for optimal redesign of groundwater monitoring networks. *Journal of Hydrology*, *552*, 267–282. https://doi.org/10.1016/j.jhydrol.2017.06.046

Jansen, T. (2013). *Analysing evolutionary algorithms. The computer science perspective*. https://doi.org/10.1007/978-3-642-17339-4

Jung, D., & Kim, J. H. (2017). Robust meter network for water distribution pipe burst detection. *Water (Switzerland)*, *9*(11). https://doi.org/10.3390/w9110820

Khorshidi, Mohammad S, Nikoo, M. R., & Sadegh, M. (2018). Optimal and objective placement of sensors in water distribution systems using information theory. *Water Research*, *143*, 218–228. https://doi.org/10.1016/j.watres.2018.06.050

Khorshidi, Mohammad Sadegh, Nikoo, M. R., Taravatrooy, N., Sadegh, M., Al-Wardy, M., & Al-Rawas, G. A. (2020). Pressure sensor placement in water distribution networks for leak detection using a hybrid information-entropy approach. *Information Sciences*, *516*, 56–71. https://doi.org/10.1016/j.ins.2019.12.043

Marquez-Calvo, O. O. (2020). Advancing Robust Multi-Objective Optimisation Applied to Complex Model-Based Water-Related Problems. In *Advancing Robust Multi-Objective Optimisation Applied to Complex Model-Based Water-Related Problems*. https://doi.org/10.1201/9781003026617

Marquez Calvo, O. O., Quintiliani, C., Alfonso, L., Di Cristo, C., Leopardi, A., Solomatine, D., & de Marinis, G. (2018). Robust optimisation of valve management to improve water quality in WDNs under demand uncertainty. *Urban Water Journal*, *15*(10), 943–952. https://doi.org/10.1080/1573062X.2019.1595673

Milly, P. C. D., Betancourt, J., Falkenmark, M., Hirsch, R. M., Kundzewicz, Z. W., Lettenmaier, D. P., Stouffer, R. J., Dettinger, M. D., & Krysanova, V. (2015). On Critiques of 'stationarity is Dead: Whither Water Management?' *Water Resources Research*, *51*(9), 7785–7789. https://doi.org/10.1002/2015WR017408

Muranho, J., Ferreira, A., Sousa, J., Gomes, A., & Sá Marques, A. (2014). Pressure-dependent demand and leakage modelling with an EPANET extension - WaterNetGen. *Procedia Engineering*, *89*, 632–639. https://doi.org/10.1016/j.proeng.2014.11.488

Nejjari, F., Sarrate, R., & Blesa, J. (2015). Optimal pressure sensor placement in water distribution networks minimizing leak location uncertainty. *Procedia Engineering*, *119*(1), 953–962. https://doi.org/10.1016/j.proeng.2015.08.979

Pathirana, A. (2012). EPANET2 desktop application for pressure driven demand modelling. *Water Distribution Systems Analysis 2010 - Proceedings of the 12th International*

Conference, WDSA 2010, *41203*(February), 65–74. https://doi.org/10.1061/41203(425)8

Pérez, R., Puig, V., Pascual, J., Peralta, A., Landeros, E., & Jordanas, L. (2009). Pressure sensor distribution for leak detection in Barcelona water distribution network. *Water Science and Technology: Water Supply*, *9*(6), 715–721. https://doi.org/10.2166/ws.2009.372

Ponce, M. V. C., Castañón, L. E. G., & Cayuela, V. P. (2014). Model-based leak detection and location in water distribution networks considering an extended-horizon analysis of pressure sensitivities. *Journal of Hydroinformatics*, *16*(3), 649–670. https://doi.org/10.2166/hydro.2013.019

Pudar, R. S., & Liggett, J. A. (1992). Leaks in Pipe Networks. *Journal of Hydraulic Engineering*, *118*(7), 1031–1046. https://doi.org/10.1061/(asce)0733-9429(1992)118:7(1031)

Quintiliani, C., Marquez-Calvo, O., Alfonso, L., Di Cristo, C., Leopardi, A., Solomatine, D. P., & De Marinis, G. (2019). Multiobjective Valve Management Optimisation Formulations for Water Quality Enhancement in Water Distribution Networks. *Journal of Water Resources Planning and Management*, *145*(12), 1–10. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001133

Quintiliani, C., & Vertommen, I. (2020). *Optimal Pressure Sensor Locations for Leak Detection in a Dutch Water Optimal Pressure Sensor Locations for Leak Detection in a Dutch Water Distribution Network †*. *September*. https://doi.org/10.3390/environsciproc2020002040

Raei, E., Nikoo, M. R., Pourshahabi, S., & Sadegh, M. (2018). Optimal joint deployment of flow and pressure sensors for leak identification in water distribution networks. *Urban Water Journal*, *15*(9), 837–846. https://doi.org/10.1080/1573062X.2018.1561915

Raei, E., Shafiee, M. E., Nikoo, M. R., & Berglund, E. (2019). Placing an ensemble of pressure sensors for leak detection in water distribution networks under measurement uncertainty. *Journal of Hydroinformatics*, *21*(2), 223–239. https://doi.org/10.2166/hydro.2018.032

Reddy, L. S., & Elango, K. (1989). Analysis of water distribution networks with head-dependent outlets. *Civil Engineering Systems*, *6*(3), 102–110. https://doi.org/10.1080/02630258908970550

Rice, E. L. (2014). *on Behalf of*. *45*(1), 15–109.

Rossman, L. a. (2000). *Epanet 2*. *September*, 104. http://www.epa.gov/nrmrl/wswrd/dw/epanet.html

Sanz, G., Pérez, R., Kapelan, Z., & Savic, D. (2016). Leak detection and localization through demand components calibration. *Journal of Water Resources Planning and Management*, *142*(2), 1–13. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000592

Segura, J. L. A. (2010). *Optimisation of Monitoring Networks for Water Systems* (Issue January 2010).

Shaqadan, A. (2008). *DigitalCommons @ USU Decision Analysis Considering Welfare Impacts in Water Resources Using the Benefit Transfer Approach*.

Steffelbauer, D. B., & Fuchs-Hanusch, D. (2016). Efficient Sensor Placement for Leak Localization Considering Uncertainties. *Water Resources Management*, *30*(14), 5517–5533. https://doi.org/10.1007/s11269-016-1504-6

Taravatrooy, N., Nikoo, M. R., Hobbi, S., Sadegh, M., & Izady, A. (2020). A novel hybrid entropy-clustering approach for optimal placement of pressure sensors for leakage detection in water distribution systems under uncertainty. *Urban Water Journal*, *17*(3), 185–198. https://doi.org/10.1080/1573062X.2020.1758162

Tavamani, J. P. (2016). Research Article Bernoulli Equation in Fluid Flow. *International Journal of Current Research*, *2*(10), 2–4.

Wagner, B. J. M., Shamir, U., & Marks, D. H. (1988). *options proposed , a simulation of these options should be done to gain a better understanding of how the proposed alternative systems will be likely to behave under real-life conditions . This paper presents an event-oriented , discrete simulation progra*. *114*(3), 276–294.

Appendices

Appendix A. - Research Ethics Declaration Form

Part A – Personal Ethics Exemption letter

United Nations Educational, Scientific and Cultural Organization | IHE DELFT Institute for Water Education under the auspices of UNESCO

Research Ethics Committee
IHE Delft Institute for
Water Education
E ResearchEthicsCommitee@un-ihe.org

Date:	29 March 2021
To:	Andrés López García
MSc Programme:	WSE-HI
Approval Number:	IHE-RECO 2020-103

Subject: Research Ethics approval

Dear Andrés López García,

Based on your application for Ethical Approval, the Research Ethics Committee (RECO), IHE Delft RECO has been approved ethical clearance for your research proposal Methodology to Optimally Place Pressure Sensors for Leak Detection in Water Distribution Systems using Value of Information and Entropy.

This approval valid until April 16, 2021. You need to notify the RECO of any modifications to your research protocol. If you do not complete your research by the specified date, you should to contact RECO to request an extension.

Please keep this letter for your records and include a copy in the final version of MSc. Thesis, together with your personal reflection. Additional information is available at https://ecampusxl.un-ihe.org/course/view.php?id=1555&section=2.

On behalf of the Research Ethics Committee, I wish you success in the completion of your research.

Yours sincerely,

AMS

Dr. Angeles Mendoza Sammet
Acting Ethics Coordinator

Copy to: Archive.

Part B – Personal written ethical statement.

The following will explain how the five main principles (honesty, scrupulousness, transparency, independence, and responsibility) required by the Netherlands Code of Conduct for Research Integrity were applied in each of the steps of this research:

Design Research

The approach of this research was based on the development of a methodology to help solve a problem of high social impact, such as the loss of water caused by the generation of leaks in distribution systems, these problems are extended for any WDS, reason why all the contribution to identifying leaks are required. The work presented in this thesis can be easily verified and replicated, having an important social relevance.

As a result of this research, it is expected to have a practical methodology for deploying pressure sensors for leak detection, reducing water volume loss. In particular, for Braila's city case, which has the goal of reducing water losses caused by leaks from a value of 750 L/h/km to a value of 50 L/h/km.

Conduct Research

This research is proposed for the NAIADES project. NAIADES is a 3-year EU funded project to generate a holistic water ecosystem for digitalization of urban water sector. The objective is to reduce uncertainty in the modelling exercise and to produce big data related to critical events to feed an Artificial Intelligence platform. Braila, Romania is a city in Muntenia, eastern Romania and a port on the Danube River., For this purpose, full responsibility was taken to use precise and widely known mathematical and physical concepts, likewise all the base information with which the different simulations were developed was manipulated and reviewed with transparency, without making alterations that lead to obtaining biased results.

In addition, the document was written in a clear and concise manner so that it can be used by the community outside of academia and the scientific community, each of the steps described are very well detailed, with examples where required and observations without omitting any information.

Report & Results

For the development of this methodology, a wide bibliography and references were consulted in each of the topics studied, each of the ideas presented in this document are properly referenced without taking credit for concepts or proposals developed by others.

The analysis and conclusions obtained in this research show that part of the results were not as expected to the initial approaches but nevertheless were properly analysed and supported, so that the reason for these new findings is understood, without hiding or masking results, on the contrary, alternatives were proposed to address the problems from another point of view.

Finally, it is made clear that some parts of the research could not be carried out due to time limitations and are expected to retake some steps in future research, the interpretation of the results is subjective based on the author's own knowledge and compared with scientific foundations found in the literature, which when used was adequately cited in the document.

Assessment & Peer Review

The development of this research was supported by expert mentors on the subject, therefore many of the decisions were made based on a joint decision, in addition we had the support of the external entity of the city where the project was carried out, thus having a work that benefited the own and other people's interests.

Communication of Research

All calculations and results presented in this methodology are accurate and reliable, no manipulation of information was incurred, thus demonstrating transparency in each of the results obtained, ensuring that the results are replicable and precise.

To summarize, it is noted that the five principles required by the Netherlands Code of Conduct for Research Integrity were considered throughout each of the phases of this research, leading me through an honest study.

Appendix B. - Scripts

Leak simulation and formulation of objective functions:

File: MainFunctions.py

```
===========================================================================
# Packages Required
===========================================================================
'''
Most of the libraries needed to run this utility are installed with python
itself.

Some of the packages not included and to be installed are:

Platypus: Platypus is a framework for evolutionary computing in Python with a
focus on multiobjective evolutionary algorithms (MOEAs).
more information: https://platypus.readthedocs.io/en/latest/index.html

WNTR: The Water Network Tool for Resilience is a Python package designed to
simulate and analyse resilience of water distribution networks. More
information in:
https://wntr.readthedocs.io/en/latest/overview.html

Plotly: Plotly's Python graphing library makes interactive, publication-
quality
graphs.
More information in: https://plotly.com/python/
'''

===========================================================================
# Libraries Required.
===========================================================================
import wntr
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import itertools
from platypus import NSGAII, Problem, Subset, unique, nondominated
import time
from collections import Counter
from scipy import stats

import plotly.express as px
from plotly.offline import plot
import random
from pyitlib import discrete_random_variable as drv

===========================================================================
# Fuctions Required:
===========================================================================
'''
To save space in the code and reuse the instructions in multiple scenarios,
we developed this section with different functions according to the needs.

In addition, some of the functions incorporated in the different packages
have
a difficult structure to memorize and read, for this reason they were
rewritten
```

```python
def RunNet (x):
    '''
    This function receives as variable an instance of a network created with
the
    OpenNet() function.
    This returns an object with all the results of the model, the queries have
    to be related to the instance created with this function.
    How to do the queries, visit:

    https://wntr.readthedocs.io/en/latest/waternetworkmodel.html

    Example: Results = RunNet(MyInstance)'''
    sim = wntr.sim.EpanetSimulator(x)
    sim = sim.run_sim(version=2.2)
    return sim

class Network:
    def __init__(self, file):
        self.file = file

    def OpenNet(self):
        '''
                This function receives the name of the file in string format
        with the extention .inp, it creates an instance of the network, it is
        important to open the folder
        where the file is located.

        Example:  MyInstance =  OpenNet('Mynetwork.inp')

               With this instance we can modify physical characteristics of
       the network an
        options before Running the model.

                Note: By now is important that the network created with
        EPANET should be
                saved using version 2.2, with SI units, 24 hours of
        simulation, 1-hour interval,
                demand model = PDD, these parameters should be modified in
        EPANET for EASIER
        use of the tool.
        '''
        return wntr.network.WaterNetworkModel(self.file)

    def GraphNet(self):
        '''
                This function receives as variable the name of the network in
        .inp format
        This returns a graph of the network
        Example:  GraphNet('Mynetwork.inp')'''
        wntr.graphics.plot_network(self.OpenNet(), title= 'Network')
```

```
    def BaseDemand (self, y = "No_defined"):
        '''
                This function receives as variables an instance of a network
        (x), this
                could be created with the OpenNet() function, and the name of
         the nodes
         (y, in list format) to be consulted.
         If any node is specified then all the base-demands are showed.
        '''
        x = self.OpenNet()
        BD = []
        if y == "No_defined":
            for i in x.junction_name_list:
                Node = x.get_node(i)
                if Node.base_demand>0:
                    BD.append(Node.base_demand)
        else:
            for i in y:
                Node = x.get_node(i)
                if Node.base_demand>0:
                        BD.append(Node.base_demand)
        return BD
#==============================================================================
# MAIN FUNCTION
#==============================================================================

def CalculateParameters(self,Leak_Nodes='All',Sensor_Nodes='All',Time=1,
        Method ='PDD',
        Leakmin='None',Leakmax='None',Num_Leakages='None',Threshold=0.5,\
        PriorBelief=0.5):

        start = time.time()  #start of the Timer, just for evaluate code
                              performance.
        Net = self.OpenNet()  #Instance of the network
        if Leak_Nodes == 'All':
            Leak_Nodes = Net.junction_name_list #Nodes where leaks are going
                                                 to be placed

        if Sensor_Nodes == 'All':
            Sensor_Nodes = Net.junction_name_list

        Net.options.hydraulic.demand_model=Method   #Using PDD as demand
                                                     method
        Net.options.time.duration = Time*24*3600    #Time of simulation
        Net_Sim = RunNet(Net)  #Run network, base case.
        St =
        int(Net.options.time.duration/Net.options.time.hydraulic_timestep)
        BPM = Net_Sim.node['pressure'].loc[1:St*3600,Net.junction_name_list]
        #Base pressures
        BD = sum(self.BaseDemand())/len(self.BaseDemand()) #Average Demand of
        the system in m3/s
```

```
if Leakmin == 'None':
    Leakmin = 0.1*BD*1000  #Minimum Leak added to nodes L/s
if Leakmax == 'None':
    Leakmax = BD * 1000 #Maximum Leak added to nodes L/s
    if Leakmax <= Leakmin:
        Leakmax = 2*Leakmin
if Num_Leakages == 'None':
    Num_Leakages = 9
    Leakinterval = (Leakmax - Leakmin)/Num_Leakages
if Num_Leakages != 'None':
    Leakinterval = (Leakmax - Leakmin )/(Num_Leakages)
LeakFlows = np.arange(Leakmin,Leakmax+0.001,Leakinterval)

LPM = []          #Leak pressure matrix
LM = []           #Leakage Matrix
DM = []           #Divergence matrix

for i in Leak_Nodes:
    for k in range(len(LeakFlows)):

      LeakFlow = [LeakFlows[k]/1000]*(24*Time+1) #array of the leak
       flow (m3/s)
      Net.add_pattern(name ='New', pattern = LeakFlow) #Add New \
       Patter To the model
      Net.get_node(i).add_demand(base= 1, pattern_name='New') #Add
       leakflow
      Net.options.time.duration = 24*Time*3600 #Time of simulation
      Net_New = RunNet(Net) # Run new model
      Net2 = Net_New.node['pressure'].loc[1:St*3600,Sensor_Nodes].\

     rename_axis('Node_'+i+','+str(round(LeakFlows[k],2))+'LPS',axis=
     1) # Give name to the dataframe
      LPM.append(Net2) # Save pressure results
      Difference = BPM[Sensor_Nodes].sub(Net2, fill_value=0) # Create
       divergence matrix
      DM.append(Difference.abs().rename_axis('Node_'+ i+', '\
            +str(round(LeakFlows[k],2))+'LPS', axis=1)) # Save
       Divergence M.
      LM.append(pd.DataFrame([k*1000 for k in LeakFlow[1:]], columns
     = ['LeakFlow']\

                   , index =list(range(3600,St*3600+3600,3600))).\
                     rename_axis('Node: ' + i, axis=1)) #Save
       leakflows used

      # Restore initial conditions:
      Net = self.OpenNet()
      Net.options.hydraulic.demand_model = Method # Change type of
       simulation to PDD
      Net.options.time.duration = 24*Time*3600
      Net_Sim = RunNet(Net)
      TM_ = []    # time when the leak was identified
```

```python
WLM = []    # Water loss Matrix (L/s), how much water is wasted

for i in range(len(DM)):
    TMtemp = []
    WLMtemp = []
    for j in Sensor_Nodes:
        WLMtemp2 = []
        for k in range(len(DM[0])):
            if DM[i][j][(k+1)*3600] <= Threshold:
                WLMtemp2.append(LM[i].LeakFlow[(k+1)*3600]*3600)
            else:
                WLMtemp2.append(LM[i].LeakFlow[(k+1)*3600]*3600)
                break
        TMtemp.append(k+1)
        WLMtemp.append(sum(WLMtemp2))
    TM_.append(TMtemp)
    WLM.append(WLMtemp)
'''
These matrices indicate the time in which each sensor could detect a
leakage here we created a dataframe indicating where the perturbation
was placed and the time of detection of each sensor.
'''
TM = []
for i in np.arange(0,len(DM),len(LeakFlows)):
      TM.append(pd.DataFrame(TM_[i:i+len(LeakFlows)],columns=Sensor_No
      des,index=LeakFlows). rename_axis('Node'+
      Leak_Nodes[int(i/len(LeakFlows))]))

FP = []     #Flow required for causing a detectable perturbation.

for k in range(len(TM)):
    FPtemp=[]
    for j in Sensor_Nodes:
        for i in range(len(LeakFlows)):
            if TM[k][j][LeakFlows[i]] == Time*24:
                pass
            else:
                FPtemp.append(LeakFlows[i])
                break
        if TM[k][j][LeakFlows[i]] == Time*24:
            FPtemp.append(1000000)
    FP.append(FPtemp)
FP = pd.DataFrame(FP, columns=Sensor_Nodes, index=Leak_Nodes)

'''
In this part the probability of detection of each sensor is calculate,
then we counted how many times in the multiple scenarios a sensor could
detect the leakage, this gave us a matrix of probability of detection
and not detection
'''
```

```
        Mes_Leak = []

        for i in Sensor_Nodes:
            Not_Detect = []
            for j in range(len(Leak_Nodes)):
                a = TM[j][i].value_counts().to_dict()
                if (Time*24 in a) == True:
                    Not_Detect.append(a[Time*24])
                else:
                    Not_Detect.append(0)
            Pro_D = 1-sum(Not_Detect)/(len(Leak_Nodes)*len(LeakFlows))
            Mes_Leak.append([Pro_D,(1-Pro_D)])

        Mes_Leak = pd.DataFrame(Mes_Leak,columns=['Detection','No
Detection'],index=Sensor_Nodes)

        self.LeakNodes = Leak_Nodes
        self.SensorNodes = Sensor_Nodes
        self.Leakmin = Leakmin
        self.Leakmax = Leakmax
        self.Leakinterval = Leakinterval
        self.LeakFlow = LeakFlows
        self.TimeDetection = TM
        self.FlowRequired = FP
        self.Perc_Detection = Mes_Leak
        self.LeakPressureMatrix = LPM
        self.BasePressureMatrix = BPM
        self.DivergenceMatrix = DM
        self.NumLeakages = Num_Leakages
        self.pb = PriorBelief

        print ("it took", time.time() - start, "seconds.") #Time taken by the
                                                             algorithm
# =============================================================================
# Objective Functions
# =============================================================================
# Value Of Information (VOI)

    def VOI(self,FalseAlarm=0.15,Cas=[[-5,-15],[-15,0]],\
            N='Sensor Nodes',L= 'List of Leak Nodes') :
      S = [self.pb,1-self.pb]
      NodesDetected = []
      QNodesDetected = []
      FlowRequiredD  = []
      for i in L:
          if (self.FlowRequired[N].loc[i] <= self.LeakFlow[-1]).any():
              NodesDetected.append(i)
              Lf = self.FlowRequired[N].loc[i].min()
              a = self.LeakFlow.tolist().index(Lf)
              QNodesDetected.append(len(self.LeakFlow)-a)
              FlowRequiredD.append(a+1)
          else:
              FlowRequiredD.append(self.FlowRequired[N].loc[i].min())
```

```
  Pro_Detection =
  sum(QNodesDetected)/(len(self.LeakNodes)*len(self.LeakFlow))
  Mes_Leak = [Pro_Detection, 1 - Pro_Detection]
  Mes_Noleak = [FalseAlarm,1-FalseAlarm]

  VOI = []
  for i in range(len(L)):
  #Posterior Probability:
     A = [S[0] * k for k in Mes_Leak]
     B = [S[1] * k for k in Mes_Noleak]
     C = (A[0]+B[0])
     D = (A[1]+B[1])
     E = A[0]/C
     F = A[1]/D
     PSM =[[E,F],[1-E,1-F]]
     G = FlowRequiredD[i] # New leak flow used
     H = [Cas[0][0]-G,Cas[0][1]-G*3] # Recalculation of CAS matrix
     I = [S[0]*l for l in H]  #expected utility decision Go to check
     J = [S[1]*m for m in Cas[1]] #expected utility decision Not to Go t
     U0 = max(I[0]+J[0],I[1]+J[1]) #Best option without extra information
   #Matrix of new utility with information:
     Cos = [[H[0],Cas[1][0]],[H[1],Cas[1][1]]]
     U1 = np.dot(Cos,PSM)
     U_1 = [max([n[0] for n in U1]),max([n[1] for n in U1])]
    #Value of each message:
     V =[n - U0 for n in U_1]
     if A[0]==0: #we evaluate if the sensor is avaible to detect a least
     # one leak event otherwise the value of information is zero
      VOI.append(0)
     else:
      VOI.append(C*V[0]+D*V[1])
  self.NodesDetected = NodesDetected
  self.Pro_Detection = Pro_Detection
  return(sum(VOI))


def Entropy (self,N='Sensor Nodes',L='List of Leak Nodes'):
    m = (len(self.LeakFlow)-1)/(self.Leakmax-self.Leakmin)
    b = m*self.Leakmin-1
    list2 = self.FlowRequired[N]*m-b
    list2[list2 >10000]=0
    DetectAcum = [int(x) for x in list2.sum(axis=1).tolist()]
    #NoDetectAcum = [(len(self.LeakFlow)*len(N))-i for i in DetectAcum]
    ND = []
    for i in range(len(self.LeakNodes)):
        ND.append([i+1]*int(DetectAcum[i]))
        #ND.append([0]*int(NoDetectAcum[i]))
    ND = sum(ND,[])
    if len(ND)==0:
        IE = 0
    else:
        IE = drv.entropy(ND).tolist()
```

```
        self.EntropyList = ND
        return (IE)

# Sensor Detection (SDV)

    def SDV (self, N = 'Sensor Nodes', L = 'List of Leak Nodes'):
      NodesDetected = []
      QNodesDetected = []
      FlowRequiredD  = []
      FlowDetected = []
      for i in L:
          if (self.FlowRequired[N].loc[i] <= self.LeakFlow[-1]).any():
              NodesDetected.append(i)
              Lf = self.FlowRequired[N].loc[i].min()
              a = self.LeakFlow.tolist().index(Lf)
              QNodesDetected.append(len(self.LeakFlow)-a)
              FlowRequiredD.append(a+1)
              FlowDetected.append(Lf)
          else:
              FlowRequiredD.append(self.FlowRequired[N].loc[i].min())
      Pro_Detection =
      sum(QNodesDetected)/(len(self.LeakNodes)*len(self.LeakFlow))
      SDV = sum([self.pb*Pro_Detection/i for i in
      FlowRequiredD])/len(self.LeakNodes)
      self.NodesDetected = NodesDetected
      self.Pro_Detection = Pro_Detection
      self.FlowDetected = sum(FlowDetected)/len(FlowDetected)
      return (SDV)
```

Multi-Objective Optimisation problem:

File: Optimisation.py

```
from MainFunctions import *
File = 'Braila.inp'  # File used for the model
network = Network(File) # Object of the file
NodeL = [] # List of nodes where leaks are not required
NodeS = []  # List of nodes where sensors are not required
NodesSen = network.DelNodes(NodeS)
NodesLeak = network.DelNodes(NodeL)
Solutions =[]
network.CalculateParameters(Threshold=0.5,Leakmin=4, Leakmax=20,\
                            Leak_Nodes=NodesLeak,Sensor_Nodes=NodesSen)
=============================================================================
# Optimisation
=============================================================================
A_S = 4
F_Nodes = []
if len(F_Nodes) != 0:
    for i in range(len(F_Nodes)):
        network.SensorNodes.pop(network.SensorNodes.index(F_Nodes[i]))
def Optimal(x):
    y = np.append(x[0],F_Nodes).tolist()
    O_1 = network.SDV(N=y,L=network.LeakNodes)
    O_2 = network.Entropy(y,network.LeakNodes)
    return [O_1,O_2]
problem = Problem(1,2)
problem.types[:] = Subset(network.SensorNodes,A_S) # define randon nodes in
the whole system
problem.function = Optimal # Function created in previous steps.
problem.directions[0] = Problem.MAXIMIZE # VOI and H(x) has to be maximized,
by default Objetives are minimized
problem.directions[1] = Problem.MAXIMIZE
algorithm = NSGAII(problem,population_size=10)
algorithm.run(10000)
Best_S = unique(nondominated(algorithm.result))
Solutions.append(Best_S)
============================================================================
# PARETO GRAPH
============================================================================
for i in range(len(Solutions)):
    VOI=[s.objectives[0] for s in Solutions[i]]
    JE =[s.objectives[1] for s in Solutions[i]]
    Elements = [tuple(i) for i in [s.variables[0] for s in Solutions[i]]]
    df = pd.DataFrame({'Nodes':Elements,'Information Entropy': JE,\
                       'Value Of
    Information':VOI}).sort_values(by=['Information Entropy'])
        plt.plot(df['Value Of Information'].to_list(),df['Information
    Entropy'].to_list(),\
     linestyle='--', marker='o',label = 'Th ='+ str(round(Th[i],2))+'mH2O')
    plt.grid(linestyle ='-', linewidth=0.5, color='gray')
    plt.xlabel('SDV')
    plt.ylabel('IE')
    #plt.yticks(np.arange(7.46,7.60,0.02))
    plt.legend(loc=(0.6,0.1),frameon=True)
    plt.grid(linestyle ='-', linewidth=0.5, color='gray')
```